Machine Learning Phase Transitions: A Probabilistic Perspective

Inauguraldissertation
zur
Erlangung der Würde eines Doktors der Philosophie
vorgelegt der
Philosophisch-Naturwissenschaftlichen Fakultät
der Universität Basel

von

Julian Arnold

Basel, 2025

Genehmigt von der Philosophisch-Naturwissenschaftlichen Fakultät

auf Antrag von

Erstbetreuer:	Prof. Dr. Christoph BRUDER
Zweitbetreuer:	Prof. Dr. Stefan GOEDECKER
Externer Experte:	Prof. Dr. Juan Felipe CARRASQUILLA ÁLVAREZ

Basel, den 18. März 2025

Prof. Dr. Heiko SCHULDT, Dekan

Summary

The detection of phase transitions and the classification of matter into its distinct phases are some of the most fundamental tasks in physics. For a long time, however, these tasks required extensive prior theoretical knowledge, human intuition, and system-specific expertise to be solved. This thesis explores the use of machine-learning methods to automate the detection of phase transitions. Here, we focus on three key machine-learning methods – so-called supervised learning, learning by confusion, and the prediction-based method – that aim to autonomously identify phase transitions by learning from data, offering an alternative to traditional approaches in physics. Since their inception, these machine-learning methods were largely treated as black boxes: They were shown to work on a case-by-case basis but their fundamental operating principles remained elusive. In particular, a theory explaining when and why they succeed or fail has been lacking. The methods were also computationally intensive and primarily restricted to applications in the domain of physics. This thesis provides several key advances:

In the first part of this thesis, we establish a rigorous theoretical foundation by deriving the optimal predictive models underlying these methods and prove that these approximate the system's Fisher information from below. This explains the successes and limitations of these machine-learning approaches in detecting different types of phase transitions. Based on these insights, we demonstrate ways how to make the methods more computationally efficient through techniques like multi-tasking and the use of generative classifiers. This makes them practical for analyzing larger systems and datasets. Similarly, these insights motivate key modifications to the methods that make them more reliable at detecting phase transitions and allow them to map out phase diagrams in higher-dimensional parameter spaces.

In the second part of this thesis, we venture beyond physics and extend these methods to detect qualitative changes in complex systems more broadly, in particular modern artificial intelligence systems and real-world data. This includes the discovery of phase-transition-like phenomena in large language models as a function of their temperature, training epoch, and prompt parameters, as well as significant changes in news articles over time.

Through this progression from physics to broader application domains, this thesis demonstrates how methods originally developed for understanding phase transitions in physical systems can be refined to provide valuable insights into complex systems more generally. The methods are shown to be capable of uncovering abrupt changes in diverse settings, ranging from spin systems at equilibrium to state-of-the-art generative artificial intelligence and daily news.

Acknowledgments

I have experienced excellent support from a large collective of great people during my doctoral studies.

I would like to express my deepest gratitude to my supervisor Professor Christoph Bruder for his invaluable counseling throughout my doctoral journey. Christoph, your ability to foster curiosity and independent thinking, while providing clear and thoughtful insights, has been instrumental in shaping my research. I am especially grateful for the freedom you granted me to explore my own ideas. You allowed me to develop my own research identity while providing guidance whenever needed. Thank you also for the many doors you opened for me and the many opportunities you created for me, from enabling me to attend workshops and conferences to connecting me with other researchers. Your encouragement, patience, and kindness have made this experience not only enriching but truly enjoyable. Remembering back to the time I was attending your electrodynamics lecture always sparks my admiration for your dedication to teaching: You create an environment where students feel genuinely comfortable asking questions and engaging in discussions, making even the most challenging physics concepts accessible and exciting to explore.

I would also like to thank my second supervisor, Prof. Stefan Goedecker, for the regular PhD committee meetings during my time as a PhD and his flexibility in that regard. Furthermore, I am thankful to Prof. Juan Carrasquilla who kindly agreed to co-referee my thesis, and to Prof. Admir Greljo for chairing my defense.

At the start of my PhD, I was fortunate enough to join a 2-week summer school in Warsaw that taught me loads and connected me with amazing people who I have been in contact with ever since. A special thank you goes to Alexander Gresch, Jakub Vrábel, Kaelan Donatella, Kim Nicoli, Rouven Koch, Borja Requena, Sebastian Wetzel, Evert van Nieuwenburg, and Eliška Greplová – and of course to Anna Dawid and Alexandre Dauphin for organizing this impactful event.

After that, many exciting conferences and intellectually stimulating discussions followed. Thank you, in particular, to Roope Uola, Oleg Kaikov, Koji Inui, Rodrigo Vargas-Hernández, Lucas Spangher, Kaito Kobayashi, Jianchao Zhang, and Ziming Liu for your time. I remember our conversations fondly to this day.

My visits to the research groups of Prof. Martin Kliesch, Prof. Florian Marquardt, and Prof. Alan Edelman also made a long-lasting impression on me. Thank you for this opportunity. I particularly remember the lengthy discussions I had with Alexander Gresch and Lennard Bittel in Düsseldorf that sparked new ways of thinking about the infamous trio of phase-transition detection methods and were highly influential to my subsequent work.

Throughout my time in Basel, I have been incredibly fortunate to work alongside people who are incredibly talented, inspiring, and supportive alike. Thank you Frank Schäfer, Axel Lode, Martin Žonda, Juan Carlos San Vincente Veliz, Debasish Koner, Narendra Singh, Raymond Bemish, Markus Meuwly, Flemming Holtorf, Chris Rackauckas, Alan Edelman, Niels Lörch, Difei Zhang, Csaba Zsolnai, and Christoph

Bruder. Our collaborations have not only fundamentally shaped the work in this thesis, but also deeply enriched my experience as a researcher. It has been a privilege and a joy to learn from and with you all.

I am particularly grateful to my closest collaborators who have profoundly influenced my development as a researcher. First and foremost, I want to express my gratitude to Frank Schäfer, who has been both a mentor and a friend throughout my PhD journey. His tireless work ethic and constant willingness to help have been truly remarkable. Frank has taught me invaluable lessons about computational physics – from the practical aspects of working with slurm and GitHub to the intricacies of Julia programming, ODE solving, and optimal control. Beyond his technical guidance, I am especially thankful for his role in helping me build my scientific network and establish new collaborations. His dedication to helping young researchers find their footing in academia is exceptional.

I am deeply grateful to Flemming Holtorf, whom I had the pleasure of meeting through Frank. Flemming's mathematical prowess and remarkably broad knowledge across different fields never cease to amaze me. His careful and critical approach to scientific work has been invaluable – having him convinced of a project's merit has always felt like passing the highest bar of scrutiny.

To Niels Lörch, I owe special thanks for sharing his wealth of experience and unique perspective on physics. His ability to distill complex problems into their fundamental elements has been eye-opening. Beyond physics, I cherish our wide-ranging discussions about economics, finance, and entrepreneurial ventures. His mathematical yet pragmatic approach to problems, combined with his encouraging "go-for-it" attitude, has been both inspiring and educational. I look forward to continuing our conversations about the many ways to apply mathematical thinking to practical challenges.

Thank you also to the Master students, I could co-supervise at the University of Basel from whom I learned a lot: Benjamin Senn, Heinz Krummenacher, Jan Neuser, and Leon Behrens.

I thank the present and former members of the Bruder group for the amazing atmosphere: Martin Koppenhöfer, Pavel Sekatski, Gaomin Tang, Ryan Tan, Frank Schäfer, Tobias Nadolny, Tobias Kehrer, Petr Zapletal, Niel Lörch, and our long-time visitor Markus Hoffmann. I cherished our tradition of having lunch together and hope that these were not our last meals!

I would also like to thank the (former) members of the group of Prof. Markus Meuwly – Juan Carlos San Vincente Veliz, JingChun Wang, Kai Töpfer, Silvan Käser, Eric Boittier, and Luis Itza Vazquez Salazar, and Prof. Meuwly himself – for keeping me part of their circle and inviting me to join their group activities. It has been fun to explore and learn about the "dirty part of physics".

I thank the members of the Potts group – Aaron Daniel, Marcelo Janovitch, Kacper Prech, Matteo Brunelli, and Prof. Patrick Potts himself – for gently "pushing" the Bruder group into the Rhein. A special thanks goes to Aaron for joining me and guiding me in how to convince kids that quantum mechanics is fun to ponder about.

Thank you also to the founding members of the Physics PhD association, particularly Dominik Koch and Lex Joosten, for their efforts in trying to make the time of PhDs at the department more lively.

Many thanks to our PhD school and its coordinator Thilo Glatzel for the valuable soft-skill courses, to Beat Glatz and Bernd Heimann for their help with IT and technical problems, respectively, to Dominique Zbinden and Tatsiana Dolmat for the administrative support, and to Debora Haller and Hans Syfrig Fongione for our collaboration on the Wissensbox.

Finally, I am deeply grateful to my family and my love Roberta for all their unrestricted support and unwavering trust I received during my doctoral studies.

Publications

Most content presented in this thesis has been published in the following peer-reviewed publications, preprints, and manuscripts:

1. *Machine learning change points in real-world news data*,
   C. Zsolnai, N. Lörch, and **J. Arnold**,
   manuscript in preparation (2025).

2. *Machine learning the Ising transition: A comparison between discriminative and generative approaches*,
   D. Zhang, F. Schäfer, and **J. Arnold**,
   arXiv:2411.19370 (2024).

3. *Phase transitions in the output distribution of large language models*,
   J. Arnold, F. Holtorf, F. Schäfer, and N. Lörch,
   arXiv:2405.17088 (2024).

4. *Mapping out phase diagrams with generative classifiers*,
   J. Arnold, F. Schäfer, A. Edelman, and C. Bruder,
   Phys. Rev. Lett. **132**, 207301 (2024).
   Featured in press releases by MIT and University of Basel.

5. *Machine learning phase transitions: Connections to the Fisher information*,
   J. Arnold, F. Holtorf, N. Lörch, and F. Schäfer,
   arXiv:2311.10710 (2023).

6. *Fast detection of phase transitions with multi-task learning-by-confusion*,
   J. Arnold, F. Schäfer, and N. Lörch,
   NeurIPS 2023 Machine Learning and the Physical Sciences Workshop,
   arXiv:2311.09128 (2023).

7. *Replacing neural networks by optimal analytical predictors for the detection of phase transitions*,
   J. Arnold and F. Schäfer,
   Phys. Rev. X **12**, 031044 (2022).
   Featured in press release by University of Basel.

During the making of this thesis, the author contributed to writing a comprehensive set of lecture notes on applications of machine learning in the quantum sciences that is currently in press as a book at Cambridge University Press:

8. *Modern applications of machine learning in quantum sciences*,
   A. Dawid, **J. Arnold**, B. Requena, A. Gresch *et al.*,
   arXiv:2204.04198 (2022).

In addition, during this time the following articles were published:

9. *Performance bounds for quantum feedback control*,
   F. Holtorf, F. Schäfer, **J. Arnold**, C. Rackauckas, and A. Edelman,
   IEEE Trans. Autom. Control **69**, 8057 (2024).

10. *Quantum simulators, phase transitions, resonant tunneling, and variances: A many-body perspective*,
    A. U. J. Lode, O. E. Alon, **J. Arnold** *et al.*,
    In: Nagel, W.E., Kröner, D.H., Resch, M.M. (eds) High Performance Computing in Science and Engineering '21, HPCSE 2021, Springer (2023).

11. *Combining machine learning and spectroscopy to model reactive atom + diatom collisions*,
    J. C. S. V. Veliz, **J. Arnold**, R. J. Bemish, and M. Meuwly,
    J. Phys. Chem. A **126**, 7971 (2022).

12. *Machine learning product state distributions from initial reactant states for a reactive atom–diatom collision system*,
    J. Arnold, J. C. S. V. Veliz, D. Koner, N. Singh, R. J. Bemish, and M. Meuwly,
    J. Chem. Phys. **156**, 034301 (2022).

Contents

List of Abbreviations

Abbreviation	Meaning	Introduced in
NN	**n**eural **n**etwork	Sec. 1, p. 1
GPU	**g**raphics **p**rocessing **u**nit	Sec. 1, p. 2
PCA	**p**rincipal **c**omponent **a**nalysis	Sec. 1, p. 2
ML	**m**achine **l**earning	Sec. 1, p. 2
SL	**s**upervised **l**earning	Sec. 1, p. 3
LBC	**l**earning **b**y **c**onfusion	Sec. 1, p. 3
PBM	**p**rediction-**b**ased **m**ethod	Sec. 1, p. 3
IGT	**I**sing **g**auge **t**heory	Sec. 2.3.2, p. 13
t-SNE	**t-**distributed **s**tochastic **n**eighbor **e**mbedding	Sec. 2.4, p. 15
CE	**c**ross **e**ntropy	Sec. 2.5.1, p. 18
MSE	**m**ean **s**quared **e**rror	Sec. 2.5.3, p. 21
ReLU	**re**ctified **l**inear **u**nit	Sec. 2.5.4, p. 22
SGD	**s**tochastic **g**radient **d**escent	Sec. 2.5.4, p. 23
CNN	**c**onvolutional **n**eural **n**etwork	Sec. 2.5.4, p. 22
BKT	**B**erezinskii–**K**osterlitz–**T**houless	Sec. 2.6, p. 24
IC-POVM	**i**nformationally-**c**omplete **p**ositive **o**perator-**v**alued **m**easure	Sec. 3.5.2, p. 45
MBL	**m**any–**b**ody **l**ocalization	Sec. 3.6.6, p. 61
CPU	**c**entral **p**rocessing **u**nit	Sec. 3.1, p. 67
SPT	**s**ymmetry-**p**rotected **t**opological	Sec. 4.4.2, p. 90
POVM	**p**ositive **o**perator-**v**alued **m**easure	Sec. 4.4.2, p. 90
DMRG	**d**ensity **m**atrix **r**enormalization **g**roup	Sec. 4.4.2, p. 91
TV	**t**otal **v**ariation	Sec. 5.2, p. 106
KL	**K**ullback-**L**eibler	Sec. 5.2, p. 106
JS	**J**ensen-**S**hannon	Sec. 5.2, p. 106
PAC	**p**robably **a**pproximately **c**orrect	Sec. 6.8, p. 133
LLM	**l**arge **l**anguage **m**odel	Sec. 8.1, p. 145
M	**m**illion	Sec. 8.2.3, p. 153
B	**b**illion	Sec. 8.2.3, p. 153
QKV	**q**uery-**k**ey-**v**alue	Sec. 8.3.3, p. 160
TF-IDF	**t**erm **f**requency-**i**nverse **d**ocument **f**requency	Sec. 9.2.1, p. 167
LM	**l**anguage **m**odel	Sec. 9.2.1, p. 168
LDA	**l**atent **D**irichlet **a**llocation	Sec. 9.2.2, p. 168
API	**a**pplication **p**rogramming **i**nterface	Sec. 9.2.3, p. 170
US	**U**nited **S**tates	Sec. 9.2.3, p. 170
UK	**U**nited **K**ingdom	Sec. 9.2.3, p. 170
GPT	**g**eneral **p**retrained **t**ransformer	Sec. 9.2.3, p. 170
GAN	**g**enerative **a**dversarial **n**etwork	App. G, p. 211

Chapter 1

Introduction

Colloquially, the term *phase transition* refers to a change among the basic phases of matter. For example, in response to changes in external conditions such as temperature or pressure, water can transition to a solid, liquid, or gaseous state. More broadly, in physics, a phase transition refers to an abrupt change in the macroscopic behavior of a large-scale system of interacting constituents [Saitta *et al.*, 2011; Sethna, 2023]. Such phenomena are extremely diverse and phase transitions come in various forms and flavors. Notable examples include transitions in the magnetic properties of materials [Onsager, 1944], transitions from a normal conducting state to a superconductor [Taillefer, 2010], transitions in the entanglement properties of quantum circuit [Lavasani *et al.*, 2021], or the collective motion of active matter such as a flock of birds [Vicsek *et al.*, 1995].

The characterization of phases and transitions between them is a difficult task. This is because the state of the systems to be studied typically lives in a very high-dimensional space. In fact, the number of possible states a system can attain usually grows exponentially with the number of its constituents (such as its particles). Foregoing this problem by studying smaller system instances only partially solves the problem. This is because phase transitions describe changes between distinct *collective* behaviors of *many interacting* particles, leading to lots of microscopic degrees of freedom. Another complication is the fact that the systems to be studied are often probabilistic, meaning that for a given set of values of tunable parameters characterizing the system, such as its temperature or pressure, it may be found in various states.

Physicists typically characterize phases and phase transitions by finding a suitable set of a few low-dimensional quantities, called order parameters [Sethna, 2023], which capture the essence of each phase of the system. For example, even though water is a highly complex system, we can detect the liquid-gas transition by looking at the density which shows a sudden jump at the boiling point and, in this case, serves as an order parameter. However, finding a suitable set of order parameters is “considered an art” [Sethna, 2023], as it requires a great deal of human intuition as well as a prior understanding of the system at hand. Systems often lose some of their symmetrical properties as they transition between phases. When water freezes, for example, its molecules arrange themselves in a regular crystal pattern, breaking the continuous symmetry they had as a liquid. Moreover, nearby particles or components often organize themselves in relation to each other, giving rise to a characteristic local pattern. If the symmetry-breaking pattern is unknown or local order is absent, the identification of an order parameter can be particularly challenging.

On the other hand, in fields such as computer vision, it has been demonstrated that neural networks can be trained to correctly classify intricate sets of labeled images, which naturally live in high dimensions. In particular, in 2012, AlexNet [Krizhevsky *et al.*, 2012] won the ImageNet Large Scale Visual Recognition Challenge, being among the first deep neural nets (NNs) to be trained on a graphics processing unit

(GPU). This represented a turning point in the history of artificial intelligence, in which feature-based methods for image recognition have been gradually replaced with learned neural-net representations. This culminated with ResNet in 2015 [He *et al.*, 2016], the deepest network ever to enter the ImageNet competition winning first place – an event that motivated the use of such techniques to tackle the problem of detecting phase transitions. And indeed, in 2016, it was shown in three seminal works that this is possible: Wang [2016] used principal component analysis (PCA) and NNs were used by Carrasquilla and Melko [2017] as well as Van Nieuwenburg, Liu, and Huber [2017]. Because generic data, such as spin configurations or energy spectra, can be utilized as input to these methods, these proposals opened up a promising route toward the discovery of novel phases of matter and phase transitions with little to no human supervision.

In the years that followed, a variety of other machine learning (ML) schemes for detecting phases and phase transitions have been developed. Classical ML methods have successfully revealed the phase diagrams of a range of systems based on data from numerical simulations [Wang, 2016; Carrasquilla and Melko, 2017; Van Nieuwenburg *et al.*, 2017; Wetzel, 2017; Wetzel and Scherzer, 2017; Ch'ng *et al.*, 2017; Ohtsuki and Ohtsuki, 2017; Schindler *et al.*, 2017; Zhang and Kim, 2017; Broecker *et al.*, 2017; Huembeli *et al.*, 2018; Liu and van Nieuwenburg, 2018; Beach *et al.*, 2018; van Nieuwenburg *et al.*, 2018; Zhang *et al.*, 2018; Venderley *et al.*, 2018; Rodriguez-Nieva and Scheurer, 2019; Huembeli *et al.*, 2019; Schäfer and Lörch, 2019; Scheurer and Slager, 2020; Greplova *et al.*, 2020; Kottmann *et al.*, 2020; Arnold *et al.*, 2021; Huang *et al.*, 2022b; Guo and He, 2023; Maskara *et al.*, 2022; Patel *et al.*, 2022; Zvyagintseva *et al.*, 2022; Sun *et al.*, 2023] as well as experimental measurements [Rem *et al.*, 2019; Käming *et al.*, 2021; Bohrdt *et al.*, 2021; Yu *et al.*, 2022; Miles *et al.*, 2023]. Many of the most powerful ML methods for detecting phase transitions utilize neural networks (NNs) at their core [Carrasquilla and Melko, 2017; Van Nieuwenburg *et al.*, 2017; Wetzel, 2017; Wetzel and Scherzer, 2017; Ch'ng *et al.*, 2017; Ohtsuki and Ohtsuki, 2017; Schindler *et al.*, 2017; Zhang and Kim, 2017; Broecker *et al.*, 2017; Huembeli *et al.*, 2018; Liu and van Nieuwenburg, 2018; Beach *et al.*, 2018; van Nieuwenburg *et al.*, 2018; Zhang *et al.*, 2018; Venderley *et al.*, 2018; Huembeli *et al.*, 2019; Schäfer and Lörch, 2019; Greplova *et al.*, 2020; Kottmann *et al.*, 2020; Zvyagintseva *et al.*, 2022; Arnold *et al.*, 2021; Guo and He, 2023; Patel *et al.*, 2022; Sun *et al.*, 2023; Maskara *et al.*, 2022], similar to how traditional ML is largely dominated by NNs nowadays. NNs are universal function approximators [Cybenko, 1989; Hornik, 1991; Lu *et al.*, 2017; Zhou, 2020] which makes these methods extremely powerful. However, the more expressive an ML model [Bengio and Delalleau, 2011; Goodfellow *et al.*, 2016; Raghu *et al.*, 2017], such as an NN, the more resources are needed to train it, and the more difficult it is to interpret the underlying functional dependence of its predictions on the input [Linardatos *et al.*, 2021; Molnar, 2022]. Therefore, NNs typically act as black boxes that can correctly highlight phase transitions but whose internal workings remain opaque to the user.

Since the proposal of these methods, there have been numerous attempts to understand their working principle, particularly through the extraction of order parameters. As an example, (kernel) support vector machines, which are easier to analyze than NNs due to their inherent linear nature, were used as predictive models [Ponte and Melko, 2017; Zhang *et al.*, 2019a; Greitemann *et al.*, 2019a; Liu *et al.*, 2019]. Other approaches to improve interpretability rely on systematic input engineering, such that the objective function that the NN learns is approximately linearly [Zhang *et al.*, 2020], or on a systematic reduction of the NN expressivity [Wetzel and Scherzer, 2017]. Another set of works [Casert *et al.*, 2019; Dawid *et al.*, 2020; Blücher *et al.*, 2020; Dawid *et al.*, 2021] analyzed trained NNs using standard interpretability tools from ML, which rely on truncated Taylor expansions. Despite these efforts, at the time of starting this thesis, we understood little about the working principle of ML methods for the detection of phase transitions based on NNs, when they fail or succeed, and how they differ [Carleo *et al.*, 2019] – in particular when deep NNs are used (i.e., in the limit of high model expressivity). These open questions reflected the general scarcity of rigorous ML theory in the field [Huang *et al.*, 2022b].

In the first part of this thesis, we will address these open challenges by developing a deeper understanding of NN-based methods for detecting phase transitions from a probabilistic perspective. This understanding will lead to both conceptual and computational improvements of these techniques as well as fundamental insights into their capabilities and limits. We will showcase applications to both classical and quantum many-body systems, and discuss the viability of ML methods for detecting phase transitions in experimental settings.

Studying how these methods perform on prototypical models in physics is crucial, as their phase diagrams are well-known and can serve as a ground truth for benchmarking and making appropriate modifications. Having refined the methods and cast them into a probabilistic framework, in the second part of this thesis, we will venture beyond detecting phase transitions in the traditional physics setting and showcase the ability of these methods to detect phase-transition-like phenomena in previously unchartered domains. This includes transitions in state-of-the-art generative models for images (diffusion models) and text (language models), as well as real-world news data. While the first part of this thesis (comprised of Chapters 2-7) falls under the umbrella of *ML for Physics*, we may classify this second part (comprised of Chapters 8-9) as *Physics for ML*. In contrast to the model systems studied in the first part of the thesis, there typically does not exist any prior consensus on the phase diagrams of the systems studied in the second part. This fact makes these applications particularly challenging and highlights the ability of our methods to discover transitions in systems with little to no prior knowledge – an exciting capability in the advent of the collective behavior of artificial neurons exhibiting signs of intelligence.

Overview of this thesis

Part I: Learning to Detect Phase Transitions in Physical Systems

Chapter 2 This chapter lays out the theoretical and contextual background for the rest of this thesis. We start by stating the problem setup for detecting phase transitions from data more formally. We discuss how physicists typically approach the detection of phase transitions and how ML may be used to help. This leads us to a brief historical account of how the field of "machine learning phase transitions" started and how it evolved. We introduce three of the most popular NN-based methods for detecting phase transitions from data in detail: so-called *supervised learning* (SL), *learning by confusion* (LBC), and the *prediction-based method* (PBM). These are the methods we will be most concerned with throughout this thesis. After demonstrating their application on two prototypical model systems – the classical ferromagnetic Ising model on a square lattice as well as the Ising gauge theory – we summarize the state of the field at the time of starting this thesis in August 2021, including some of the open issues that will be tackled in the subsequent chapters, particularly Chapter 3. This includes questions such as how the capacity of the employed

NN model influences the results or how the computational efficiency of these methods may be improved.

Chapter 3 In Chapter 3, we start to tackle these questions by deriving analytical expressions for the optimal output of SL, LBC, and PBM. Here, optimality refers to the fact that the corresponding predictive models minimize the relevant loss function. As such, in practice, optimal predictions can, for example, be approximated using sufficiently large, well-trained NNs. The core contributions of this chapter are two-fold, enhancing both our conceptual understanding of the methods and yielding improved computational procedures. On the conceptual side, the inner workings of the considered methods are revealed through the explicit dependence of the optimal output on the input data. Most notably, we find that the methods inherently rely on detecting changes in the probability distributions governing the input data. This key insight will be further expanded upon and made rigorous in Chapter 6. On the computational side, given access to the probability distributions underlying the system, we can identify phase transitions directly without training NNs by evaluating the analytical expressions. This forms the basis of a novel procedure for detecting phase transitions from data that is favorable in terms of computation time. This procedure will be refined and generalized in Chapter 4. Our theoretical results are supported by extensive numerical simulations covering, e.g., topological, quantum, and many-body localization phase transitions.

Chapter 3 is largely based on:

Replacing neural networks by optimal analytical predictors for the detection of phase transitions, J. Arnold and F. Schäfer, Phys. Rev. X **12**, 031044 (2022).

Chapter 4 Classification problems are typically tackled using *discriminative classifiers* that explicitly model the probability of the labels for a given sample. This corresponds to the traditional way of applying NNs in SL, LBC, and PBM to detect phase transitions. In this chapter, we show that the classification problems arising in these methods are naturally suited to be solved using so-called *generative classifiers* based on probabilistic models of the measurement statistics underlying the physical system. This generalizes the approach from Chapter 3 to any probabilistic model with an explicit tractable density, allowing us to tackle larger system sizes. Moreover, we formulate the methods in a fully probabilistic manner. This brings to light hidden assumptions on prior distributions, allowing for conceptual method improvements. Finally, the methods are also extended to work in higher-dimensional parameter spaces. We showcase the power of these modifications in applications to classical equilibrium systems and quantum ground states.

Chapter 4 is largely based on:

Mapping out phase diagrams with generative classifiers, J. Arnold, F. Schäfer, A. Edelman, and C. Bruder, Phys. Rev. Lett. **132**, 207301 (2024).

Chapter 5 Having formulated the methods in a fully probabilistic fashion, it is time to review statistical concepts to make further progress. In Chapter 5, we introduce the notion of a statistical distance and discuss the role of such distance in information geometry as well as the statistical tasks of hypothesis testing and parameter estimation that underly SL, LBC, and PBM. This background forms the basis for our analysis in Chapter 6.

> Chapter 5 is largely based on:
>
> *Machine learning phase transitions: Connections to the Fisher information*, J. Arnold, F. Holtorf, N. Lörch, and F. Schäfer, arXiv:2311.10710 (2023).

Chapter 6 The statistical background of Chapter 5 allows us to explain the inner workings and identify potential failure modes of ML methods for detecting phase transitions by rooting them in information-theoretic concepts. Using tools from information geometry, we prove that the indicators of phase transitions of SL, LBC, and PBM approximate the square root of the system's (quantum) Fisher information from below – a quantity that is known to indicate phase transitions but is often difficult to compute from data. We numerically demonstrate the quality of these bounds for phase transitions in classical and quantum systems.

> Chapter 6 is largely based on:
>
> *Machine learning phase transitions: Connections to the Fisher information*, J. Arnold, F. Holtorf, N. Lörch, and F. Schäfer, arXiv:2311.10710 (2023).

Chapter 7 Up to now, the discriminative version of LBC required training a distinct binary classifier for each possible splitting of the set of sampled values of a tunable parameter (such as temperature or pressure) into two sides, resulting in a computational cost that scales linearly with the number of grid points. In this chapter, we propose and showcase an alternative implementation that only requires the training of a *single* multi-class classifier. Ideally, such multi-task learning eliminates the scaling with respect to the number of grid points. In practice, we find significant speedups that, apart from small deviations, correspond to this ideal case.

We also take our first footstep outside the application domain of physics by showcasing that learning by confusion can detect changes in images created by the popular text-to-image generative model *Stable Diffusion*. The tuning parameter corresponds to an integer in the generation prompt. To this end, we utilize the multi-tasking approach proposed in Chapter 7, further underpinning its computational advantage.

> Chapter 7 is largely based on:
>
> *Fast detection of phase transitions with multi-task learning-by-confusion*, J. Arnold, F. Schäfer, and N. Lörch, NeurIPS 2023 Machine Learning and the Physical Sciences Workshop, arXiv:2311.09128 (2023).

Part II: Venturing Beyond Physics

Chapter 8 In this chapter, we boldly venture beyond the lands of physics and study transitions in the behavior of large language models. To this end, we quantify distributional changes in the generated output via statistical distances, which can be efficiently estimated with access to the probability distribution over next-tokens. This is akin to the generative approach to LBC introduced in Chapter 4. We find various phase-transition-like phenomena occurring as a function of three different control parameters in Pythia, Mistral, and Llama language models: an integer in the input prompt, the temperature hyperparameter for text generation, and the model's training epoch.

Chapter 8 is largely based on:

Phase transitions in the output distribution of large language models, J. Arnold, F. Holtorf, F. Schäfer, and N. Lörch, arXiv:2405.17088 (2024).

Chapter 9 We conclude our adventure by showcasing the ability of the LBC method to detect rapid changes in news articles from *The Guardian* – a British daily newspaper – over time. We show that significant events, such as the September 11 attacks in 2001 or the outbreak of the Coronavirus pandemic, manifest themselves as phase-transition-like phenomena (i.e., *change points*) in the news that can be effectively detected using our NN-based method.

Chapter 9 is largely based on:

Machine learning change points in real-world news data, C. Zsolnai, N. Lörch, and J. Arnold, manuscript in preparation (2025).

Part III: Conclusion and Appendices

We conclude this thesis in Chapter 10. Additional material supporting the findings presented in the first two parts of this thesis can be found in Appendices A-H.

Part I

Learning to Detect Phase Transitions in Physical Systems

Chapter 2

Theoretical Background

The discussion of the Ising model and Ising gauge theory as well as the description of the learning schemes presented in this chapter are compiled from the following publication:

Replacing neural networks by optimal analytical predictors for the detection of phase transitions,
J. Arnold and F. Schäfer,
Phys. Rev. X **12**, 031044 (2022).

2.1 The problem setup

Let us start by formally introducing the task of detecting phase transitions from data, i.e., let us state what is given to us and what we want a potential algorithm to accomplish, see Figure 2.1 for a schematic illustration. In the most general case, we assume our physical system to be characterized by a vector $\boldsymbol{\gamma} \in \mathbb{R}^d$ of tuning parameters.[1] These could, for example, be external parameters such as temperature or pressure, or internal parameters, such as coupling strengths. This defines the parameter space in which we want to characterize the phases of our system and detect transitions between them. We are investigating the system at a discrete sampled set of tuning parameter values Γ, where $|\Gamma|$ denotes the number of sampled points.

In general, we only have a description of a system with tunable parameters $\boldsymbol{\gamma}$ in terms of an observed state $\boldsymbol{x} \in \mathcal{X}$. Here, $\mathcal{X}$ denotes the relevant state space, i.e., the space of all possible observable states. In this thesis, we only deal with state spaces that are discrete and countable. The system is thus characterized by a dataset $\mathcal{D}_{\boldsymbol{\gamma}}$ of observed states at each sampled point $\boldsymbol{\gamma} \in \Gamma$. Throughout this thesis, we will refer to collections of observed data as (data)sets, as is commonly done in the ML literature. Strictly speaking, these are *multisets* given that they may contain duplicates. We will distinguish a set obtained by removing duplicates from its multiset with $\bar{\cdot}$. We are going to refer to the set of data across all sampled points as $\mathcal{D} = \biguplus_{\gamma \in \Gamma} \mathcal{D}_{\gamma}$, where $\uplus$ denotes the multiset union (which takes multiplicities into account).[2] In the context of physics, we can view these observations as a result of probing the system by performing a measurement. Note that this description does allow for the possibility of measurement noise, i.e., the observed state $\boldsymbol{x}$ of the system may be composed of its "true state" plus some additional noise.

Based on the datasets $\{\mathcal{D}_{\boldsymbol{\gamma}}\}_{\boldsymbol{\gamma} \in \Gamma}$ and knowledge of Γ, we want to determine the location of all critical points (i.e., phase boundaries) in the observed region of the

[1] Throughout this thesis, we will deal with $d \leq 2$. This is mostly for reasons of visualization.

[2] The following example illustrates the difference between the simple union (denoted by $\cup$) and the multiset union (also called additive union): $\{1, 2\} \cup \{2, 3\} = \{1, 2, 3\}$ whereas $\{1, 2\} \uplus \{2, 3\} = \{1, 2, 2, 3\}$.

parameter space. Errors due to the finite resolution of the parameter space through sampling are inevitable. Instead, we aim to obtain a close approximation of the true phase boundaries across the sampled parameter values Γ.

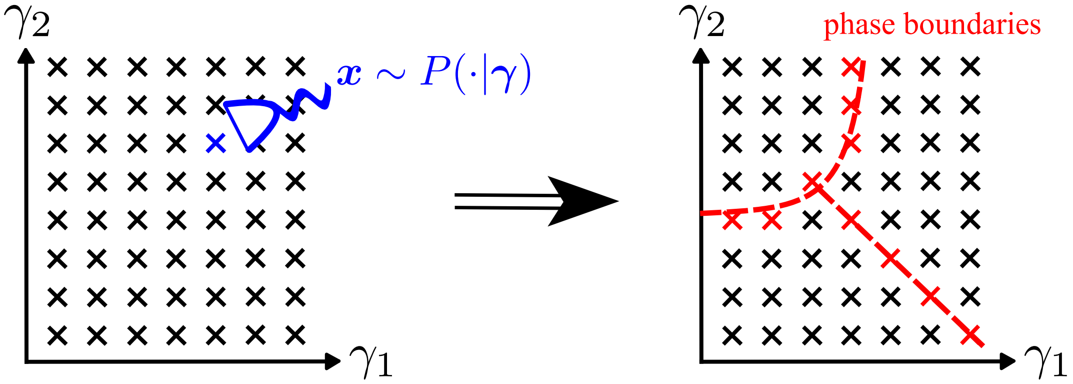

FIGURE 2.1: Schematic illustration of the setup for detecting phase transitions from data. We are given measurements of the system's state $\boldsymbol{x} \sim P(\cdot|\boldsymbol{\gamma})$ at various values of the tuning parameter $\boldsymbol{\gamma} \in \Gamma$ (denoted by the crosses). Based on this data, we would like to determine the location of the phase boundaries (i.e., critical points) within the system. Here, the true phase boundaries are denoted by dashed lines which we are only able to determine up to the closest sampled point in parameter space (crosses). Here, the parameter space is two-dimensional and hosts three distinct phases.

Without loss of generality, we may assume that when observing (i.e., measuring) a system described by the tunable parameters $\boldsymbol{\gamma}$, we find it in state $\boldsymbol{x} \in \mathcal{X}$ with probability $P(\boldsymbol{x}|\boldsymbol{\gamma})$. That is, we can assume the dataset $\mathcal{D}_{\boldsymbol{\gamma}}$ at a given point $\boldsymbol{\gamma}$ to be composed of independent and identically distributed observations. We will make this assumption throughout this thesis. In Chapters 3 and 4 we will explore how we can determine critical points and phase boundaries more efficiently given access to the underlying probability distributions $\{P(\cdot|\boldsymbol{\gamma})\}_{\boldsymbol{\gamma}\in\Gamma}$.[3]

2.2 A physicist's approach

One way a physicist would traditionally approach this problem is by coming up with a function that takes as input the collection of measurement results across the parameter space and spits out a low-dimensional quantity that distinguishes one phase from another. This could, for example, be a quantity that is non-zero for all parameter values within a selected phase and zero otherwise, i.e., what physicists refer to as an *order parameter*. The key characteristic of such a quantity is that it takes on a unique set of values within and outside a given phase, i.e., it characterizes the phase.

[3] Throughout this thesis, we generally use $P(\cdot)$ or P to denote probability distributions whereas $P(\boldsymbol{x})$ refers to the probability associated with a specific sample $\boldsymbol{x}$. When dealing with distributions over different sample spaces, however, we may include the argument even when referring to the distribution itself to make explicit the sample space over which the distribution is defined. Distributions over energies E or spin configurations $\boldsymbol{\sigma}$, for example, may then be denoted as $P(E)$ or $P(\boldsymbol{\sigma})$, respectively. Whether we refer to the probability of a specific sample or the entire distribution is to be inferred from the context. We also use this convention for other functions.

Another possibility would be to come up with quantities that instead take on characteristic values in the presence of a phase transition. In physics, *response functions*, such as the heat capacity or magnetic susceptibility, for example, highlight the presence of a phase transition through their divergence. Note that given an order parameter, one can typically derive a quantity of the latter type by taking appropriate derivatives. The "magic" lies in coming up with an appropriate function of one of the two types. Up to now, this required a great deal of human intuition as well as an extensive prior understanding of the system at hand.

The above description of the problem of detecting phase transitions as a search for appropriate functions suggests we could develop an algorithm that *learns* the appropriate function from the data, rather than having physicists determine the appropriate function manually on a case-by-case basis. This is where ML comes into play. We will continue this discussion in Section 2.4.

2.3 Two prototypical problem instances

Before we dive into how ML can be used to detect phase transitions, let us review some concrete examples of prototypical instances of the phase-transition-detection problem in physics. This will clarify the physicist's approach and give us further hints on how ML may help.

2.3.1 Ising model

The classical square-lattice ferromagnetic Ising model is described by the following Hamiltonian

$$H(\boldsymbol{\sigma}) = -J \sum_{\langle ij \rangle} \sigma_i \sigma_j, \tag{2.1}$$

where the sum runs over all nearest-neighboring sites (with periodic boundary conditions) and J is the interaction strength ($J > 0$). At each lattice site i, there is a discrete spin variable $\sigma_i \in \{+1, -1\}$. This results in a state space $\mathcal{X}$ of size $2^{L\times L}$ for a square lattice of linear size L. The system is completely characterized by its spin configuration $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \ldots, \sigma_{L\times L})$. Assuming that the system is in thermal equilibrium at temperature T, the probability of finding the system in the state $\boldsymbol{\sigma}$ is given by the Boltzmann distribution

$$P(\boldsymbol{\sigma}|T) = \frac{e^{-\beta H(\boldsymbol{\sigma})}}{Z_T}, \tag{2.2}$$

where $Z_T = \sum_{\boldsymbol{\sigma}\in\mathcal{X}} e^{-\beta H(\boldsymbol{\sigma})}$ is the partition function, the sum runs over all possible spin configurations, and $\beta = 1/k_{\mathrm{B}}T$ is the inverse temperature with k_{B} denoting Boltzmann's constant.

To sample spin configurations from the thermal distribution at a given temperature T, we use the Metropolis-Hastings algorithm [Metropolis *et al.*, 1953]. The lattice is initialized in a state with all spins pointing up. This breaks the $\mathbb{Z}_2$ symmetry as this is one of the two ground states. We update the lattice by drawing a random spin, which is flipped with probability $\min\{1, e^{-\Delta E/k_{\mathrm{B}}T}\}$, where ΔE is the energy difference resulting from the considered flip. To ensure that the system is sufficiently thermalized, we sweep the complete lattice 10^5 times[4], where each lattice site is updated once per sweep. After the thermalization period, which we find to be sufficient for achieving convergence, we collect samples. We start at a low temperature and increase the temperature gradually.

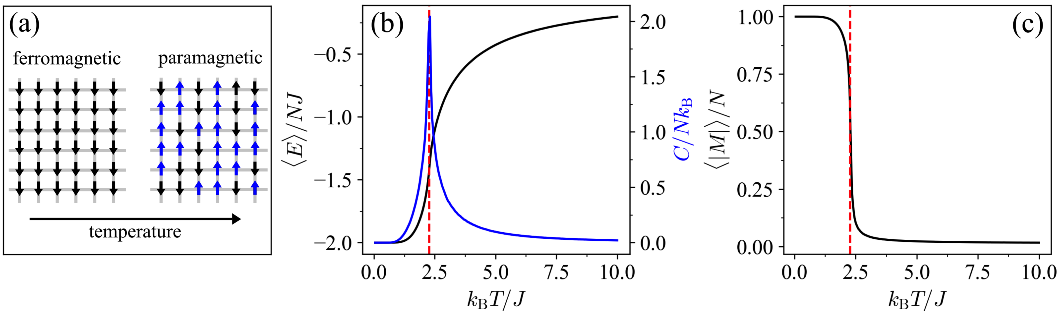

Figure 2.2: (a) Illustration of the symmetry-breaking phase transition in the Ising model. (b) Average energy per site (black) and associated heat capacity (blue) as a function of temperature. (c) Average magnetization per site as a function of temperature. These results were obtained for a lattice with linear size $L = 60$ ($N = L^2$ denotes the number of spins) and 10^5 sampled spin configurations per tuning parameter value.

Two example spin configurations of the Ising model at different temperatures are shown in Figure 2.2(a). The Ising model exhibits a symmetry-breaking phase transition between a paramagnetic (disordered) phase at high temperature and a ferromagnetic (ordered) phase at low temperature, where the critical temperature is [Onsager, 1944]

$$T_c = \frac{2J}{k_B \ln(1+\sqrt{2})}. \tag{2.3}$$

Spontaneous magnetization occurs below the critical temperature T_c, where the interaction is sufficiently strong to cause neighboring spins to align. This spontaneous symmetry breaking leads to a non-zero mean magnetization. Above T_c, thermal fluctuations dominate over spin alignment resulting in a vanishing magnetization. Consequently, the phase transition can be characterized by the magnetization $M(\boldsymbol{\sigma}) = \sum_{i=1}^{L^2} \sigma_i$ which serves as an order parameter that is zero within the paramagnetic phase and approaches one in the ferromagnetic phase, see Figure 2.2(b). The phase transition can also be revealed by the heat capacity

$$C(T) = \frac{d\langle E\rangle_T}{dT} = \frac{\langle E^2\rangle_T - \langle E\rangle_T^2}{k_B T^2} \tag{2.4}$$

which diverges at T_c [see Figure 2.2(c)]. Here, we use $\langle\cdot\rangle_T$ to denote expected values of an observable $O(\boldsymbol{\sigma})$ at temperature T:

$$\langle O\rangle_T = \mathbb{E}_{\boldsymbol{\sigma}\sim P(\cdot|T)}\left[O(\boldsymbol{\sigma})\right] = \sum_{\boldsymbol{\sigma}\in\mathcal{X}} P(\boldsymbol{\sigma}|T)O(\boldsymbol{\sigma}). \tag{2.5}$$

The second equality in Equation (2.4) follows from writing out the expected value according to Equation (2.5) with the Boltzmann distribution [Equation (2.2)] and

[4] In this thesis, we analyze the Ising model on lattices with linear size of at maximum $L = 60$.

carrying out the corresponding derivative.

2.3.2 Ising gauge theory

Wegner's Ising gauge theory (IGT) [Wegner, 1971] is described by the following Hamiltonian

$$H(\boldsymbol{\sigma}) = -J \sum_{\mathbb{P}} \prod_{i \in \mathbb{P}} \sigma_i, \tag{2.6}$$

where $\mathbb{P}$ refers to plaquettes on the lattice, see Fig 2.3(a). It is a spin model ($\sigma_i \in \{+1, -1\}$) defined on a square lattice of linear size L (with periodic boundary conditions) where the spins are placed on the lattice bonds [see Figure 2.3(a)]. As for the Ising model (Section 2.3.1), we use the Metropolis-Hastings algorithm [Metropolis *et al.*, 1953] to draw spin configurations from the Boltzmann distribution at various temperatures.[5] In the case of the IGT, the lattice is initialized in a random spin configuration.

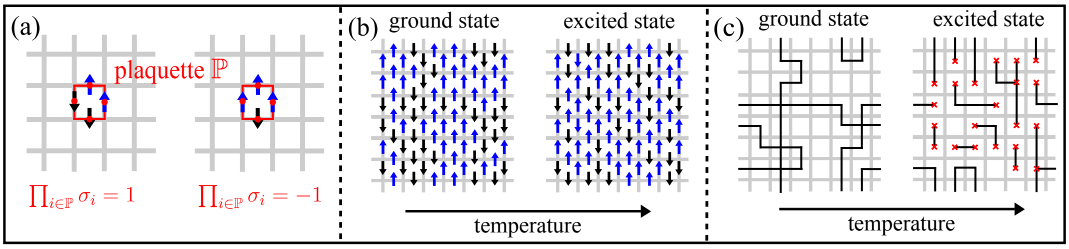

FIGURE 2.3: Schematic illustration of the topological crossover in the IGT as a function of temperature. (a) Examples of plaquettes $\mathbb{P}$ where the topological constraint is met ($\prod_{i\in\mathbb{P}} \sigma_i = 1$) and violated ($\prod_{i\in\mathbb{P}} \sigma_i = -1$). (b) Examples of spin configurations within the topological ground-state phase (left) and phase with violated topological constraints at high temperature (right) and (c) their corresponding Wilson loops.

In the Landau paradigm of phase transitions [Landau, 1937a,b], transitions between phases of matter are intrinsically linked to changes in underlying symmetries. The transition in the Ising model (Section 2.3.1), for example, is a symmetry-breaking transition. In this case, the broken symmetry is $\mathbb{Z}_2$, which refers to the invariance of the system's Hamiltonian under flipping all spins simultaneously. At high temperatures, the Ising model has disordered spins with no net magnetization, preserving the $\mathbb{Z}_2$ symmetry. However, below the critical temperature, the system undergoes a phase transition to a magnetized phase where the spins jointly align in one of the two directions, breaking the $\mathbb{Z}_2$ symmetry. This spontaneous symmetry breaking leads to a non-zero magnetization, marking the ordered phase. As a consequence, the magnetization serves as a local order parameter of the model.

Landau's theory does, however, fail to account for topological phases of matter [Wen, 1990]. The IGT is a prototypical example of a classical system that exhibits a topological phase of matter [Kogut, 1979]. The ground state of the IGT is a degenerate manifold made up of all states which fulfill the condition that the product of spins on each plaquette is $\prod_{i\in\mathbb{P}} \sigma_i = 1$ corresponding to a topological phase. These topological constraints can be violated at finite temperatures, where the system

[5] In this thesis, we analyze the IGT on lattices with maximum $L = 28$.

leaves its ground state. Note that there is no phase transition at finite temperature: the critical temperature approaches zero in the thermodynamic limit. In finite-sized systems, however, the violations of local constraints are suppressed. Therefore, the system exhibits a crossover from the topological phase at low temperature to a phase with violated topological constraints at high temperature. The crossover temperature T_c is defined by the first appearance of a violated local constraint and scales as $T_\mathrm{c} \propto 1/\ln(2L^2)$ [Castelnovo and Chamon, 2007].

The topological character of the ground-state phase can be revealed through Wilson loops. These are formed by connecting edges with spins of the same orientation, see Figures 2.3(b) and (c). In the ground-state phase, all such loops are closed. The violation of a plaquette constraint breaks a loop. Figure 2.3(b), which shows typical spin configurations of the IGT, highlights that the phases of the IGT are hard to distinguish visually without prior knowledge of the local constraints or a dual representation [Carrasquilla and Melko, 2017; Greplova *et al.*, 2020]. This reflects the more general fact that, *a priori*, systems characterized by nonlocal and long-range correlations represent a challenge for physicists and machines alike trying to detect phase transitions.

2.4 How can machine learning help?

ML methods for detecting phase transitions largely emerged by adapting techniques from the vast zoo of general ML approaches, utilized, for example, in computer vision, and tailoring them to the task at hand. Without trying to give an accurate account of the entire history of this field, let us venture back into the past. One of the earliest foundational works of the field can be attributed to Wang [2016] who showed that the phase transition in the prototypical Ising model (Section 2.3.1) can be identified by mapping its spin configurations to a low-dimensional space spanned by its first few principal components computed via principal component analysis (PCA), see Figure 2.4(a). The resulting points may then be clustered into three distinct groups using k-means (two for the ordered phase corresponding to the two distinct magnetization directions and one for the disordered phase). In fact, the first principal component indeed corresponds to the magnetization, i.e., an order parameter of the Ising model.

However, PCA performs a linear projection of the data and relies on the Euclidean distance as a notion of similarity between spin configurations. While this suffices to discover simple order parameters, such as the magnetization in the case of the Ising model, it can fail in more complicated scenarios. In particular, data often resides on a lower-dimensional manifold within the original space that cannot necessarily be parametrized by linear transformations of the original coordinates. In such cases, a dimensionality reduction using PCA does not preserve the relative pairwise distance (or similarity) between data points. For example, PCA fails to separate the topological trivial and topological non-trivial phases of the IGT [Carrasquilla and Melko, 2017; Dawid *et al.*, 2022], see Figure 2.4(b). Recall that the topological phase of the IGT is characterized by all products of spins on plaquettes resulting in $+1$. Products of spins, however, cannot be attained by performing a linear transformation and projection of a raw spin configuration.

Consequently, the focus shifted toward nonlinear methods to construct representations of the data that are in turn linearly separable, i.e., representations in which the data naturally separates itself into its distinct phases. Examples are nonlinear dimensionality reduction techniques, such as t-distributed stochastic neighbor embedding

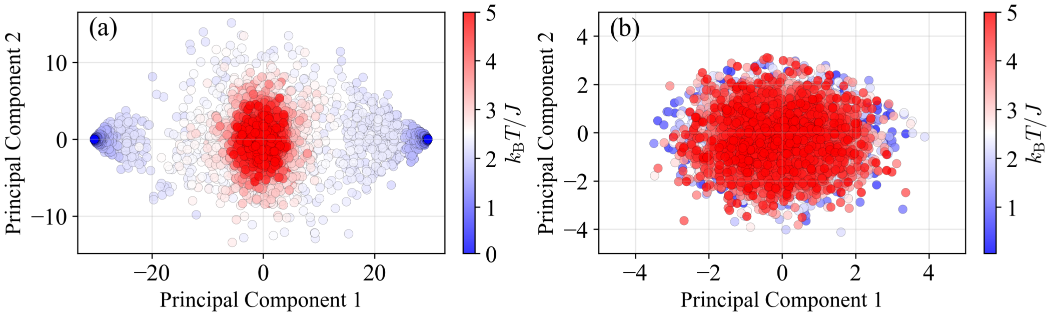

FIGURE 2.4: Results of PCA being applied to spin configurations of (a) the Ising model ($L = 30$) and (b) the IGT ($L = 16$). We sampled 1000 spin configurations per temperature on a temperature grid ranging from $k_{\mathrm{B}}T/J = 0.05$ to $k_{\mathrm{B}}T/J = 5.0$ in steps of 0.05. In the Ising model, we explicitly include spin configurations of two independent Markov chains starting in one of the two ground states. This results in two clusters at low temperatures.

(t-SNE) [Wetzel, 2017], as well as kernel methods [Ponte and Melko, 2017], and NNs more generally [Carrasquilla and Melko, 2017; Van Nieuwenburg *et al.*, 2017; Wetzel, 2017]. The question then becomes how we can learn useful representations for the data: Can we come up with an algorithm that yields such a representation for novel systems for which we do not have much prior knowledge? Put differently, having a way to express nonlinear functions, e.g., using neural networks as universal function approximations, how do we find the right nonlinear function?

Taking PCA as a starting point, one branch of research continued as follows: PCA can be understood as finding the linear transformation that minimizes the reconstruction error when mapping between the original space and the one of reduced dimensionality spanned by the principal components (see Appendix A in [Dawid *et al.*, 2022]). Autoencoders – a special class of neural network architectures – can be viewed to generalize PCA in the sense that they learn a nonlinear mapping between the original space and a lower-dimensional latent space by minimizing the reconstruction error. And indeed, the latent space of trained autoencoders does encode information about the phases of a system [Wetzel, 2017], i.e., it separates configurations belonging to distinct phases into distinct clusters. It turns out that one may also use the reconstruction error itself as a signal that highlights distinct phases: one can train the autoencoder within a phase and apply it across a larger range of the parameter space. The reconstruction error will assume larger (and possibly distinct) values in the phases beyond the original phase the autoencoder was trained in [Kottmann *et al.*, 2020]. This takes inspiration from the task of "anomaly detection": inputs belonging to a phase distinct from the one in which the autoencoder was trained, are detected as anomalies.

More generally, ML methods for detecting phase transitions often leverage the fact that when a machine tries to learn to perform a prediction task based on data that is subject to a phase transition, this sudden change in the high-dimensional input will also manifest itself as a sudden change in a low-dimensional *bottleneck*. Following Dawid *et al.* [2022], here we adopt the loose definition of a bottleneck as a low-dimensional set of quantities that emerges within or at the end of an ML pipeline.

The learning task and the bottleneck may differ from method to method and researchers have investigated all kinds of learning tasks and predictive models known in traditional ML to detect transitions. In the example mentioned in the previous paragraph, the learning task is a reconstruction task, the predictive model is an autoencoder, and the bottleneck is either the latent space of the autoencoder [Wetzel, 2017] or the reconstruction loss itself [Kottmann *et al.*, 2020]. One may also view PCA as a variant of this learning scheme where the learning task is a reconstruction task, the bottleneck corresponds to the subspace spanned by the principal components, and the predictive model is linear.

One of the oldest and arguably most successful classes of methods focuses on classification and regression tasks, where the output, i.e., the predicted variable, or some function thereof (such as the loss) takes the role of the low-dimensional bottleneck. Throughout the remainder of this thesis, this will be the class of methods we are most concerned about. The most prominent methods belonging to this class are the so-called supervised learning (SL) method originally proposed by Carrasquilla and Melko [2017], the learning-by-confusion (LBC) method originally proposed by Van Nieuwenburg, Liu, and Huber [2017], and the prediction-based method (PBM) originally proposed by Schäfer and Lörch [2019].

All three methods feature a bottleneck that in turn gives rise to a "response function", highlighting the presence of a phase transition. The workflow for computing this function is similar for all three methods as illustrated in Figure 2.5. The methods take as input samples that represent the state of a physical system at various values of a tuning parameter. The samples are, in general, processed by an NN whose parameters are tuned to minimize a specific loss function. However, nothing prevents these schemes from being applied with other predictive models provided they can solve the task at hand somewhat well – similar to how a reconstruction task may be tackled using a nonlinear model, such as an autoencoder, or a linear model, as in the case of PCA. By analyzing the predictions, one can compute a scalar quantity that assumes large values at a critical value of the tuning parameter at which the system's state changes most. As such, this quantity highlights phase boundaries and serves as an indicator for phase transitions. This is the "response function" alluded to above. The three methods differ in their choice of loss function, i.e., in the formulation of the underlying classification or regression task, and thus in the resulting indicator for phase transitions.

2.5 Casting the detection of phase transitions as a prediction task

To introduce these methods in their simplest form we will simplify the problem setup outlined in Section 2.1, see Figure 2.5. We will consider the physical system to be characterized by a single tuning parameter γ sampled equidistantly with a grid spacing $\Delta\gamma$ such that $\Gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_K\}$.[6] In the following, we denote the points at the boundary of the sampled region as γ_1 and γ_K with $K \in \mathbb{N}$ sampled points in total ($K = \frac{\gamma_K - \gamma_1}{\Delta\gamma} + 1$). At each sampled point $\gamma \in \Gamma$ we draw the same number of samples from the system's state. This constitutes our available data. At the core of each of the three methods for detecting phase transitions under consideration lies a predictive

[6] The assumption of an equidistant sampling grid is not essential and is made here for convenience, particularly for handling numerical derivatives. The methods may be straightforwardly applied to parameter spaces that are not sampled equidistantly.

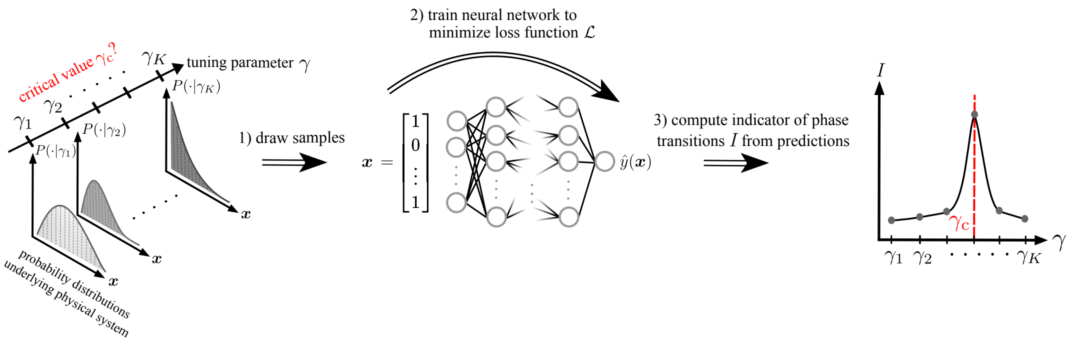
critical value γ_c?
γ_1
γ_2
γ_K
tuning parameter γ
$P(\cdot|\gamma_1)$
$P(\cdot|\gamma_2)$
$P(\cdot|\gamma_K)$
x
probability distributions underlying physical system
1) draw samples
2) train neural network to minimize loss function $\mathcal{L}$
$x =$ 1 0 ⋮ 1
$\hat{y}(x)$
3) compute indicator of phase transitions I from predictions
I
γ_c
γ_1 γ_2 γ_K γ

FIGURE 2.5: Schematic representation of the setup and workflow of SL, LBC, and PBM for detecting phase transitions from data using NNs in a one-dimensional parameter space. The goal is to identify the critical value of the tuning parameter γ_c at which the system transitions from one phase to another. In a first step (step 1), the state $\boldsymbol{x}$ of the physical system is (repeatedly) sampled at various values of the tuning parameter $\Gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_K\}$, where $\{P(\cdot|\gamma_1), P(\cdot|\gamma_2), \ldots, P(\cdot|\gamma_K)\}$ are the corresponding probability distributions. Based on these samples, an NN is trained to perform a particular classification or regression task, i.e., its tunable parameters are updated to minimize a particular loss function (step 2). The three ML methods for detecting phase transitions differ in their formulation of the underlying NN tasks. Having trained the NN, its predictions $\hat{y}$ are used to compute the value of an indicator of phase transitions I at fixed values of the tuning parameter (step 3). Ideally, the indicator has a local maximum at γ_c where the largest change in the state of the system occurs. As a result, the ML methods then autonomously highlight phase boundaries along the chosen scanning range of the tuning parameter.

model $m : \mathcal{X} \to \mathbb{R}$, such as an NN. In the following, we may also refer to the predictive model as $\hat{y}$ or $\hat{\gamma}$ if it tries to learn the label y or γ of a given sample $\boldsymbol{x}$.

We assume that the system is present either in a single phase A or two distinct phases A and B across the sampled range of the tuning parameter. The task is then to compute a scalar indicator $I(\gamma)$ for all $\gamma \in \Gamma$ that peaks at the phase boundary if two distinct phases are present, i.e., has a local maximum, and does not exhibit a peak otherwise. More specifically, if the system is in phase A from γ_1 to γ_c and phase B from γ_c to γ_K with critical point γ_c (not necessarily a sampled point), the indicator I should exhibit a local maximum at the sampled point closest to the critical point $\operatorname{argmin}_{\gamma \in \Gamma} |\gamma_c - \gamma|$.

We will tackle the problem as outlined in Section 2.1 in its full generality in Chapter 4. This will require introducing appropriate generalizations of the three ML methods. In particular, while PBM applies to high-dimensional parameter spaces in its original form, SL and LBC have originally been proposed with one-dimensional parameter spaces in mind. Similarly, while PBM can be applied to parameter spaces featuring more than two distinct phases in its original form, SL and LBC have originally been showcased only for systems featuring one or two phases. We will also discuss how to handle instances where a distinct number of samples is drawn per parameter value.

Splitting the data into various sets

As we will see later, it may be helpful to split the available data $\mathcal{D}_\gamma$ at each point $\gamma \in \Gamma$ into different sets when trying to construct a predictive model. In general, we may consider three distinct sets: a training, a validation, and an evaluation (i.e., a test) set. We refer to the training set as the dataset that is explicitly used to train the predictive model. The training set is, for example, used to compute the gradient signal when using gradient descent to train a model. The validation set is used as a proxy to evaluate the performance of the model on unseen data and to avoid overfitting, for example, by adjusting model hyperparameters such as the number of training epochs accordingly. The evaluation set (or test set) is the data that is used to evaluate the final predictive model and to obtain the indicator I. We denote the training set at point γ by $\mathcal{T}_\gamma$ and the overall training set as $\mathcal{T} = \biguplus_{\gamma\in\Gamma}\mathcal{T}_\gamma$. Similarly, we will refer to the validation set by $\mathcal{V}$ and the evaluation set (or test set) by $\mathcal{E}$. Note that the training, validation, and test set may not add up to the overall dataset given that the test set can contain data points that are also present in the training or validation set. The data points in the training and validation set, however, are distinct. In the following, we will assume that each non-empty set per sampled parameter value contains the same number of data points. For example, $|\mathcal{T}_{\gamma_1}| = |\mathcal{T}_{\gamma_2}|$ if $|\mathcal{T}_{\gamma_1}| = |\mathcal{T}_{\gamma_2}| \neq 0$. How to handle instances where the datasets are of different sizes will be discussed in Chapter 4.

2.5.1 Supervised learning

In SL, the predictive model is trained on the data available in regions near the two boundaries of the chosen parameter range denoted by I and II. Regions I and II are comprised of the set of sampled points $\Gamma_{\mathrm{I}} = \{\gamma_k | 1 \leq k \leq r_{\mathrm{I}}\}$ and $\Gamma_{\mathrm{II}} = \{\gamma_k | l_{\mathrm{II}} \leq k \leq K\}$, respectively. Here, r_{I}, $l_{\mathrm{II}} \in \mathbb{N}$ denote the rightmost and leftmost parameter points in regions I and II, respectively. In SL, we assume that there exist two distinct phases A and B, with the regions I and II being located deep within these phases. In physics, one often encounters the situation that the system is simple to analyze in certain limits. These points may then be used to construct the labeled training data in SL. In the Ising model, for example, it is clear that $T = 0$ corresponds to an ordered state and $T \to \infty$ is a disordered state. What is unclear, however, is at which intermediate temperature T_c the system transitions from one phase to another.

Without loss of generality, we assign the labels $y = 1$ and $y = 0$ to data obtained in regions I and II, respectively.[7] The predictive model $\hat{y}$ is trained to minimize a cross-entropy (CE) loss

$$\mathcal{L}_{\mathrm{SL}} = -\frac{1}{|\mathcal{T}|}\sum_{\boldsymbol{x}\in\mathcal{T}}\Big(y(\boldsymbol{x})\ln\left[\hat{y}(\boldsymbol{x})\right] + \left[1 - y(\boldsymbol{x})\right]\ln\left[1 - \hat{y}(\boldsymbol{x})\right]\Big). \tag{2.7}$$

The sum runs over all data points in the training set $\mathcal{T}$, where $\mathcal{T}_\gamma = \{\}$ for all $\gamma \notin \Gamma_{\mathrm{I}} \cup \Gamma_{\mathrm{II}}$. Let us denote the set of training data underlying regions I and II as $\mathcal{T}_{\mathrm{I}} = \biguplus_{\gamma\in\Gamma_{\mathrm{I}}}\mathcal{T}_\gamma$ and $\mathcal{T}_{\mathrm{II}} = \biguplus_{\gamma\in\Gamma_{\mathrm{II}}}\mathcal{T}_\gamma$, respectively. To prevent overfitting during training, one may make use of a validation set $\mathcal{V}$, add regularization terms to Equation (2.7), or utilize other common tricks in ML [Goodfellow *et al.*, 2016].[8] The output of the predictive model $\hat{y}(\boldsymbol{x}) \in [0, 1]$ corresponds to the probability of input $\boldsymbol{x}$ having the

[7] In subsequent chapters, we may sometimes switch the ordering of these labels.

[8] In Equation (2.7), we evaluated the loss over the training data $\mathcal{T}$. We may also perform the sum over all data contained within the validation set or the test set. The corresponding loss is then referred to as validation loss and test loss, respectively.

label $y = 1$, whereas $1 - \hat{y}(\boldsymbol{x})$ is the probability that the input $\boldsymbol{x}$ carries the label $y = 0$.

After training the predictive model to minimize the loss function in Equation (2.7), it is evaluated on the evaluation set $\mathcal{E}$, which contains data across the entire sampled parameter range. In particular, $|\mathcal{E}_\gamma| \neq 0$ for all $\gamma \notin \Gamma_{\mathrm{I}} \cup \Gamma_{\mathrm{II}}$. Averaging over the predictions $\hat{y}(\boldsymbol{x})$ for all data $\mathcal{E}_\gamma$ at a given point $\gamma \in \Gamma$ yields a prediction as a function of the tuning parameter

$$\hat{y}_{\mathrm{SL}}(\gamma) = \frac{1}{|\mathcal{E}_\gamma|} \sum_{\boldsymbol{x} \in \mathcal{E}_\gamma} \hat{y}(\boldsymbol{x}). \tag{2.8}$$

The indicator for phase transitions in SL, I_{SL}, is then given by the negative derivative of the prediction with respect to the tuning parameter

$$I_{\mathrm{SL}}(\gamma) = - \left. \frac{\partial \hat{y}_{\mathrm{SL}}(\gamma)}{\partial \gamma} \right|_{\gamma} . \tag{2.9}$$

The estimated critical value of the tuning parameter in SL corresponds to the location of the global maximum in its indicator [Equation (2.9)], which can easily be determined in an automated fashion without human supervision. If one chooses to label data obtained in region I with $y = 0$ and region II with $y = 1$ instead, the same indicator signal can be recovered via a sign change $I_{\mathrm{SL}} \to -I_{\mathrm{SL}}$. Alternatively, one may simply define the indicator as the absolute value of the derivative. Note that in the original proposal of SL by Carrasquilla and Melko [2017], the estimated critical value of the tuning parameter in SL was defined as $\operatorname{argmin}_{\gamma \in \Gamma} |\hat{y}(\gamma) - 0.5|$, see Appendix A for a comparison motivating our choice.

Intuitively, if there is a transition from one phase to another (phase A to phase B) when varying the tuning parameter γ, the mean predictions $\hat{y}_{\mathrm{SL}}(\gamma)$ should drop from $\hat{y}_{\mathrm{SL}}(\gamma_1) \approx 1$ (deep within phase A) to $\hat{y}_{\mathrm{SL}}(\gamma_K) \approx 0$ (deep within phase B) as γ is increased, see Figure 2.6(a). If the transition is sharp, the predictions should also change abruptly. Such a change results in a peak in the negative derivative of the predictions, i.e., in the indicator for phase transitions. In that case, the predictive model acts as an order parameter that approaches 1/0 deep within phase A/B. In general, one expects the predictions – and thus the indicator – to vary most strongly at the critical point γ_c. Instead, if there is only a single phase, one expects the predictions to be approximately constant, resulting in a flat indicator I_{SL}.

2.5.2 Learning by confusion

In this section, we present the original formulation of LBC. While SL requires partial knowledge of the phase diagram, i.e., the rough location of the phase boundary, LBC does not.[9] The labels are obtained by performing a split of the sampled parameter range into two neighboring regions labeled I and II. Each input $\boldsymbol{x}$ drawn in region I or II carries the label $y = 1$ or $y = 0$, respectively. The values of the tuning parameters that realize each of the $K+1$ possible bipartitions are given as $\gamma_k^{\mathrm{bp}} = \gamma_1 - \Delta\gamma/2 + (k-1)\Delta\gamma$, where $1 \leq k \leq K+1$.[10] For a given bipartition point γ_k^{bp}, regions I and II are then

[9] Phase transitions can only be detected in the sampled region of the parameter space Γ. Oftentimes the systems to be analyzed are, however, appropriately scaled such that $\gamma_c = \mathcal{O}(1)$ where γ is a normalized or dimensionless quantity. This can inform the choice of Γ.

[10] The choice of placing the bipartition point exactly in-between two sampled point is arbitrary. One has the freedom to place the bipartition point γ_k^{bp} at any location in the interval $[\gamma_{k-1}, \gamma_k]$. This reflects the fact that one can only determine the location of the critical point γ_c with a resolution of $\Delta\gamma$ (the grid spacing). We will make use of this freedom throughout this thesis.

comprised of the sampled points $\Gamma_{\mathrm{I}} = \{\gamma_j | \gamma_j < \gamma_k^{\mathrm{bp}},\, 1 \leq j \leq K\}$ and $\Gamma_{\mathrm{II}} = \{\gamma_j | \gamma_j > \gamma_k^{\mathrm{bp}},\, 1 \leq j \leq K\}$. Note that for bipartitions 1 ($\gamma_1^{\mathrm{bp}} = \gamma_1 - \Delta\gamma/2$) and $K+1$ ($\gamma_{K+1}^{\mathrm{bp}} = \gamma_K + \Delta\gamma/2$), regions I and II encompass the entire sampled parameter range and all data is assigned the label 1 or 0, respectively.

To each bipartition, i.e., choice of data labeling, we associate a distinct predictive model $\hat{y}$ that is trained to minimize a CE loss

$$\mathcal{L}_{\mathrm{LBC}} = -\frac{1}{|\mathcal{T}|} \sum_{\boldsymbol{x} \in \mathcal{T}} \Big(y(\boldsymbol{x}) \ln\left[\hat{y}(\boldsymbol{x})\right] + \left[1 - y(\boldsymbol{x})\right] \ln\left[1 - \hat{y}(\boldsymbol{x})\right] \Big), \tag{2.10}$$

where the training set $\mathcal{T}$ contains data points at each sampled point of the parameter space. Training $K+1$ distinct predictive models compared to a single predictive model in SL and PBM makes LBC computationally more demanding. We will discuss an approach to decrease the computational cost of LBC in Chapter 7. To prevent overfitting during training, one may make use of a validation set $\mathcal{V}$, add regularization terms to Equation (2.10), or utilize other common tricks in ML [Goodfellow *et al.*, 2016]. Again, the output of the predictive model $\hat{y}(\boldsymbol{x}) \in [0, 1]$ corresponds to the probability of input $\boldsymbol{x}$ having the label $y = 1$, whereas $1 - \hat{y}(\boldsymbol{x})$ is the probability of the input $\boldsymbol{x}$ carrying the label $y = 0$.

Once a predictive model has been trained to minimize the loss function in Equation (2.10) for a given bipartition, it is evaluated on a corresponding test set containing labeled data at each sampled point of the parameter space. In particular, we can compute the mean classification accuracy as a function of the bipartition parameter γ_k^{bp} ($1 \leq k \leq K+1$) as

$$I_{\mathrm{LBC}}(\gamma_k^{\mathrm{bp}}) = 1 - \frac{1}{|\mathcal{E}|} \sum_{\boldsymbol{x} \in \mathcal{E}} \left| \Theta\left[\hat{y}(\boldsymbol{x}) - 0.5\right] - y(\boldsymbol{x}) \right|, \tag{2.11}$$

where Θ denotes the Heaviside step function. The predictions $\hat{y}$ are obtained from the predictive model associated with the bipartition point γ_k^{bp}, and y are the corresponding labels.

Clearly, the mean classification accuracy I_{LBC} will exhibit trivial local maxima at the points $\gamma_1^{\mathrm{bp}} = \gamma_1 - \Delta\gamma/2$ and $\gamma_{K+1}^{\mathrm{bp}} = \gamma_K + \Delta\gamma/2$, where the entire data is assigned the label 0 or 1, respectively. Therefore, a predictive model effortlessly reaches a perfect accuracy of 1, because it simply needs to predict a single label regardless of the input. However, given that the underlying data can be separated into two distinct classes of similar character (i.e., phases) through appropriate bipartitioning of the parameter range at γ_{c}, one also expects the classification accuracy to have a local maximum at γ_{c}. At such a point, the predictive model is "least confused" by the choice of data labeling. Hence, the mean classification accuracy serves as the indicator for phase transitions within LBC. Typically, the indicator shows a characteristic W-shape, see Figure 2.6(b), where the middle-peak occurs at the bipartition point γ_k^{bp} closest to γ_{c}.

2.5.3 Prediction-based method

In PBM, a predictive model is trained on the entire parameter range to infer the value of the tuning parameter $\gamma \in \Gamma$ at which an input $\boldsymbol{x}$ was drawn. Similar to LBC, PBM does not require knowledge of the rough location of the underlying phases. While SL and LBC constitute supervised *classification* tasks, PBM corresponds to a supervised *regression* task, where the label is given by the tuning parameter itself, $y(\boldsymbol{x}) = \gamma$ for all $\boldsymbol{x} \in \mathcal{D}_\gamma$.[11]

We train the predictive model to minimize a mean-squared-error (MSE) loss function

$$\mathcal{L}_{\text{PBM}} = \frac{1}{|\mathcal{T}|} \sum_{\boldsymbol{x} \in \mathcal{T}} \left[\hat{y}(\boldsymbol{x}) - y(\boldsymbol{x})\right]^2 , \tag{2.12}$$

where the training set contains data points across the entire parameter range. To prevent overfitting during training, one may make use of a validation set $\mathcal{V}$, add regularization terms to Equation (2.12), or utilize other common tricks in ML [Goodfellow *et al.*, 2016]. After training, the predictive model is evaluated on a test set $\mathcal{E}$ that contains data across the entire parameter range. Averaging over the predictions $\hat{y}(\boldsymbol{x})$ for all data $\mathcal{E}_\gamma$ at a given point $\gamma \in \Gamma$ yields a mean prediction as a function of the tuning parameter

$$\hat{y}_{\text{PBM}}(\gamma) = \frac{1}{|\mathcal{E}_\gamma|} \sum_{\boldsymbol{x} \in \mathcal{E}_\gamma} \hat{y}(\boldsymbol{x}). \tag{2.13}$$

We then compute the deviation of the prediction from the true underlying value of the tuning parameter $\delta y_{\text{PBM}}(\gamma) = \hat{y}_{\text{PBM}}(\gamma) - \gamma$. The indicator for phase transitions of PBM, I_{PBM}, is then given by the derivative of this deviation with respect to the tuning parameter[12]

$$I_{\text{PBM}}(\gamma) = \left.\frac{\partial \delta y_{\text{PBM}}(\gamma)}{\partial \gamma}\right|_\gamma = \left.\frac{\partial \hat{y}_{\text{PBM}}(\gamma)}{\partial \gamma}\right|_\gamma - 1. \tag{2.14}$$

The estimated critical value of the tuning parameter in PBM corresponds to the location of the global maximum in its indicator [Equation (2.14)].

Intuitively, if there is only a single phase, in which inputs cannot be distinguished well by the predictive model, one expects the mean predictions to be approximately constant, see Figure 2.6(c). This results in the deviations δy_{PBM} varying approximately linear with the tuning parameter. Hence, the indicator I_{PBM} will be roughly constant. However, if there is a transition from one phase to another as the tuning parameter is varied, the predictions and the corresponding deviations also vary sharply. This results in a peak in the derivative of the deviations, i.e., the indicator for phase transitions I_{PBM}. In particular, one expects that the predictions are most susceptible at the phase boundary. Thus, its derivative should vary most strongly at the critical point γ_{c}.

2.5.4 Using neural networks as predictive models

In the previous sections, we have explicitly been vague about the choice of predictive model and details on the training procedure. Intuitively, one expects that SL, LBC, and PBM should be able to detect phase transitions in conjunction with any predictive model that can solve the underlying prediction task somewhat well, i.e., can guess the label y (or value γ) of a sample $\boldsymbol{x}$ somewhat well. As such, while the specific model and training choices may drastically influence the computational efficiency of the methods, many distinct choices may eventually correctly detect phase transitions.

[11] In principle, one can formulate PBM as a supervised classification task with $K = |\Gamma|$ distinct classes – one class for each sampled parameter value. We will make use of this formulation in Chapter 4.

[12] In Chapter 4 and onwards, we simply focus on the derivative of the predictions themselves, dropping the constant offset of -1.

The three methods have originally been proposed with NNs in mind, making them highly flexible and applicable without much prior knowledge of the underlying system. The corresponding training, as well as evaluation procedures, have been fairly standard following the common practices for NN applications to supervised classification and regression tasks, see [Goodfellow *et al.*, 2016; Dawid *et al.*, 2022] for further information on the basics of NN training and ML in general. In the following, we will showcase this by considering an application of SL, LBC, and PBM with NNs to the Ising model (Section 2.3.1).

Data preparation

The input data consists of spin configurations $\boldsymbol{\sigma}$ on a square lattice sampled from Boltzmann distributions at various temperatures. We consider a lattice of linear size $L = 60$. Before training, each spin configuration $\boldsymbol{\sigma} = \{\sigma_1, \sigma_2, \dots, \sigma_{L\times L}\}$ is flattened to a binary vector and standardized via the following affine transformation

$$\sigma_i \mapsto \frac{\sigma_i - \langle \sigma_i \rangle}{\mathrm{std}_i}, \tag{2.15}$$

where $\langle \sigma_i \rangle$ and std_i are the mean value and standard deviation of std_i across the training data, respectively. Standardization generally leads to a faster rate of convergence when applying gradient-based optimizers [LeCun *et al.*, 2012].

The sampled range of the tuning parameter $\gamma = k_{\mathrm{B}}T/J$ runs from $\gamma_1 = 0.05$ to $\gamma_K = 10.0$ with a spacing $\Delta\gamma = 0.05$. In SL, we choose the training data to be located at the edges of this interval such that $\Gamma_{\mathrm{I}} = \{0.05, 0.1, 0.15, \dots, 1.0\}$ and $\Gamma_{\mathrm{II}} = \{9.0, 9.05, 9.1, \dots, 10.0\}$. At each point $\gamma \in \Gamma$, we have a total of $|\mathcal{D}_\gamma| = 10^3$ sampled spin configuration. We split this data randomly into training sets containing 80 % of the data and test sets containing the other 20 % of the data. In LBC and PBM, we have $|\mathcal{T}_\gamma| = 0.8 \cdot |\mathcal{D}_\gamma|$ and $|\mathcal{E}_\gamma| = 0.2 \cdot |\mathcal{D}_\gamma|$ for all $\gamma \in \Gamma$, whereas in SL $|\mathcal{T}_\gamma| = 0$ for all $\gamma \notin \Gamma_{\mathrm{I}} \cup \Gamma_{\mathrm{II}}$. We do not consider any separate validation set.

Neural network architecture

The NN is composed of a series of fully-connected layers, where rectified linear units (ReLUs), $f(z) = \max(0, z)$, are used as activation functions. Note that the “image-like” nature of the input data and the fact that the relevant correlations in the Ising model are between nearest neighbors, using a convolutional NN (CNN) would be more ideal (i.e., parameter efficient).

The NNs for SL and LBC have two output nodes, where a softmax activation function

$$f_i(\boldsymbol{z}) = \frac{e^{z_i}}{\sum_j e^{z_j}} \tag{2.16}$$

is used in the output layer to guarantee that $\hat{y} \in [0, 1]$. Here, the sum runs over all output nodes, and $\hat{y}$ corresponds to the value of one of the output nodes after an application of the softmax activation function. In PBM, no activation function is used for the output layer. The value of the single output node corresponds to $\hat{y}(\boldsymbol{\sigma})$, which is the estimated value of the tuning parameter at which the input $\boldsymbol{\sigma}$ was drawn. Here, we use NNs with a single hidden layer containing 10 nodes.

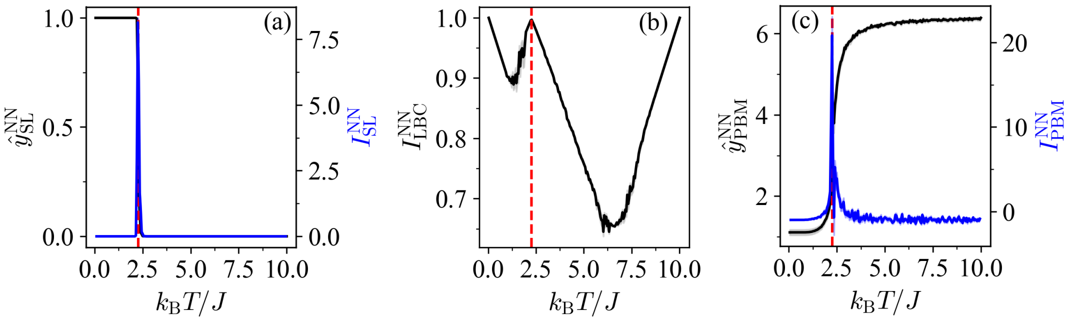

FIGURE 2.6: Results for the Ising model ($L = 60$) with the dimensionless temperature as a tuning parameter $\gamma = k_{\mathrm{B}}T/J$. (a) Mean NN-based prediction $\hat{y}_{\mathrm{SL}}^{\mathrm{NN}}$ in SL (black) and the corresponding indicator $I_{\mathrm{SL}}^{\mathrm{NN}}$ (blue). (b) NN-based indicator of LBC, $I_{\mathrm{LBC}}^{\mathrm{NN}}$ (black). (c) Mean NN-based prediction $\hat{y}_{\mathrm{PBM}}^{\mathrm{NN}}$ in PBM (black) and the corresponding indicator $I_{\mathrm{PBM}}^{\mathrm{NN}}$ (blue). The critical temperature [Equation (2.3)] is highlighted by a red dashed line. Bold lines show mean results over 10 independent NN training runs and the shaded band depicts the corresponding standard error of the mean.

Training

The NNs are implemented using PyTorch [Paszke *et al.*, 2019] in `Python`, where the weights and biases are adjusted via minibatch stochastic gradient descent (SGD) with the Adam optimizer [Kingma and Ba, 2014] (using standard settings) to minimize the training loss function over a series of training epochs. In SL and LBC, we train on a CE loss function [Equation (2.7) and (2.10), respectively], whereas in PBM we train on an MSE loss function [Equation (2.12)]. Gradients are calculated using backpropagation [Rumelhart *et al.*, 1986; Goodfellow *et al.*, 2016; Baydin *et al.*, 2018]. We train for 20 epochs with a learning rate of 0.001 and a batch size of 64.

Results

The results obtained using SL, LBC, and PBM with these simple NNs are shown in Figure 2.6. The indicator of all three methods correctly highlights the critical transition temperature of the Ising model. The mean predictions and indicators exhibit the characteristic properties outlined in Secs. 2.5.1-2.5.3. In particular, the mean predictions of SL and PBM serve as order parameters attaining distinct values within the ordered and disordered phase. The indicator of LBC displays a characteristic W-shape.

2.6 State of the field and open questions

Since their inception, SL [Carrasquilla and Melko, 2017; Wetzel and Scherzer, 2017; Schindler *et al.*, 2017; Ch'ng *et al.*, 2017; Ohtsuki and Ohtsuki, 2017; Broecker *et al.*, 2017; Venderley *et al.*, 2018; Beach *et al.*, 2018; Rem *et al.*, 2019; Käming *et al.*, 2021; Bohrdt *et al.*, 2021; Huang *et al.*, 2022b; Maskara *et al.*, 2022; Liu *et al.*, 2023; Miles *et al.*, 2023; Cybiński *et al.*, 2024b], LBC [Van Nieuwenburg *et al.*, 2017; Liu and van Nieuwenburg, 2018; Beach *et al.*, 2018; Suchsland and Wessel, 2018; Lee and Kim, 2019; Ni *et al.*, 2019b,a; Kharkov *et al.*, 2020; Greplova *et al.*, 2020; Bohrdt *et al.*, 2021; Corte *et al.*, 2021; He *et al.*, 2022; Gavreev *et al.*, 2022; Zvyagintseva *et al.*, 2022; Sun *et al.*, 2023; Richter-Laskowska *et al.*, 2023; Schlömer and Bohrdt, 2023; Guo and He, 2023; Guo *et al.*, 2023; Tao *et al.*, 2023; Zhao *et al.*, 2024; Cohen *et al.*, 2024; Kasatkin *et al.*, 2024a,b; Ghosh and Sarkar, 2024; Caleca *et al.*, 2024; Issa *et al.*, 2025], and PBM [Schäfer and Lörch, 2019; Greplova *et al.*, 2020; Ge and Tang, 2021; Singh *et al.*, 2021; Bohrdt *et al.*, 2021; Arnold *et al.*, 2021; Tao *et al.*, 2023; Cybiński *et al.*, 2024a; Frk *et al.*, 2024] (as well as related methods based on regression of the tuning parameter [Kashiwa *et al.*, 2019; Ho and Wang, 2021; Guo and He, 2023; Guo *et al.*, 2023; Ho and Wang, 2023]) have successfully been applied in conjunction with NNs as predictive models to detect phase transitions in a variety of systems ranging from models in classical equilibrium physics and quantum non-equilibrium dynamics to molecular dynamics simulations of protein folding, gene expression in mouse muscle regeneration, or air pollution data.[13]

Sometimes, however, the methods do fail. In [Carrasquilla and Melko, 2017], for example, it has been reported that the topological crossover of the IGT cannot be detected well using SL with simple feedforward NNs. Instead, CNNs must be used. Greplova *et al.* [2020] also reported difficulties in detecting the topological crossover of the IGT using LBC – the indicator of NN-based LBC did not show any clear peak even when utilizing CNNs of various sizes. In contrast, PBM succeeded. In [Beach *et al.*, 2018], it has been found that SL must rely on feature engineering to explicitly encode topological information to properly detect the Berezinskii–Kosterlitz–Thouless (BKT) transition in the classical 2D XY model (see Section 3.6.3), and the peak in the LBC indicator seems to occur at the same location as the peak in the heat capacity slightly above the true critical point. In [Schäfer and Lörch, 2019], it was found that while the indicator of PBM they obtained shows a clear peak at the critical temperature of the Ising model early on during training, a spurious peak appears as the training progresses that dominates over the correct signal.

At the time of starting this thesis in August 2021, it was largely unclear why the methods failed in these particular instances and succeeded in others. The field largely progressed via trial and error: the methods were applied to prototypical models with known phase diagrams to explore the boundaries of their capabilities. It was difficult to predict which phase transitions a method may be able to detect and on which ones it would fail, and there were no formal guarantees. In hindsight, this may not be surprising given the fact that the methods have been motivated heuristically, largely based on the success of NNs in image classification tasks, and little to no physical principles are explicitly present in their definition. Moreover, NNs are black-box models that are notoriously difficult to interpret.

From the discussion above, it becomes clear that in certain instances, a method fails when using a predictive model that is too weak, in others it fails when the predictive model gets too powerful, and sometimes a method does not seem to work with a predictive model of any kind.[14] This hints at a core issue with SL, LBC, and PBM. Recall that the exciting promise of these methods is their ability to detect previously unexplored phase transitions with little prior knowledge. This becomes difficult, however, if their success heavily relies on choosing a model with appropriate capacity.[15]

[13] While the list of references utilizing and developing SL is far from complete, the lists for LBC and PBM are – to the best of the author's knowledge – exhaustive.

[14] It is no coincidence that we have been vague about how we chose the hyperparameters – such as the NN architecture, number of training epochs, or the choice of training region in SL – when investigating the Ising model in Section 2.5.4. The reader may rightfully wonder whether the phase transition can robustly be detected with different hyperparameter settings. We will investigate this question in more detail in Chapter 3.

[15] By the capacity of a model we mean its ability to fit a wide variety of functions [Goodfellow *et al.*, 2016; Hu *et al.*, 2021]. Training time, NN size, and regularization are all factors that influence

Imagine a scenario where the predicted critical point shifts with varying model capacity – is there even an underlying critical point and if so, where is it located? Ideally, we would like the methods to succeed with powerful models. In this case, we are guaranteed that as we put more resources into our predictive model – for example by increasing its expressivity (such as the number of neurons), training time, or training data – the estimate of the critical point improves. This corresponds to a scenario where we know the core objective (i.e., loss function) to be optimized. If this does not occur, it signals a mismatch between our utilized objective and the actual objective to be optimized for detecting phase transitions from data. Using NNs alone, it is difficult to assess whether the previous failures can indeed be attributed to a more fundamental issue or are instead related to our inability to craft powerful predictive models, e.g., due to a limited amount of data, limited training time, or suboptimal choice of hyperparameters. Put differently, if a method fails with a certain choice of data, NN, and training procedure, one can never be sure whether this may change by appropriate modifications to the latter.

We have highlighted several instances where SL, LBC, or PBM failed to detect phase transitions in the past. However, even if a method succeeds, it remains largely unclear how it does so due to the black-box nature of the NN-based predictive model. Various works have tried to unravel what physical quantities the NNs utilize to distinguish phases of matter and detect phase transitions. In [Carrasquilla and Melko, 2017], for example, it was shown that a simple NN with three hidden nodes can in principle distinguish the ordered and disordered states of the Ising model in SL based on the magnetization. Similarly, through an *a posteriori* analysis of small trained NNs via SL, it is found that they do indeed classify spin configurations based on their magnetization. In [Wetzel and Scherzer, 2017], the receptive field size of CNNs classifying spin configurations of the Ising model was systematically reduced, restricting the functions that the CNN can compute. It was found that the classification accuracy remains largely unchanged up until a receptive field size of 1×2. At this size, it is found that the CNN computes the expected energy per site (i.e., the energy of the spin configuration). The classification accuracy is observed to drop slightly once the CNN is reduced to a 1×1 receptive field, corresponding to a computation of the magnetization. Similarly, in [Suchsland and Wessel, 2018] it was found that NN-based indicator curves in LBC can be qualitatively reproduced by a handcrafted model that classifies spin configurations of the Ising model by comparing their energy to the mean energy at the bipartition temperature. In contrast, while a handcrafted model based on magnetization still highlights the critical point via LBC, its corresponding accuracy (i.e., the LBC indicator) is significantly lower. In conclusion, the numerical results of [Wetzel and Scherzer, 2017] and [Suchsland and Wessel, 2018] suggest that the energy is sufficient for distinguishing the two phases of the Ising model via SL and LBC, i.e., all other information contained in a spin configuration is redundant for the task at hand). The questions that remain are concerned with whether this statement holds for other systems, predictive models, and methods, whether the statement is only true approximately or whether it is exact, and whether it can be proven from first principles.

Identifying what physical quantities predictive models learn to compute to distinguish phases and detect phase transitions remains an active field of research [Wetzel and Scherzer, 2017; Zhang *et al.*, 2020; Miles *et al.*, 2021, 2023; Schlömer and Bohrdt,

the effective capacity of a model and thus determine its ability to approximate the optimal predictive model, i.e., its ability to attain the global minimum of the relevant loss function. We will investigate the influence of model capacity on the NN-based methods for detecting phase transitions in Chapter 3.

2023; Zhang *et al.*, 2024b; Cybiński *et al.*, 2024a]. Most works approach this question by choosing an interpretable NN architecture[16], training the NNs in a principled manner to gradually vary their model capacity (e.g., initializing training runs with distinct regularization strengths [Miles *et al.*, 2021]), and analyzing the trained NNs *a posteriori* to determine the key learned physical quantities.

Taking a different stance, it would be great to be able to understand based on what *physical principle* these ML methods detect phase transitions. In contrast to previous works trying to answer this question explicitly for a single model system and method (with potential variations in the predictive model, i.e., NN architecture), we want to aim for a statement that encompasses various models and methods. Moreover, the statement should be largely insensitive to the choice of NN: the NN may not need to be explicitly interpretable, i.e., act as a “white box”, but can remain a black box. The search for such a holistic working principle may also guide us in terms of modifications that need to be performed to remedy the remaining issues of the methods (recall that the methods do fail in certain instances).

In the first part of this thesis (starting with Chapter 3), we will tackle the fundamental open problems mentioned above. While we focus on the methods of SL, LBC, and PBM in this thesis, many other ML methods for detecting phase transitions from data face the same issues. As such, the findings of this thesis are also expected to be relevant for gaining a better understanding of other ML methods and improving their performance.

The results and figures presented in this chapter have been published in parts in [Arnold and Schäfer, 2022b].

[16] A CNN can, for example, be interpreted by looking at its filters.

Chapter 3

Optimal Predictive Models for Detecting Phase Transitions

The results presented in this chapter are based on the following publication:

Replacing neural networks by optimal analytical predictors for the detection of phase transitions,
J. Arnold and F. Schäfer,
Phys. Rev. X **12**, 031044 (2022).

3.1 Motivation

In this chapter, we start addressing the gaps in knowledge outlined in Section 2.6 by deriving analytical expressions for the optimal predictions of the NNs underlying supervised learning (SL), learning by confusion (LBC), and the prediction-based method (PBM). The predictions are optimal in the sense that they minimize the target loss function, i.e., the corresponding model performs the desired classification or regression task (as specified by the loss function) optimally.[1] Based on the optimal predictions, we find analytical expressions for the optimal indicators of phase transitions of these three methods. The optimal indicators correspond to the output of the methods when using optimal predictive models, i.e., ideal high-capacity predictive models, such as well-trained, highly expressive NNs. The inner workings of these methods are revealed through the dependence of the optimal indicators on the input data. Moreover, the analytical expressions make it possible to compute the optimal indicator directly from the input data without training NNs, see step 2*) in Figure 3.1, manifesting an alternative numerical routine to infer phase transitions from data. We demonstrate the procedure in a numerical study on a variety of models exhibiting, e.g., symmetry-breaking, topological, quantum, and many-body localization phase transitions.

3.2 Optimal predictions and indicators

In this section, we derive and discuss the optimal indicators of phase transitions I^{opt} for SL, LBC, and PBM (recall Section 2.5) in detail. The optimal indicators can be directly calculated given the predictions $\hat{y}^{\mathrm{opt}}(\boldsymbol{x})$ of an optimal model which minimizes the corresponding training loss function. If the training loss function is computed using a finite amount of data, the model is said to be *empirically optimal*. This means that while we do not know how the model performs on unseen data, i.e.,

[1] As we will see in this chapter, this does, however, not necessarily mean that such optimal predictive models are ideal for highlighting phase transitions via these methods. In some instances, predictive models that do not minimize the target loss function may highlight the critical point more clearly compared to optimal predictive models.

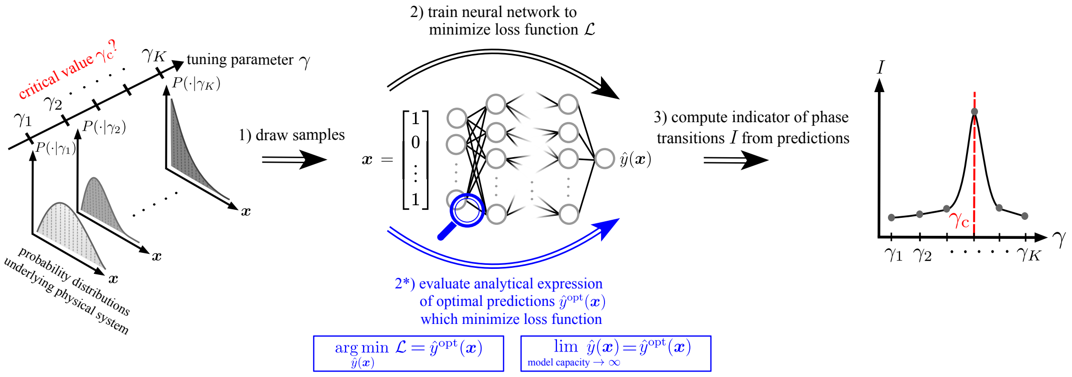
critical value γ_c?
γ_1
γ_2
γ_K
tuning parameter γ
$P(\cdot|\gamma_1)$
$P(\cdot|\gamma_2)$
$P(\cdot|\gamma_K)$
$\boldsymbol{x}$
probability distributions underlying physical system
1) draw samples
$\boldsymbol{x} = [1, 0, \vdots, 1]$
2) train neural network to minimize loss function $\mathcal{L}$
$\hat{y}(\boldsymbol{x})$
2*) evaluate analytical expression of optimal predictions $\hat{y}^{\text{opt}}(\boldsymbol{x})$ which minimize loss function
$\arg\min_{\hat{y}(\boldsymbol{x})} \mathcal{L} = \hat{y}^{\text{opt}}(\boldsymbol{x})$
$\lim_{\text{model capacity} \to \infty} \hat{y}(\boldsymbol{x}) = \hat{y}^{\text{opt}}(\boldsymbol{x})$
3) compute indicator of phase transitions I from predictions
I
γ_c
γ
γ_1 γ_2 γ_K

FIGURE 3.1: Schematic representation of the setup and workflow of SL, LBC, and PBM for detecting phase transitions from data, cf. Figure 2.5. The key contribution of this chapter is highlighted in blue: We derive analytical expressions for the optimal predictions $\hat{y}^{\text{opt}}$ of the NNs used in these three methods. The optimal predictions minimize the corresponding loss function and are thus achieved by NNs whose capacity, i.e., their ability to fit a wide variety of functions, is sufficiently high. The optimal predictions can solely be expressed in terms of the probability distributions underlying the physical system. Using the optimal predictions $\hat{y}^{\text{opt}}$ in place of the NN predictions $\hat{y}$, we further obtain analytical expressions for the optimal indicators of phase transitions I^{opt} (step 2*). Evaluating these analytical expressions provides an alternative path for computing indicators of phase transitions without *ever* training NNs.

what its performance is across the entire distribution of inputs, we know that no other model outperforms it on the selected set of training data. In the limit of infinite training data, i.e., given accurate estimates of the probability distributions underlying the physical system $\{P(\cdot|\gamma)\}_{\gamma\in\Gamma}$, the empirically-optimal model approaches *Bayes optimality* [Devroye *et al.*, 1996; Goodfellow *et al.*, 2016]: No other statistical model can outperform it on the classification or regression task at hand (on average) – considering in-distribution input data. In this case, the loss value achieved by the model coincides with the *Bayes error* [Devroye *et al.*, 1996; Goodfellow *et al.*, 2016], i.e., the intrinsic irreducible error inherent to the problem.[2]

3.2.1 Supervised learning

In SL (see Section 2.5.1), for any $\boldsymbol{x} \in \bar{\mathcal{T}}$, the optimal predictions are given as

$$\hat{y}_{\text{SL}}^{\text{opt}}(\boldsymbol{x}) = \frac{\tilde{P}_{\text{I}}^{(\mathcal{T})}(\boldsymbol{x})}{\tilde{P}_{\text{I}}^{(\mathcal{T})}(\boldsymbol{x}) + \tilde{P}_{\text{II}}^{(\mathcal{T})}(\boldsymbol{x})}, \tag{3.1}$$

where

$$\tilde{P}_{\text{I}}^{(\mathcal{T})}(\boldsymbol{x}) = \sum_{\gamma\in\Gamma_{\text{I}}} \tilde{P}^{(\mathcal{T})}(\boldsymbol{x}|\gamma) \tag{3.2}$$

[2] In the following, we will often not explicitly specify whether the model is "only" empirically optimal or indeed Bayes optimal. This can be inferred from the context, i.e., the amount of training data used – or equivalently – the quality of the estimate of the underlying probability distributions $\{P(\cdot|\gamma)\}_{\gamma\in\Gamma}$.

and

$$\tilde{P}_{\text{II}}^{(\mathcal{T})}(\boldsymbol{x}) = \sum_{\gamma \in \Gamma_{\text{II}}} \tilde{P}^{(\mathcal{T})}(\boldsymbol{x}|\gamma) \tag{3.3}$$

are the (unnormalized) probabilities of drawing an input $\boldsymbol{x}$ in region I and II, respectively, as estimated based on the training data set $\mathcal{T}$. The proof can be found below. Hence, the optimal prediction for a particular input corresponds to the probability of drawing that input in region I compared to region II. Here, $\tilde{P}^{(\mathcal{T})}(\boldsymbol{x}|\gamma)$ denotes the (normalized) probability to draw the input $\boldsymbol{x}$ at the sampled point γ as estimated based on the training data $\mathcal{T}$, i.e., $\tilde{P}^{(\mathcal{T})}(\boldsymbol{x}|\gamma) = \mathcal{T}_{\gamma}(\boldsymbol{x})/|\mathcal{T}_{\gamma}| \approx P(\boldsymbol{x}|\gamma)$, where $\mathcal{T}_{\gamma}(\boldsymbol{x})$ is the number of times the input $\boldsymbol{x}$ is present in the training set $\mathcal{T}_{\gamma}$. An expression for the optimal value of the loss in SL, $\mathcal{L}_{\text{SL}}^{\text{opt}}$, can be obtained by replacing $\hat{y}(\boldsymbol{x})$ with $\hat{y}_{\text{SL}}^{\text{opt}}(\boldsymbol{x})$ in Equation (2.7), where, by definition, $\mathcal{L}_{\text{SL}}^{\text{opt}} \leq \mathcal{L}_{\text{SL}}$.

Assuming that all inputs within the evaluation set $\mathcal{E}$ are already present in the training set $\mathcal{T}$, i.e., $\bar{\mathcal{T}} = \bar{\mathcal{E}}$, the mean optimal prediction at a given point $\gamma \in \Gamma$ is

$$\hat{y}_{\text{SL}}^{\text{opt}}(\gamma) = \sum_{\boldsymbol{x} \in \bar{\mathcal{E}}} \tilde{P}^{(\mathcal{E})}(\boldsymbol{x}|\gamma) \hat{y}_{\text{SL}}^{\text{opt}}(\boldsymbol{x}). \tag{3.4}$$

This corresponds to the estimated probability of finding an input drawn at that point γ in region I compared to region II, where $\tilde{P}^{(\mathcal{E})}(\boldsymbol{x}|\gamma) = \mathcal{E}_{\gamma}(\boldsymbol{x})/|\mathcal{E}_{\gamma}| \approx P(\boldsymbol{x}|\gamma)$. We find the assumption $\bar{\mathcal{T}} = \bar{\mathcal{E}}$ to be (approximately) satisfied for all physical systems analyzed in this chapter and can estimate the errors arising from a violation, see Section 3.5 and Appendix B. The optimal indicator of phase transitions in SL is then given as

$$I_{\text{SL}}^{\text{opt}}(\gamma) = -\left. \frac{\partial \hat{y}_{\text{SL}}^{\text{opt}}(\gamma)}{\partial \gamma} \right|_{\gamma} . \tag{3.5}$$

In general, there will be a transition point where the probability in Equation (3.4) changes most and thus where its derivative, the optimal indicator in Equation (3.5), peaks.

Proof

In SL, a predictive model is trained to minimize the CE loss function given in Equation (2.7). Now, consider a particular input contained within the training set $\boldsymbol{x}' \in \bar{\mathcal{T}}$. We can determine the optimal model prediction $\hat{y}_{\text{SL}}^{\text{opt}}(\boldsymbol{x}')$ for this particular input by minimizing the loss function in Equation (2.7) with respect to $\hat{y}(\boldsymbol{x}')$, i.e., by solving the necessary condition

$$\frac{\partial \mathcal{L}_{\text{SL}}}{\partial \hat{y}(\boldsymbol{x}')} = -\frac{1}{|\mathcal{T}|} \sum_{\boldsymbol{x}' \in \mathcal{T}} \left(\frac{y(\boldsymbol{x}')}{\hat{y}(\boldsymbol{x}')} - \frac{1 - y(\boldsymbol{x}')}{1 - \hat{y}(\boldsymbol{x}')} \right) = 0. \tag{3.6}$$

Using the explicit expressions for the labels ($y = 1$ and $y = 0$ for all inputs drawn in region I and II, respectively) in Equation (3.6), we have

$$\frac{\sum_{\gamma \in \Gamma_{\text{I}}} \mathcal{T}_{\gamma}(\boldsymbol{x}')}{\sum_{\gamma \in \Gamma_{\text{II}}} \mathcal{T}_{\gamma}(\boldsymbol{x}')} = \frac{\mathcal{T}_{\text{I}}(\boldsymbol{x}')}{\mathcal{T}_{\text{II}}(\boldsymbol{x}')} = \frac{\hat{y}(\boldsymbol{x}')}{1 - \hat{y}(\boldsymbol{x}')}. \tag{3.7}$$

Here, $\mathcal{T}_{\text{I/II}}(\boldsymbol{x}')$ denotes the number of times the input $\boldsymbol{x}'$ is found in region I or II, respectively, given the training set. In SL, the predictive model must, by definition, satisfy $\hat{y}(\boldsymbol{x}) \in [0,1]$ for all $\boldsymbol{x}$. Thus, Equation (3.7) is satisfied given predictions of the form

$$\hat{y}_{\mathrm{SL}}^{\mathrm{opt}}(\boldsymbol{x}') = \frac{\mathcal{T}_{\mathrm{I}}(\boldsymbol{x}')}{\mathcal{T}_{\mathrm{I}}(\boldsymbol{x}') + \mathcal{T}_{\mathrm{II}}(\boldsymbol{x}')}. \tag{3.8}$$

The opposite choice of labeling ($y = 0$ and $y = 1$ for all inputs drawn in region I and II, respectively) is equally valid and would result in

$$\hat{y}_{\mathrm{SL}}^{\mathrm{opt}}(\boldsymbol{x}') = \frac{\mathcal{T}_{\mathrm{II}}(\boldsymbol{x}')}{\mathcal{T}_{\mathrm{I}}(\boldsymbol{x}') + \mathcal{T}_{\mathrm{II}}(\boldsymbol{x}')}. \tag{3.9}$$

That is, the role of $\hat{y}_{\mathrm{SL}}^{\mathrm{opt}}(\boldsymbol{x}')$ and $1 - \hat{y}_{\mathrm{SL}}^{\mathrm{opt}}(\boldsymbol{x}')$ are swapped. In this thesis, we stick to the former choice [Equation (3.8)]. The optimality of the predictions in Equation (3.8) can be confirmed by calculating the second derivative of the loss function

$$\frac{\partial^2 \mathcal{L}_{\mathrm{SL}}}{\partial \hat{y}(\boldsymbol{x}')^2} = \frac{\mathcal{T}_{\mathrm{I}}(\boldsymbol{x}')}{|\mathcal{T}|} \frac{1}{\hat{y}(\boldsymbol{x}')^2} + \frac{\mathcal{T}_{\mathrm{II}}(\boldsymbol{x}')}{|\mathcal{T}|} \frac{1}{[1 - \hat{y}(\boldsymbol{x}')]^2} > 0. \tag{3.10}$$

Let us denote the empirical probability distribution governing the input data within the training set as $\tilde{P}^{(\mathcal{T})}(\boldsymbol{x}'|\gamma) = \mathcal{T}_\gamma(\boldsymbol{x}')/|\mathcal{T}_\gamma|$. Assuming that the training set at each point $\gamma \in \Gamma_{\mathrm{I}} \cup \Gamma_{\mathrm{II}}$ is of equal size, this allows for Equation (3.8) to be expressed as

$$\hat{y}_{\mathrm{SL}}^{\mathrm{opt}}(\boldsymbol{x}') = \frac{\tilde{P}_{\mathrm{I}}^{(\mathcal{T})}(\boldsymbol{x}')}{\tilde{P}_{\mathrm{I}}^{(\mathcal{T})}(\boldsymbol{x}') + \tilde{P}_{\mathrm{II}}^{(\mathcal{T})}(\boldsymbol{x}')}, \tag{3.11}$$

where

$$\tilde{P}_{\mathrm{I}}^{(\mathcal{T})}(\boldsymbol{x}') = \sum_{\gamma \in \Gamma_{\mathrm{I}}} \tilde{P}^{(\mathcal{T})}(\boldsymbol{x}'|\gamma) \tag{3.12}$$

and

$$\tilde{P}_{\mathrm{II}}^{(\mathcal{T})}(\boldsymbol{x}') = \sum_{\gamma \in \Gamma_{\mathrm{II}}} \tilde{P}^{(\mathcal{T})}(\boldsymbol{x}'|\gamma). \tag{3.13}$$

Repeating the above procedure for all inputs within the training set $\bar{\mathcal{T}}$, we obtain the expression in Equation (3.1).

The same optimal predictions are obtained when training on a MSE loss function

$$\mathcal{L}_{\mathrm{MSE}} = \frac{1}{|\mathcal{T}|} \sum_{\boldsymbol{x} \in \mathcal{T}} \left[\hat{y}(\boldsymbol{x}) - y(\boldsymbol{x})\right]^2, \tag{3.14}$$

instead of a CE loss function. Again, consider a particular input $\boldsymbol{x}'$ contained within the training set $\bar{\mathcal{T}}$. We can determine the optimal model prediction $\hat{y}_{\mathrm{SL}}^{\mathrm{opt}}(\boldsymbol{x}')$ for this input by minimizing the loss function in Equation (3.14) with respect to $\hat{y}(\boldsymbol{x}')$, i.e., by solving

$$\frac{\partial \mathcal{L}_{\mathrm{MSE}}}{\partial \hat{y}(\boldsymbol{x}')} = \frac{2}{|\mathcal{T}|} \sum_{\boldsymbol{x}' \in \mathcal{T}} \left[\hat{y}(\boldsymbol{x}') - y(\boldsymbol{x}')\right] = 0. \tag{3.15}$$

Plugging the expression for the labels given by a one-hot-encoding in Equation (3.15), we have

$$\mathcal{T}_{\mathrm{I}}(\boldsymbol{x}')\left[1 - \hat{y}(\boldsymbol{x}')\right] - \mathcal{T}_{\mathrm{II}}(\boldsymbol{x}')\hat{y}(\boldsymbol{x}') = 0. \tag{3.16}$$

This coincides with the condition for the predictions given in Equation (3.7) obtained from a CE loss function. Their optimality can be confirmed via

$$\frac{\partial^2 \mathcal{L}_{\mathrm{MSE}}}{\partial \hat{y}(\boldsymbol{x}')^2} = 2 \cdot \frac{\mathcal{T}_{\mathrm{I}}(\boldsymbol{x}') + \mathcal{T}_{\mathrm{II}}(\boldsymbol{x}')}{|\mathcal{T}|} > 0. \tag{3.17}$$

Therefore, in SL, the optimal predictions and indicators associated with optimal models trained on a CE or MSE loss function are identical.

3.2.2 Learning by confusion

For a given bipartition of the parameter range into regions I and II, the optimal predictions of LBC (see Section 2.5.2) for $\boldsymbol{x} \in \bar{\mathcal{T}}$ are given as (see below for a proof)

$$\hat{y}^{\mathrm{opt}}_{\mathrm{LBC}}(\boldsymbol{x}) = \frac{\tilde{P}^{(\mathcal{T})}_{\mathrm{I}}(\boldsymbol{x})}{\tilde{P}^{(\mathcal{T})}_{\mathrm{I}}(\boldsymbol{x}) + \tilde{P}^{(\mathcal{T})}_{\mathrm{II}}(\boldsymbol{x})}, \tag{3.18}$$

which corresponds to the probability of drawing the input in region I compared to region II as inferred from the training data. See below for a proof. This characteristic is inherent to the underlying classification task [compare Equations (3.1) and (3.18)]. The classification error associated with an input $\boldsymbol{x}$ in the training set is given by $\left|\Theta\left[\hat{y}^{\mathrm{opt}}_{\mathrm{LBC}}(\boldsymbol{x}) - 0.5\right] - y(\boldsymbol{x})\right|$. It arises from a "confusion" of the model: different labels can be assigned to the same input due to an overlap of the probability distributions underlying region I and II. The mean classification accuracy as evaluated on the evaluation set for a particular choice of bipartition, i.e., labeling of the data, then corresponds to

$$I^{\mathrm{opt}}_{\mathrm{LBC}} = 1 - \frac{1}{|\Gamma|} \sum_{\gamma \in \Gamma} \sum_{\boldsymbol{x} \in \bar{\mathcal{E}}_\gamma} \tilde{P}^{(\mathcal{E})}(\boldsymbol{x}|\gamma) \left|\Theta\left[\hat{y}^{\mathrm{opt}}_{\mathrm{LBC}}(\boldsymbol{x}) - 0.5\right] - y(\boldsymbol{x})\right|. \tag{3.19}$$

This forms the optimal indicator for phase transitions in LBC. If $\mathcal{T} = \mathcal{E}$, one may instead use the following relation (see proof below)

$$I^{\mathrm{opt}}_{\mathrm{LBC}} = 1 - \frac{1}{|\Gamma|} \sum_{\gamma \in \Gamma} \sum_{\boldsymbol{x} \in \bar{\mathcal{E}}_\gamma} \tilde{P}^{(\mathcal{E})}(\boldsymbol{x}|\gamma) \mathrm{min}\{\hat{y}^{\mathrm{opt}}_{\mathrm{LBC}}(\boldsymbol{x}), 1 - \hat{y}^{\mathrm{opt}}_{\mathrm{LBC}}(\boldsymbol{x})\}. \tag{3.20}$$

An expression for the optimal value of the loss in LBC, $\mathcal{L}^{\mathrm{opt}}_{\mathrm{LBC}}$, can be obtained by replacing $\hat{y}(\boldsymbol{x})$ with $\hat{y}^{\mathrm{opt}}_{\mathrm{LBC}}(\boldsymbol{x})$ in Equation (2.10). The critical point γ_{c} is highlighted by a dip in the mean classification error, i.e., by a peak in the mean classification accuracy [Equation (3.20)]. It corresponds to the bipartition point for which the probability distributions underlying the two regions have the least overlap (on average), resulting in the highest classification accuracy and the least confusion. While confusion can arise due to sub-optimal predictions of models with restricted capacity (see Section 3.7.5 for a concrete example), we find that confusion can even persist in the limit of high model capacity if it is inherent to the underlying data. Based on the analytical expressions, we thus gained an intuitive and rigorous understanding of the concept of confusion underlying LBC.

Proof

To reveal the phase transition by means of LBC, we perform several splits of the parameter range into two neighboring regions labeled I and II. For a fixed bipartition, we minimize a CE [Equation (2.10)] or MSE loss function

$$\mathcal{L}_{\text{MSE}} = \frac{1}{|\mathcal{T}|} \sum_{\boldsymbol{x} \in \mathcal{T}} \left[\hat{y}(\boldsymbol{x}) - y(\boldsymbol{x})\right]^2 . \tag{3.21}$$

Following the analysis of SL presented above for any $\boldsymbol{x} \in \mathcal{T}$, we obtain a similar expression for the optimal predictions

$$\hat{y}^{\text{opt}}_{\text{LBC}}(\boldsymbol{x}) = \frac{\tilde{P}^{(\mathcal{T})}_{\text{I}}(\boldsymbol{x})}{\tilde{P}^{(\mathcal{T})}_{\text{I}}(\boldsymbol{x}) + \tilde{P}^{(\mathcal{T})}_{\text{II}}(\boldsymbol{x})}, \tag{3.22}$$

Thus, we recover the expression in Equation (3.18). Their optimality can be confirmed via

$$\frac{\partial^2 \mathcal{L}_{\text{LBC}}}{\partial \hat{y}(\boldsymbol{x})^2} = \frac{\mathcal{T}_{\text{I}}(\boldsymbol{x})}{|\mathcal{T}|} \frac{1}{\hat{y}(\boldsymbol{x})^2} + \frac{\mathcal{T}_{\text{II}}(\boldsymbol{x})}{|\mathcal{T}|} \frac{1}{[1 - \hat{y}(\boldsymbol{x})]^2} > 0 \tag{3.23}$$

or

$$\frac{\partial^2 \mathcal{L}_{\text{MSE}}}{\partial \hat{y}(\boldsymbol{x})^2} = 2 \cdot \frac{\mathcal{T}_{\text{I}}(\boldsymbol{x}) + \mathcal{T}_{\text{II}}(\boldsymbol{x})}{|\mathcal{T}|} > 0, \tag{3.24}$$

in the case of a CE or MSE loss, respectively.

The value of the indicator in LBC for a given bipartition corresponds to the mean classification accuracy [Equation (2.11)], where the continuous predictions $\hat{y}(\boldsymbol{x}) \in [0, 1]$ are mapped to binary labels via $\Theta\left[\hat{y}(\boldsymbol{x}) - 0.5\right]$. Using the optimal prediction in Equation (3.22), the mean classification error for a given input $\boldsymbol{x}$ is $\min\{\hat{y}^{\text{opt}}_{\text{LBC}}(\boldsymbol{x}), 1 - \hat{y}^{\text{opt}}_{\text{LBC}}(\boldsymbol{x})\}$. Assuming that $\mathcal{T} = \mathcal{E}$ and weighting the contribution of each input $\boldsymbol{x}$ to the mean classification error by its probability $\tilde{P}^{(\mathcal{E})}(\boldsymbol{x}|\gamma)$, we arrive at Equation (3.20). One can show that Equation (3.19) and Equation (3.20) are equivalent under the assumption that $\mathcal{T} = \mathcal{E}$. Starting from Equation (3.19) and considering the optimal predictions, we have

$$I^{\text{opt}}_{\text{LBC}} = 1 - \frac{1}{|\Gamma|} \left(\sum_{\substack{\boldsymbol{x} \in \bar{\mathcal{E}} \\ \tilde{P}^{(\mathcal{T})}_{\text{I}}(\boldsymbol{x}) < \tilde{P}^{(\mathcal{T})}_{\text{II}}(\boldsymbol{x})}} \tilde{P}^{(\mathcal{E})}_{\text{I}}(\boldsymbol{x}) + \sum_{\substack{\boldsymbol{x} \in \bar{\mathcal{E}} \\ \tilde{P}^{(\mathcal{T})}_{\text{I}}(\boldsymbol{x}) > \tilde{P}^{(\mathcal{T})}_{\text{II}}(\boldsymbol{x})}} \tilde{P}^{(\mathcal{E})}_{\text{II}}(\boldsymbol{x}) \right) . \tag{3.25}$$

Similarly, starting from Equation (3.20), we obtain

$$\begin{aligned} I^{\text{opt}}_{\text{LBC}} = 1 - \frac{1}{|\Gamma|} \Bigg(& \sum_{\substack{\boldsymbol{x} \in \bar{\mathcal{E}} \\ \tilde{P}^{(\mathcal{T})}_{\text{I}}(\boldsymbol{x}) < \tilde{P}^{(\mathcal{T})}_{\text{II}}(\boldsymbol{x})}} \frac{\tilde{P}^{(\mathcal{T})}_{\text{I}}(\boldsymbol{x})}{\tilde{P}^{(\mathcal{T})}_{\text{I}}(\boldsymbol{x}) + \tilde{P}^{(\mathcal{T})}_{\text{II}}(\boldsymbol{x})} \left[\tilde{P}^{(\mathcal{E})}_{\text{I}}(\boldsymbol{x}) + \tilde{P}^{(\mathcal{E})}_{\text{II}}(\boldsymbol{x})\right] + \\ & \sum_{\substack{\boldsymbol{x} \in \bar{\mathcal{E}} \\ \tilde{P}^{(\mathcal{T})}_{\text{I}}(\boldsymbol{x}) > \tilde{P}^{(\mathcal{T})}_{\text{II}}(\boldsymbol{x})}} \frac{\tilde{P}^{(\mathcal{T})}_{\text{II}}(\boldsymbol{x})}{\tilde{P}^{(\mathcal{T})}_{\text{I}}(\boldsymbol{x}) + \tilde{P}^{(\mathcal{T})}_{\text{II}}(\boldsymbol{x})} \left[\tilde{P}^{(\mathcal{E})}_{\text{I}}(\boldsymbol{x}) + \tilde{P}^{(\mathcal{E})}_{\text{II}}(\boldsymbol{x})\right] \Bigg). \end{aligned} \tag{3.26}$$

The two expressions are equivalent given that $\mathcal{T} = \mathcal{E}$.

3.2.3 Prediction-based method

The optimal predictions within PBM (see Section 2.5.3) for a given $\boldsymbol{x} \in \bar{\mathcal{T}}$ are

$$\hat{y}_{\mathrm{PBM}}^{\mathrm{opt}}\left(\boldsymbol{x}\right) = \frac{\sum_{\gamma \in \Gamma} \tilde{P}^{(\mathcal{T})}\left(\boldsymbol{x}|\gamma\right)\gamma}{\sum_{\gamma \in \Gamma} \tilde{P}^{(\mathcal{T})}\left(\boldsymbol{x}|\gamma\right)}. \tag{3.27}$$

See below for a proof. Here, the optimal prediction for a given input is obtained by a weighted sum over each point in the parameter range, where the weight of each point γ corresponds to the probability of obtaining the input at that point along the parameter range compared to all other points. Therefore, the prediction accuracy decreases if the same input can be drawn at multiple values of the tuning parameter, i.e., when the underlying probability distributions overlap. An expression for the optimal value of the loss in PBM, $\mathcal{L}_{\mathrm{PBM}}^{\mathrm{opt}}$, can be obtained by replacing $\hat{y}(\boldsymbol{x})$ by $\hat{y}_{\mathrm{PBM}}^{\mathrm{opt}}(\boldsymbol{x})$ in Equation (2.12). The mean prediction of an optimal model at a sampled point γ is given by

$$\hat{y}_{\mathrm{PBM}}^{\mathrm{opt}}(\gamma) = \sum_{\boldsymbol{x} \in \bar{\mathcal{E}}_{\gamma}} \tilde{P}^{(\mathcal{E})}(\boldsymbol{x}|\gamma)\hat{y}_{\mathrm{PBM}}^{\mathrm{opt}}(\boldsymbol{x}). \tag{3.28}$$

Thus, the optimal indicator for phase transitions is

$$I_{\mathrm{PBM}}^{\mathrm{opt}}(\gamma) = \left.\frac{\partial \delta y_{\mathrm{PBM}}^{\mathrm{opt}}(\gamma)}{\partial \gamma}\right|_{\gamma}, \tag{3.29}$$

where $\delta y_{\mathrm{PBM}}^{\mathrm{opt}}(\gamma) = \hat{y}_{\mathrm{PBM}}^{\mathrm{opt}}(\gamma) - \gamma$. Recall that in PBM, phase transitions are detected by analyzing the dependence of the prediction error on the tuning parameter. The optimal indicator [Equation (3.29)] highlights the value of the tuning parameter at which the mean predictions change most, i.e., where the overlap of the underlying probability distributions changes most. The optimal predictions and indicators of PBM have previously been derived in [Arnold *et al.*, 2021] but have neither been utilized in a numerical routine, nor been used to explain previous studies.

Proof

In PBM, a predictive model is trained to minimize the MSE loss function $\mathcal{L}_{\mathrm{PBM}}$ specified in Equation (2.12). Consider a particular input $\boldsymbol{x}' \in \bar{\mathcal{T}}$. We can determine the optimal model prediction $\hat{y}_{\mathrm{PBM}}^{\mathrm{opt}}(\boldsymbol{x}')$ for this input by minimizing the loss function in Equation (2.12) with respect to $\hat{y}(\boldsymbol{x}')$, i.e., by solving

$$\frac{\partial \mathcal{L}_{\mathrm{PBM}}}{\partial \hat{y}(\boldsymbol{x}')} = \frac{2}{|\Gamma|} \sum_{\gamma \in \Gamma} \tilde{P}^{(\mathcal{T})}(\boldsymbol{x}'|\gamma) \left[\hat{y}(\boldsymbol{x}') - \gamma\right] = 0. \tag{3.30}$$

This yields

$$\hat{y}_{\mathrm{PBM}}^{\mathrm{opt}}(\boldsymbol{x}') = \frac{\sum_{\gamma \in \Gamma} \tilde{P}^{(\mathcal{T})}(\boldsymbol{x}'|\gamma)\gamma}{\sum_{\gamma \in \Gamma} \tilde{P}^{(\mathcal{T})}(\boldsymbol{x}'|\gamma)}. \tag{3.31}$$

This prediction is indeed optimal, as

$$\frac{\partial^2 \mathcal{L}_{\mathrm{PBM}}}{\partial \hat{y}(\boldsymbol{x}')^2} = \frac{2}{|\Gamma|} \sum_{\gamma \in \Gamma} \tilde{P}^{(\mathcal{T})}(\boldsymbol{x}'|\gamma) > 0. \tag{3.32}$$

Repeating this procedure for all available inputs $\boldsymbol{x} \in \bar{\mathcal{T}}$ yields Equation (3.27).

We note in passing that this derivation can be generalized to higher dimensional parameter spaces in a straightforward manner following Arnold *et al.* [2021], resulting in

$$\hat{\boldsymbol{y}}^{\mathrm{opt}}_{\mathrm{PBM}}(\boldsymbol{x}) = \frac{\sum_{\boldsymbol{\gamma} \in \Gamma} \tilde{P}^{(\mathcal{T})}(\boldsymbol{x}|\boldsymbol{\gamma})\boldsymbol{\gamma}}{\sum_{\boldsymbol{\gamma} \in \Gamma} \tilde{P}^{(\mathcal{T})}(\boldsymbol{x}|\boldsymbol{\gamma})}. \tag{3.33}$$

Here, the sum runs over all sampled points $\boldsymbol{\gamma}$ in parameter space. The optimal indicator is then given as a divergence

$$I^{\mathrm{opt}}_{\mathrm{PBM}}(\boldsymbol{\gamma}) = \nabla_{\boldsymbol{\gamma}} \boldsymbol{\delta y}^{\mathrm{opt}}_{\mathrm{PBM}}(\boldsymbol{\gamma}), \tag{3.34}$$

where $\boldsymbol{\delta y}^{\mathrm{opt}}_{\mathrm{PBM}}(\boldsymbol{\gamma}) = \sum_{\boldsymbol{x} \in \bar{\mathcal{E}}} \tilde{P}^{(\mathcal{E})}(\boldsymbol{x}|\boldsymbol{\gamma}) \hat{\boldsymbol{y}}^{\mathrm{opt}}_{\mathrm{PBM}}(\boldsymbol{x}) - \boldsymbol{\gamma}$.

3.2.4 Discussion

The empirically optimal predictions of SL, LBC, and PBM can be expressed *solely* in terms of the probability distributions $\{\tilde{P}^{(\mathcal{T}/\mathcal{E})}(\cdot|\gamma)\}_{\gamma \in \Gamma}$ governing the input data, i.e., the empirical distributions underlying the training and evaluation set, respectively.[3] In case we have access to the exact underlying probability distributions $\{P(\cdot|\gamma)\}_{\gamma \in \Gamma}$ in terms of numerical values or analytical expressions, we may utilize them instead ($\tilde{P}^{(\mathcal{T}/\mathcal{E})} \mapsto P$) to compute Bayes-optimal predictions.

Notice that the optimal predictions – and thus the optimal indicators of phase transitions – only depend on the input through its probability and are thus not explicitly dependent on the particular nature of an input or how similar it is to other inputs. Such notions of similarity form the basis of a large set of other phase-classification methods, e.g., based on principal component analysis [Wang, 2016], diffusion maps [Rodriguez-Nieva and Scheurer, 2019], or anomaly detection [Kottmann *et al.*, 2020]. The analytical form of the optimal predictions indicates that SL, LBC, and PBM ultimately gauge changes in the probability distributions governing the data akin to statistical distances.[4] In particular, the optimal predictions and indicators are invariant under transformations of the input data with bijective functions. One may also use knowledge of the system's underlying symmetries to group input data together and calculate indicators of phase transitions more efficiently. We will make use of this in Section 3.5.

Splitting data into training, validation, and test sets

Throughout this thesis, if not stated otherwise, when using PBM and LBC we are not going to explicitly split the data set $\mathcal{D}$ into a training, validation, and test set. Instead, we will use all the available data for training and evaluation, i.e., $\mathcal{D} = \mathcal{T} = \mathcal{E}$ with $\mathcal{V} = \{\}$. Similarly, in SL, we will generally use all the available data for evaluation $\mathcal{D} = \mathcal{E}$ with $\mathcal{V} = \{\}$, and use all the data in regions I and II for training

[3] We will explore the use of approximate distributions other than the empirical one in Chapter 4.

[4] This connection will be made more rigorous in Chapter 6.

$\mathcal{T} = \biguplus_{\gamma \in \Gamma_{\text{I}} \cup \Gamma_{\text{II}}} \mathcal{D}_\gamma$. Looking back at the expressions for the optimal predictions and indicators in Section 3.2, we may thus write $\tilde{P}^{(\mathcal{E})} = \tilde{P}^{(\mathcal{T})} = \tilde{P}^{(\mathcal{D})} = \tilde{P}$, i.e., whenever we encounter an empirical probability distribution it will be constructed based on all the available data.

As we discussed in Section 2.5, in many standard applications of NNs it is typical to split the available data into multiple sets, in particular to ensure good performance on unseen data [Goodfellow *et al.*, 2016]. For example, suppose we aim to construct an accurate on-the-fly classifier of individual samples into distinct phases of matter that can handle samples not contained within the training set. In this case, it may be beneficial to split the available data into a training and validation set to avoid overfitting if only a limited amount of data is available.

In this thesis, however, we are ultimately interested in the detection of phase transitions given the data at hand. While we use predictive models for that, we are not explicitly interested in them performing well on unseen data. Rather, we want to leverage the dataset $\mathcal{D}$ we have as best as we can to accurately estimate the critical point. As such, the data set does not *necessarily* need to be split. However, it can still be useful to perform early stopping with NNs as a regularization technique, see Section 3.7.5 for concrete examples. Similarly, one may want to introduce a splitting to assess sampling convergence by comparing the predictions obtained on the training set and test set.

In the limit of infinite samples, all splits of a dataset will be identical (assuming that all samples are drawn independently from the same probability distributions underlying the physical system, see Figure 3.1). Therefore, the predictions and indicators obtained by training NNs or analytical construction using multiple distinct data sets will coincide with the values obtained using the entire data set for training and evaluation up to deviations arising from finite-sample statistics. That is, in the limit of a sufficient number of samples, the results obtained in the two scenarios coincide [Blumer *et al.*, 1989; Vapnik, 1999; Goodfellow *et al.*, 2016]. And, given a fixed amount of data, better statistics may be achieved by utilizing the entire data for training and evaluation.

3.3 Demonstration on prototypical probability distributions

In this section, we compute the Bayes-optimal indicators of SL, LBC, and PBM for a set of simple probability distributions governing the input data. As we will see later, the probability distributions governing the data in physical systems can be regarded as generalizations of the special cases discussed in this section. Thus, they serve as a reasonable basis for understanding. We compare these results to the indicators obtained by numerical optimization of NNs. The details on the NN architecture and training, including the corresponding hyperparameters, will be discussed in Section 3.7.1. This first demonstration shows how the analytical expressions can be used to calculate the optimal indicator directly from input data without NNs. Moreover, it confirms that the optimal predictive models can be recovered by training NNs with sufficient expressive power.

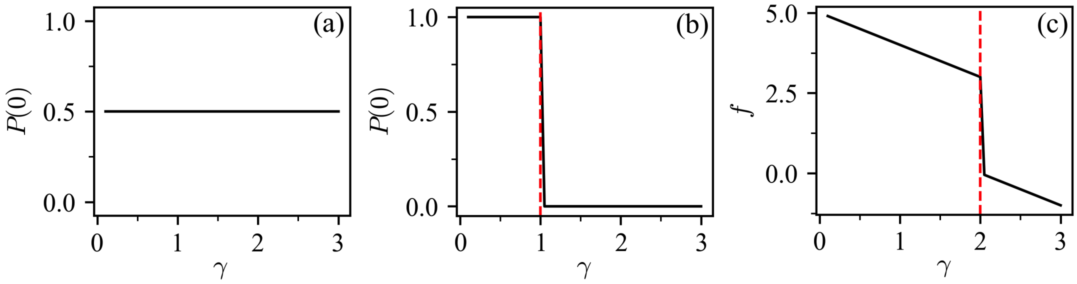

FIGURE 3.2: Prototypical probability distributions for demonstrating the working principle of SL, LBC, and PBM. (a) Case 1 where $\forall\gamma \in \Gamma P(\cdot|\gamma) = P(\cdot)$ with $P(0) = P(1) = 0.5$. (b) Case 2 given by Equation (3.36) with $P_{\mathrm{A}}(0) = 1$, $P_{\mathrm{B}}(0) = 0$ and $\gamma_{\mathrm{c}} = 1$. (c) Case 3 described by Equations (3.43)-(3.44). The tuning parameter ranges from $\gamma = 0.1$ to $\gamma = 3$ with $\Delta\gamma = 0.05$. Critical values of the tuning parameter are highlighted with red dashed lines.

Case 1

Let us first consider the case where the probability distribution governing the data is identical across the parameter range, i.e., for any $\gamma \in \Gamma$, $P(\cdot|\gamma) = P(\cdot)$ [see Figure 3.2(a)]. Clearly, in this case, all three methods should indicate the presence of a single phase. The (Bayes) optimal prediction in SL is

$$\hat{y}_{\mathrm{SL}}^{\mathrm{opt}}(\gamma) = \frac{|\Gamma_{\mathrm{I}}|}{|\Gamma_{\mathrm{I}}| + |\Gamma_{\mathrm{II}}|} = \text{const.}, \tag{3.35}$$

corresponding to the relative size of region I compared to region II [see Figure 3.3(a)]. Taking the derivative of Equation (3.35) results in a flat indicator signal $I_{\mathrm{SL}}^{\mathrm{opt}} = 0$. In LBC, the (Bayes) optimal classification accuracy for a particular bipartition is given by $I_{\mathrm{LBC}}^{\mathrm{opt}} = \max\{|\Gamma_{\mathrm{I}}|/|\Gamma|, |\Gamma_{\mathrm{II}}|/|\Gamma|\}$. This results in a characteristic V-shape [Van Nieuwenburg *et al.*, 2017], which has its minimum at the center of the parameter range under consideration, see Figure 3.3(d). In PBM, the (Bayes) optimal mean prediction is also placed at the center of mass $\hat{y}_{\mathrm{PBM}}^{\mathrm{opt}}(\gamma) = 1/|\Gamma| \sum_{\gamma\in\Gamma} \gamma = \text{const.}$, which results in a constant indicator $I_{\mathrm{PBM}}^{\mathrm{opt}} = -1$ [see Figure 3.3(g)]. As such, all three methods yield optimal indicators that correctly signal the presence of a single phase, i.e., the absence of two distinct phases. For a concrete numerical demonstration, we consider the case of binary inputs $\mathcal{X} = \{0, 1\}$ with equal probability $P(0) = P(1) = 0.5$. Figures 3.3(a),(d),(g), and (j) show the results for all three methods using the analytical expressions as well as NNs. Note that the analytical predictions and indicators can be approximated well using NNs as predictive models.

Case 2

Next, we consider the case where the input data naturally separates into two distinct sets. That is, the underlying probability distributions result in a bipartition of the parameter range into two regions A and B, with sampled points Γ_{A} and Γ_{B}, where each input can only be drawn in one of the two regions. In these regions, we choose the probability distributions to be identical

$$P(\cdot|\gamma) = \begin{cases} P_{\mathrm{A}}(\cdot) \text{ if } \gamma \leq \gamma_{\mathrm{c}}, \\ P_{\mathrm{B}}(\cdot) \text{ otherwise.} \end{cases} \tag{3.36}$$

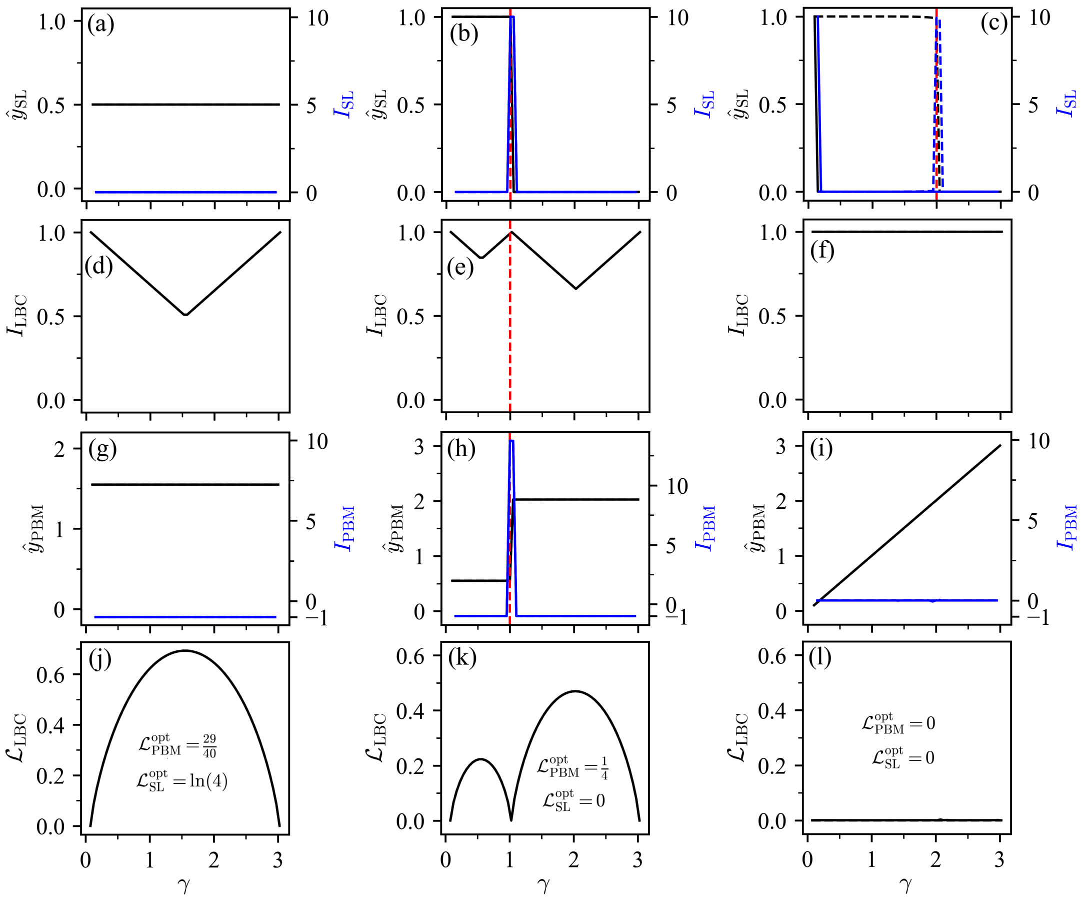

FIGURE 3.3: Results for the prototypical probability distributions depicted in Figure 3.2 with (a),(d),(g),(j) corresponding to case 1, (b),(e),(h),(k) to case 2, and (c),(f),(i),(l) to case 3. Critical values of the tuning parameter are highlighted with red dashed lines. For details on SL, LBC, and PBM using NNs, see Section 3.7.1. (a)-(c) Mean prediction $\hat{y}_{\mathrm{SL}}$ obtained using the analytical expression (black, solid) or an NN (black, dashed), as well as the corresponding indicator I_{SL} (blue). Here, we choose $r_{\mathrm{I}} = 1$ and $l_{\mathrm{II}} = |\Gamma|$. (d)-(f) The indicator of LBC, I_{LBC}, obtained using the analytical expression (black, solid) or an NN (black, dashed). (g)-(i) Mean prediction $\hat{y}_{\mathrm{PBM}}$ of PBM obtained using the analytical expression (black, solid) or an NN (black, dashed), as well as the corresponding indicator I_{PBM} (blue). (j)-(l) Value of the loss function in LBC, $\mathcal{L}_{\mathrm{LBC}}$, for each bipartition point γ^{bp} obtained using the analytical expression (black, solid) or evaluated after NN training (black, dashed). In addition, the optimal values of the loss function for SL and PBM obtained by evaluating the analytical expressions are reported. Note that, by definition, $\mathcal{L}^{\mathrm{opt}} \leq \mathcal{L}$ for all three methods.

This is a prototypical example for the case where the physical system transitions from phase A to B when crossing a critical value of the tuning parameter γ_{c} [see Figure 3.2(b)]. Here, γ_{c} corresponds to a sampled value of the tuning parameter, which may, in general, not be the case.

Using SL, the (Bayes) optimal strategy corresponds to

$$\hat{y}^{\mathrm{opt}}_{\mathrm{SL}}(\gamma) = \begin{cases} 1 \text{ if } \gamma \leq \gamma_{\mathrm{c}}, \\ 0 \text{ otherwise.} \end{cases} \tag{3.37}$$

This results in

$$I_{\mathrm{SL}}^{\mathrm{opt}}(\gamma) = \begin{cases} 0 \text{ if } \gamma < \gamma_{\mathrm{c}}, \\ \frac{1}{2\Delta\gamma} \text{ if } \gamma \in \{\gamma_{\mathrm{c}}, \gamma_{\mathrm{c}} + \Delta\gamma\}, \\ 0 \text{ otherwise}, \end{cases} \tag{3.38}$$

which diverges as $\Delta\gamma \to 0$ and exhibits a peak at the two points that constitute the boundary between regions A and B [see Figure 3.3(b)]. Here, we approximate the derivative in Equation (3.5) by a symmetric difference quotient

$$I_{\mathrm{SL}}^{\mathrm{opt}}(\gamma) \approx \frac{|\hat{y}_{\mathrm{SL}}^{\mathrm{opt}}(\gamma + \Delta\gamma) - \hat{y}_{\mathrm{SL}}^{\mathrm{opt}}(\gamma - \Delta\gamma)|}{2\Delta\gamma}, \tag{3.39}$$

where we ignore the two points γ at the edges of the sampled interval Γ.

In LBC, one can reach a perfect (error-free) classification when matching the natural bipartition present in the data. Let us denote the region between the bipartition point underlying the data, γ_{c}, and the chosen bipartition point in the LBC scheme, γ^{bp}, as III. The number of sampled parameter values within the smallest region between I, II, and III is $\min\{|\Gamma_{\mathrm{I}}|, |\Gamma_{\mathrm{II}}|, |\Gamma_{\mathrm{III}}|\}$. Note that all input data drawn within one of these regions must be misclassified. Thus, the optimal strategy that yields the smallest classification error corresponds to misclassifying all input data drawn within the smallest region. The (Bayes) optimal classification accuracy is then given as

$$I_{\mathrm{LBC}}^{\mathrm{opt}}(\gamma^{\mathrm{bp}}) = 1 - \frac{\min\{|\Gamma_{\mathrm{I}}|, |\Gamma_{\mathrm{II}}|, |\Gamma_{\mathrm{III}}|\}}{|\Gamma|}. \tag{3.40}$$

This results in the characteristic W-shape of the indicator [Van Nieuwenburg *et al.*, 2017], see Figure 3.3(e), where the middle-peak occurs at the bipartition point γ^{bp} closest to γ_{c}.

In PBM, we have

$$\hat{y}_{\mathrm{PBM}}^{\mathrm{opt}}(\gamma) = \begin{cases} \langle\gamma\rangle_{\mathrm{A}} = \frac{1}{|\Gamma_{\mathrm{A}}|} \sum_{\gamma \in \Gamma_{\mathrm{A}}} \gamma, \\ \langle\gamma\rangle_{\mathrm{B}} = \frac{1}{|\Gamma_{\mathrm{B}}|} \sum_{\gamma \in \Gamma_{\mathrm{B}}} \gamma, \end{cases} \tag{3.41}$$

where $\langle\gamma\rangle_{\mathrm{A/B}}$ denotes the center of region A and B, respectively. This results in

$$I_{\mathrm{PBM}}^{\mathrm{opt}}(\gamma) = \begin{cases} -1 \text{ if } \gamma < \gamma_{\mathrm{c}}, \\ \frac{\langle\gamma\rangle_{\mathrm{B}} - \langle\gamma\rangle_{\mathrm{A}}}{2\Delta\gamma} \text{ if } \gamma \in \{\gamma_{\mathrm{c}}, \gamma_{\mathrm{c}} + \Delta\gamma\}, \\ -1 \text{ otherwise}, \end{cases} \tag{3.42}$$

where we approximated the derivative in Equation (3.29) by a symmetric difference quotient [see Figure 3.3(h)]. The expression in Equation (3.42) diverges as $\Delta\gamma \to 0$ for $\gamma \in \{\gamma_{\mathrm{c}}, \gamma_{\mathrm{c}} + \Delta\gamma\}$ and results in a peak at the two points which constitute the boundary between regions A and B. As such, the optimal indicators of all three methods correctly indicate the presence of two distinct sets of data, i.e., two distinct phases. The results obtained using the analytical expressions can be approximated well using NNs as predictive models. This is illustrated in Figures 3.3(b),(e),(h), and (k), where we consider the special case of binary inputs with $P_{\mathrm{A}}(0) = 1$, $P_{\mathrm{B}}(0) = 0$, and $\gamma_{\mathrm{c}} = 1$.

Case 3

Lastly, we consider the case where the probability distributions underlying the data do not overlap, i.e., the probability of drawing a given input at two distinct values of the tuning parameter vanishes. In particular, this situation can occur when dealing with large state spaces, which are prone to result in insufficient sampling statistics in practice. That is, even in scenarios where the ground-truth probability distributions underlying the data *do* overlap, the estimated probabilities $\tilde{P}(\boldsymbol{x}|\gamma) \approx \mathcal{D}_\gamma(\boldsymbol{x})/|\mathcal{D}_\gamma|$ based on the drawn data set $\mathcal{D}$ may not (see Section 3.6.5 for a concrete physical example). Many image classification tasks encountered in traditional ML applications [Fei-Fei *et al.*, 2004; LeCun *et al.*, 2004; Griffin *et al.*, 2007; Krizhevsky, 2009; Deng *et al.*, 2009; Russakovsky *et al.*, 2015; Spanhol *et al.*, 2016] *a priori* fall into this category. In particular, the probability distributions underlying the data are typically not known in these cases. Therefore, constructing optimal models, in particular Bayes-optimal models, largely remains conceptual in nature [Devroye *et al.*, 1996; James *et al.*, 2013].

Imagine trying to classify images of cats and dogs. You will rarely encounter the exact same image (with the exact same pixel values) twice in your dataset. Hence, trying to estimate the underlying distribution from counting (as we do here) is hopeless. In fact, the distribution underlying the dataset is largely inaccessible and abstract. We do not know them (or even their form) *a priori*. This changes when considering data from the domain of statistical and quantum physics, which is the key fact we exploit in this chapter to gain additional insights into the underlying ML methods and come up with more efficient numerical routines.

In case 3, a Bayes-optimal predictive model is capable of distinguishing between samples obtained at distinct values of the tuning parameter with perfect accuracy. This results in $I^{\text{opt}}_{\text{LBC}}(\gamma^{\text{bp}}) = 1$ for LBC [see Figure 3.3(f)]. In the case of PBM, we have $\hat{y}^{\text{opt}}_{\text{PBM}}(\gamma) = \gamma$ such that $I^{\text{opt}}_{\text{PBM}}(\gamma) = -1$, see Figure 3.3(i). In both cases, the indicator signals the absence of two distinct sets of data, i.e., phases. The optimal predictions of SL for $\boldsymbol{x} \in \bar{\mathcal{E}}$ are underdetermined: only the predictions for inputs within the training data $\boldsymbol{x} \in \bar{\mathcal{T}}$ are fixed after training and the assumption that $\bar{\mathcal{T}} = \bar{\mathcal{E}}$ is violated in this particular case [see Figure 3.3(c)]. Note, however, that the predictions are, in principle, also unconstrained when using SL with NNs. For a simple numerical example, we consider the case where a single unique (scalar) input is drawn at each point along the parameter range

$$P(x|\gamma) = \begin{cases} 1 \text{ if } x = f(\gamma), \\ 0 \text{ otherwise}, \end{cases} \tag{3.43}$$

with

$$f(\gamma) = \begin{cases} 5 - \gamma \text{ if } \gamma \leq 2, \\ 2 - \gamma \text{ otherwise}. \end{cases} \tag{3.44}$$

The results are shown in Figures 3.3(c),(f),(i), and (l). In practice, NNs will tend to predict similar outputs for similar inputs. The interpolating nature of the NN results in SL highlighting the value of the tuning parameter $\gamma = 2$ where a discontinuity in the input data is present. We also observe this tendency for the NNs in LBC and PBM during training.

3.4 Computational complexity

We can use the analytical expressions derived in Section 3.2 to assess the computational cost associated with the evaluation of the mean optimal predictions and optimal indicators of SL, LBC, and PBM for a given set of input data. In our estimation, we neglect the overhead arising from the computation of the probability distributions $\{P(\cdot|\gamma)\}_{\gamma\in\Gamma}$, or $\{\tilde{P}(\cdot|\gamma)\}_{\gamma\in\Gamma}$, which is identical for all three methods. We will also ignore any other constant overhead and factors. The computation of the optimal predictions and indicators can be approached in two ways: Either the optimal predictions for a given input $\hat{y}^{\text{opt}}(\boldsymbol{x})$ are recomputed in each function call, or they are cached. We report the required number of floating-point operations in both instances, which can be counted based on the analytical expressions reported in Section 3.2. This counting represents a rough, hardware-independent estimate of the required computational cost. For computation times measured on hardware, see Table 3.1. In the following, we will assume that the optimal indicators in SL and PBM are computed using a symmetric difference quotient, cf. Equation (3.39).

Supervised learning

The computation of $\hat{y}_{\text{SL}}^{\text{opt}}$ for all $\boldsymbol{x} \in \bar{\mathcal{D}}$ requires $|\bar{\mathcal{D}}|(|\Gamma_{\text{I}}| + |\Gamma_{\text{II}}|)$ floating-point operations. Note the appearance of $|\bar{\mathcal{D}}|$ which can result in an exponential scaling for quantum problems due to the exponential growth of the Hilbert space $\mathcal{H}$ (and thus the state space). Caching the values of $\hat{y}_{\text{SL}}^{\text{opt}}$ for all $\boldsymbol{x} \in \bar{\mathcal{D}}$, the number of operations required to compute the mean optimal prediction $\hat{y}_{\text{SL}}^{\text{opt}}$ for all $\{\gamma\}_{\gamma\in\Gamma}$ is $|\Gamma|(2|\bar{\mathcal{D}}| - 1) + |\bar{\mathcal{D}}|(|\Gamma_{\text{I}}| + |\Gamma_{\text{II}}|)$. Thus, computing the optimal indicator requires $|\bar{\mathcal{D}}|(2|\Gamma| + |\Gamma_{\text{I}}| + |\Gamma_{\text{II}}|) + |\Gamma|$ operations. Typically, in SL we have $|\Gamma_{\text{I}}| + |\Gamma_{\text{II}}| \ll |\Gamma|$. Under this assumption, the computation of the mean optimal predictions and the optimal indicators each require $O(|\bar{\mathcal{D}}||\Gamma|)$ operations.[5] If the values $\hat{y}_{\text{SL}}^{\text{opt}}(\boldsymbol{x})$ are not cached for all $\boldsymbol{x} \in \bar{\mathcal{D}}$, computing the mean optimal prediction instead requires $|\Gamma| \left[(2|\bar{\mathcal{D}}| - 1) + |\bar{\mathcal{D}}|(|\Gamma_{\text{I}}| + |\Gamma_{\text{II}}|)\right]$ operations. Computing the optimal indicator then requires $|\bar{\mathcal{D}}||\Gamma|(2 + |\Gamma_{\text{I}}| + |\Gamma_{\text{II}}|) + |\Gamma|$ operations. For both quantities, this corresponds to $O(|\bar{\mathcal{D}}||\Gamma|)$ operations.

Learning by confusion

The computation of $\hat{y}_{\text{LBC}}^{\text{opt}}$ for all $\boldsymbol{x} \in \bar{\mathcal{D}}$ requires $|\bar{\mathcal{D}}||\Gamma|$ floating-point operations. Caching these values, the number of operations required to compute the optimal indicator is $|\bar{\mathcal{D}}||\Gamma|^2(F_{\min}+2)$, where $F_{\min}$ denotes the number of floating-point operations required to compute $\min\{\hat{y}_{\text{LBC}}^{\text{opt}}(\boldsymbol{x}), 1-\hat{y}_{\text{LBC}}^{\text{opt}}(\boldsymbol{x})\}$. This corresponds to $O(|\bar{\mathcal{D}}||\Gamma|^2)$ operations. Without caching, the optimal indicator requires $|\bar{\mathcal{D}}||\Gamma|^3+|\bar{\mathcal{D}}||\Gamma|^2(F_{\min}+2)+|\Gamma|$ operations to compute, resulting in a cost of $O(|\bar{\mathcal{D}}||\Gamma|^3)$.

Prediction-based method

In PBM, the computation of $\hat{y}_{\text{PBM}}^{\text{opt}}$ for all $\boldsymbol{x} \in \bar{\mathcal{D}}$ requires $|\bar{\mathcal{D}}|(3|\Gamma| - 1)$ floating-point operations. Caching these values, the number of operations required to compute the mean optimal prediction $\hat{y}_{\text{PBM}}^{\text{opt}}$ for all $\gamma \in \Gamma$ is $5|\bar{\mathcal{D}}||\Gamma| - |\Gamma| - |\bar{\mathcal{D}}|$. Computing the optimal indicator then requires $|\bar{\mathcal{D}}|(5|\Gamma|-1)+|\Gamma|$ operations. Hence, the computations of the mean optimal predictions and the optimal indicator each require $O(|\bar{\mathcal{D}}||\Gamma|)$

[5] Throughout this thesis, the use of O indicates big O notation. To denote remainder terms, e.g., in Taylor expansions, we use $\mathcal{O}$ instead.

operations. If the values $\hat{y}^{\mathrm{opt}}_{\mathrm{PBM}}(\boldsymbol{x})$ are not cached for all $\boldsymbol{x} \in \bar{\mathcal{D}}$, computing the mean optimal prediction instead requires $3|\bar{\mathcal{D}}||\Gamma|^2 + |\Gamma|(|\bar{\mathcal{D}}| - 1)$ operations. Computing the optimal indicator then requires $3|\bar{\mathcal{D}}||\Gamma|^2 + |\Gamma||\bar{\mathcal{D}}| + |\Gamma|$ operations. For both quantities, this results in a scaling of $O(|\bar{\mathcal{D}}||\Gamma|^2)$.

3.5 Application to physical systems

In this section, we will discuss how one can compute the optimal predictions and indicators of phase transitions of SL, LBC, and PBM, using the analytical expressions introduced in Section 3.2 for the Ising model, Ising gauge theory, XY model, XXZ model, Kitaev model, and Bose-Hubbard model. The results will be presented in Section 3.6.

3.5.1 Classical equilibrium systems

In this chapter, we will study the Ising model, Ising gauge theory, and XY model as examples of classical equilibrium systems. For each of these models, we sample 10^5 spin configurations from a thermal distribution at each temperature T using the Metropolis-Hastings algorithm [Metropolis *et al.*, 1953].[6] Here, the temperature serves as a tuning parameter. The probability that a system in equilibrium at inverse temperature $\beta = 1/k_{\mathrm{B}}T$ is found in a state with spin configuration $\boldsymbol{\sigma}$ is given by a Boltzmann distribution

$$P(\boldsymbol{\sigma}|T) = \frac{e^{-H(\boldsymbol{\sigma})/k_{\mathrm{B}}T}}{Z_T}, \tag{3.45}$$

where $Z_T = \sum_{\boldsymbol{\sigma} \in \mathcal{X}} e^{-H(\boldsymbol{\sigma})/k_{\mathrm{B}}T}$ is the partition function and H is the respective system Hamiltonian. In principle, one could use the raw spin configurations as input, i.e., estimate the underlying probability distributions as $P(\boldsymbol{\sigma}|T) \approx \tilde{P}(\boldsymbol{\sigma}|T) = \mathcal{D}_T(\boldsymbol{\sigma})/|\mathcal{D}_T|$. However, the probability of drawing a particular spin configuration only depends on its energy [see Equation (3.45)]. One can show that the Bayes-optimal predictions and indicators remain identical when the energy is used as input instead of the raw configurations, i.e., when the probability distributions governing the data are given by

$$P(E|T) = \frac{g(E)e^{-E/k_{\mathrm{B}}T}}{Z_T}, \tag{3.46}$$

where $g(E)$ is the degeneracy factor.

Proof

We consider the case where the drawn inputs $\boldsymbol{x}$, such as spin configurations, follow a Boltzmann distribution

$$P(\boldsymbol{x}|T) = \frac{e^{-H(\boldsymbol{x})/k_{\mathrm{B}}T}}{Z_T}. \tag{3.47}$$

[6] In this chapter, we analyze systems on lattices with linear size of at maximum $L = 60$. For this system size, we find this number of samples to be sufficient to accurately determine the corresponding empirically optimal predictions and indicators. That is, the empirically optimal results are close to being Bayes-optimal.

The probability to draw a sample with energy E is thus given by

$$P(E|T) = \frac{g(E)e^{-E/k_{\mathrm{B}}T}}{Z_T}, \tag{3.48}$$

where $g(E)$ is the corresponding degeneracy factor

$$g(E) = \sum_{\boldsymbol{x}\in\mathcal{X}} \delta_{H(\boldsymbol{x}),E}. \tag{3.49}$$

Here, $\mathcal{X}$ denotes the state space of the samples $\boldsymbol{x}$, i.e., the set of all unique samples without duplicates. Therefore, we have

$$P(\boldsymbol{x}|T) = P\left(H(\boldsymbol{x})|T\right)/g\left(H(\boldsymbol{x})\right). \tag{3.50}$$

Supervised learning.—Plugging Equation (3.50) into Equation (3.1), we immediately find that for any $\boldsymbol{x} \in \mathcal{X}$,

$$\begin{aligned} \hat{y}_{\mathrm{SL}}^{\mathrm{opt}}(\boldsymbol{x}) &= \frac{P_{\mathrm{I}}(H(\boldsymbol{x}))}{P_{\mathrm{I}}(H(\boldsymbol{x})) + P_{\mathrm{II}}(H(\boldsymbol{x}))} \\ &= \hat{y}_{\mathrm{SL}}^{\mathrm{opt}}(H(\boldsymbol{x})), \end{aligned} \tag{3.51}$$

where we assume that $\bar{\mathcal{T}} = \bar{\mathcal{D}} = \mathcal{X}$. Using Equation (3.4), we have

$$\begin{aligned} \hat{y}_{\mathrm{SL}}^{\mathrm{opt}}(T) &= \sum_{\boldsymbol{x}\in\mathcal{X}} P(\boldsymbol{x}|T)\hat{y}_{\mathrm{SL}}^{\mathrm{opt}}(\boldsymbol{x}) \\ &= \sum_{\boldsymbol{x}\in\mathcal{X}} P(H(\boldsymbol{x})|T)\hat{y}_{\mathrm{SL}}^{\mathrm{opt}}(H(\boldsymbol{x}))/g(H(\boldsymbol{x})) \\ &= \sum_{E\in\mathcal{X}_E} P(E|T)\hat{y}_{\mathrm{SL}}^{\mathrm{opt}}(E), \end{aligned} \tag{3.52}$$

where $\mathcal{X}_E$ is the set of unique energies corresponding to the state space $\mathcal{X}$. To obtain an expression for the optimal loss, we can rewrite Equation (2.7) as

$$\begin{aligned} \mathcal{L}_{\mathrm{SL}} = - &\frac{1}{|\Gamma_{\mathrm{I}}| + |\Gamma_{\mathrm{II}}|} \sum_{T\in\Gamma_{\mathrm{I}}\cup\Gamma_{\mathrm{II}}} \sum_{\boldsymbol{x}\in\mathcal{X}} P(\boldsymbol{x}|T) \\ &\Big(y(\boldsymbol{x}) \ln\left[\hat{y}_{\mathrm{SL}}(\boldsymbol{x})\right] + \left[1 - y(\boldsymbol{x})\right] \ln\left[1 - \hat{y}_{\mathrm{SL}}(\boldsymbol{x})\right]\Big). \end{aligned} \tag{3.53}$$

Using Equation (3.51), we have

$$\begin{aligned} \mathcal{L}_{\mathrm{SL}}^{\mathrm{opt}} = - &\frac{1}{|\Gamma_{\mathrm{I}}| + |\Gamma_{\mathrm{II}}|} \sum_{T\in\Gamma_{\mathrm{I}}\cup\Gamma_{\mathrm{II}}} \sum_{\boldsymbol{x}\in\mathcal{X}} P(H(\boldsymbol{x})|T) \\ &\Big(y(H(\boldsymbol{x})) \ln\left[\hat{y}_{\mathrm{SL}}^{\mathrm{opt}}(H(\boldsymbol{x}))\right] + \left[1 - y(H(\boldsymbol{x}))\right] \ln\left[1 - \hat{y}_{\mathrm{SL}}^{\mathrm{opt}}(H(\boldsymbol{x}))\right]\Big), \end{aligned} \tag{3.54}$$

where we use the fact that $y(\boldsymbol{x}) = y(H(\boldsymbol{x}))$, i.e., the assigned labels remain identical. Equation (3.54) can be simplified to

$$\begin{aligned}\mathcal{L}_{\mathrm{SL}}^{\mathrm{opt}} = &-\frac{1}{|\Gamma_{\mathrm{I}}| + |\Gamma_{\mathrm{II}}|} \sum_{T\in\Gamma_{\mathrm{I}}\cup\Gamma_{\mathrm{II}}} \sum_{E\in\mathcal{X}_E} P(E|T) \\ &\Big(y(E)\ln\Big[\hat{y}_{\mathrm{SL}}^{\mathrm{opt}}(E)\Big] + [1-y(E)]\ln\Big[1-\hat{y}_{\mathrm{SL}}^{\mathrm{opt}}(E)\Big]\Big),\end{aligned} \tag{3.55}$$

using Equation (3.50).

Learning by confusion.—For a fixed bipartition in LBC, we can proceed in a similar manner. Plugging Equation (3.50) into Equation (3.18) assuming $\bar{\mathcal{D}} = \mathcal{X}$, for any $\boldsymbol{x} \in \mathcal{X}$ we have

$$\begin{aligned}\hat{y}_{\mathrm{LBC}}^{\mathrm{opt}}(\boldsymbol{x}) &= \frac{P_{\mathrm{I}}(H(\boldsymbol{x}))}{P_{\mathrm{I}}(H(\boldsymbol{x})) + P_{\mathrm{II}}(H(\boldsymbol{x}))} \\ &= \hat{y}_{\mathrm{LBC}}^{\mathrm{opt}}(H(\boldsymbol{x})).\end{aligned} \tag{3.56}$$

Using Equation (3.20), this yields

$$\begin{aligned}I_{\mathrm{LBC}}^{\mathrm{opt}} &= 1 - \frac{1}{|\Gamma|}\sum_{T\in\Gamma}\sum_{\boldsymbol{x}\in\mathcal{X}} P(\boldsymbol{x}|T)\min\{\hat{y}_{\mathrm{LBC}}^{\mathrm{opt}}(\boldsymbol{x}), 1-\hat{y}_{\mathrm{LBC}}^{\mathrm{opt}}(\boldsymbol{x})\} \\ &= 1 - \frac{1}{|\Gamma|}\sum_{T\in\Gamma}\sum_{E\in\mathcal{X}_E} P(E|T)\min\{\hat{y}_{\mathrm{LBC}}^{\mathrm{opt}}(E), 1-\hat{y}_{\mathrm{LBC}}^{\mathrm{opt}}(E)\}.\end{aligned} \tag{3.57}$$

To obtain an expression for the optimal loss, we follow the above procedure outlined for SL starting with Equation (2.10) and eventually arrive at

$$\begin{aligned}\mathcal{L}_{\mathrm{LBC}}^{\mathrm{opt}} = &-\frac{1}{|\Gamma|}\sum_{T\in\Gamma}\sum_{E\in\mathcal{X}_E} P(E|T) \\ &\Big(y(E)\ln\Big[\hat{y}_{\mathrm{LBC}}^{\mathrm{opt}}(E)\Big] + [1-y(E)]\ln\Big[1-\hat{y}_{\mathrm{LBC}}^{\mathrm{opt}}(E)\Big]\Big).\end{aligned} \tag{3.58}$$

Prediction-based method.—Plugging Equation (3.50) into Equation (3.27) assuming $\bar{\mathcal{D}} = \mathcal{X}$, for any $\boldsymbol{x} \in \mathcal{X}$ we find that

$$\begin{aligned}\hat{y}_{\mathrm{PBM}}^{\mathrm{opt}}(\boldsymbol{x}) &= \frac{\sum_{T\in\Gamma} P\left(H(\boldsymbol{x})|T\right)T}{\sum_{T\in\Gamma} P\left(H(\boldsymbol{x})|T\right)} \\ &= \hat{y}_{\mathrm{PBM}}^{\mathrm{opt}}\left(H(\boldsymbol{x})\right).\end{aligned} \tag{3.59}$$

Using Equation (3.28), we have

$$\begin{aligned}\hat{y}_{\mathrm{PBM}}^{\mathrm{opt}}(T) &= \sum_{\boldsymbol{x}\in\mathcal{X}} P(\boldsymbol{x}|T)\hat{y}_{\mathrm{PBM}}^{\mathrm{opt}}(\boldsymbol{x}) \\ &= \sum_{E\in\mathcal{X}_E} P(E|T)\hat{y}_{\mathrm{PBM}}^{\mathrm{opt}}(E).\end{aligned} \tag{3.60}$$

To obtain an expression for the optimal loss, we rewrite Equation (2.12) as

$$\mathcal{L}_{\mathrm{PBM}} = \frac{1}{|\Gamma|} \sum_{T \in \Gamma} \sum_{\boldsymbol{x} \in \mathcal{X}} P(\boldsymbol{x}|T) \left[\hat{y}_{\mathrm{PBM}}(\boldsymbol{x}) - y(\boldsymbol{x})\right]^2 . \tag{3.61}$$

Using Equation (3.59), we have

$$\mathcal{L}^{\mathrm{opt}}_{\mathrm{PBM}} = \frac{1}{|\Gamma|} \sum_{T \in \Gamma} \sum_{\boldsymbol{x} \in \mathcal{X}} P(\boldsymbol{x}|T) \left[\hat{y}^{\mathrm{opt}}_{\mathrm{PBM}}\left(H(\boldsymbol{x})\right) - y(H(\boldsymbol{x}))\right]^2 , \tag{3.62}$$

where $y(\boldsymbol{x}) = y(H(\boldsymbol{x}))$. With Equation (3.50) we finally get

$$\mathcal{L}^{\mathrm{opt}}_{\mathrm{PBM}} = \frac{1}{|\Gamma|} \sum_{T \in \Gamma} \sum_{E \in \mathcal{X}_E} P(E|T) \left[\hat{y}^{\mathrm{opt}}_{\mathrm{PBM}}\left(E\right) - y(E)\right]^2 . \tag{3.63}$$

We have explicitly shown that the Bayes-optimal predictions, indicators, and loss values of SL, LBC, and PBM remain identical when configuration samples that follow a Boltzmann distribution are used as input, or when the corresponding energies are used as input instead, confirming the numerical observations of [Wetzel and Scherzer, 2017] and [Suchsland and Wessel, 2018]. In practice, given a finite set of samples, the inferred probability distribution $\tilde{P}(\boldsymbol{x}|T) = \mathcal{D}_T(\boldsymbol{x})/|\mathcal{D}_T|$ is only approximately Boltzmann, i.e., $\bar{\mathcal{T}}, \bar{\mathcal{D}} \approx \mathcal{X}$, and the two scenarios are only equivalent up to deviations due to finite-sample statistics. In particular, the distribution over energies obtained from the empirical distribution $\tilde{P}(\boldsymbol{x}|T) = \mathcal{D}_T(\boldsymbol{x})/|\mathcal{D}_T|$ based on raw configuration samples with the conversion being done using the estimated degeneracy factor

$$g(E) = \sum_{\boldsymbol{x} \in \bar{\mathcal{D}}} \delta_{H(\boldsymbol{x}),E}, \tag{3.64}$$

may not coincide with the empirical distribution $\tilde{P}(E|T) = \mathcal{D}_T(E)/|\mathcal{D}_T|$ over the corresponding energy. However, using the energy as input instead of configuration samples yields a more accurate estimate of the ground-truth distribution. This is because the associated state space $\mathcal{X}_E$ is significantly smaller compared to the entire configuration space $\mathcal{X}$, resulting in better statistics given a fixed number of samples. In the 2D Ising model, for example, the size of the configuration space is 2^{L^2}, whereas there are $L^2 - 1$ unique energies (for even L). Therefore, the optimal predictions and indicators obtained using the energy as input converge significantly faster compared to the case where raw spin configurations are used. Similarly, one could take advantage of the symmetries of the system by adopting a symmetry-adapted representation. Oftentimes, the energy is readily available in numerical studies. However, in principle, one can obtain accurate results without having access to the energy given that a sufficient number of raw configurations are sampled.

3.5.2 Quantum systems

In the quantum case, we will typically be looking at a state associated with a Hamiltonian $H(\gamma)$ that depends on the tuning parameter γ. This state could, for example, be the ground state or a state that has undergone unitary time evolution starting from a fixed initial state. Having chosen a complete orthonormal basis $\{|j\rangle\}_{j=1}^{\dim(\mathcal{H})}$ to study the system, the relevant quantum state at γ can be written as $|\Psi(\gamma)\rangle = \sum_{j=1}^{\dim(\mathcal{H})} c_j(\gamma)|j\rangle$. Thus, the probability distribution associated with a given value γ of the tuning parameter is $P(j|\gamma) = |c_j(\gamma)|^2$ with $1 \leq j \leq \dim(\mathcal{H})$. The value of $P(j|\gamma)$ corresponds to the probability of measuring the system in state $|j\rangle$ given that the value of the tuning parameter is γ. This corresponds to using the indices of the basis states $|j\rangle$ $[1 \leq j \leq \dim(\mathcal{H})]$ as inputs, which are governed by the probability distributions $\{P(\cdot|\gamma)\}_{\gamma\in\Gamma}$. For simplicity, we choose $|\bar{\mathcal{D}}| = \dim(\mathcal{H})$. In the case of spin systems, we use the S^z basis. Here, S^x, S^y, and S^z denote the spin-operators. For bosonic and fermionic systems, we choose the Fock basis. This choice of bases corresponds to experimentally accessible local measurements [Simon *et al.*, 2011; Bernien *et al.*, 2017; Lukin *et al.*, 2019; Rispoli *et al.*, 2019; Jepsen *et al.*, 2020, 2021; Ebadi *et al.*, 2021]. In this chapter, to perform exact diagonalization and solve the Schrödinger equation, we use the QuSpin package [Weinberg and Bukov, 2017, 2019] in `Python`. Thus, we have direct access to the underlying probability distributions and do not rely on sampling. In Section 3.6.5, we show that the optimal indicators can also be obtained from individual samples, i.e., measurement outcomes (similar to the classical case). As such, the procedure is *in principle* applicable to experimental scenarios.

In general, we can consider scenarios where a state $|\Psi_i\rangle$ is drawn with probability $a(i|\gamma)$ at parameter value $\gamma \in \Gamma$. Then, the relevant quantum state is given by a classical probabilistic mixture $\rho(\gamma) = \sum_i a(i|\gamma)|\Psi_i\rangle\langle\Psi_i|,\ i \in \mathbb{N}$. The probability distribution associated with such a state is $P(j|\gamma) = \sum_i a(i|\gamma)|c_{ij}|^2$, where $|\Psi_i\rangle = \sum_{j=1}^{\dim(\mathcal{H})} c_{ij}|j\rangle$. This case will be particularly relevant for the study of many-body localization phase transitions where disorder is naturally present (see Section 3.6.6).

Clearly, in the quantum case there is an ambiguity in the choice of input, or equivalently, the choice of measurement basis. Changing the measurement basis may change the probability distributions underlying the data, and thus the corresponding optimal predictors and indicators. In turn, the estimated critical value of the tuning parameter may change (in a way that is difficult to assess *a priori*). To avoid an explicit choice of measurement basis, sampling over various classical projections can be performed. In general, measurements given by informationally complete positive operator-valued measures (IC-POVMs) may be used [Nielsen and Chuang, 2010; Carrasquilla *et al.*, 2019].[7] Classical representations of quantum states obtained via classical shadow tomography [Huang *et al.*, 2020, 2022b,a] can be considered an example of this. However, projective measurements in a single basis have been the most common choice in the literature on detecting phase transitions from data, reflecting experimental constraints or prior knowledge of the system [Torlai *et al.*, 2019; Greplova *et al.*, 2020; Miles *et al.*, 2021; Bohrdt *et al.*, 2021; Maskara *et al.*, 2022; Miles *et al.*, 2023]. Hence, for this chapter, we will rely on projective measurement in a single basis.

3.6 Results

Let us finally discuss the optimal predictions and indicators of phase transitions of SL, LBC, and PBM, obtained for the Ising model, Ising gauge theory, XY model, XXZ model, Kitaev model, and Bose-Hubbard model. Whenever exact expressions for the underlying distributions are available, such as for the XXZ model or the Kitaev

[7] We will explore measurements described by IC-POVMs in Chapter 4 and discuss the influence of the choice of measurement further in Chapter 6.

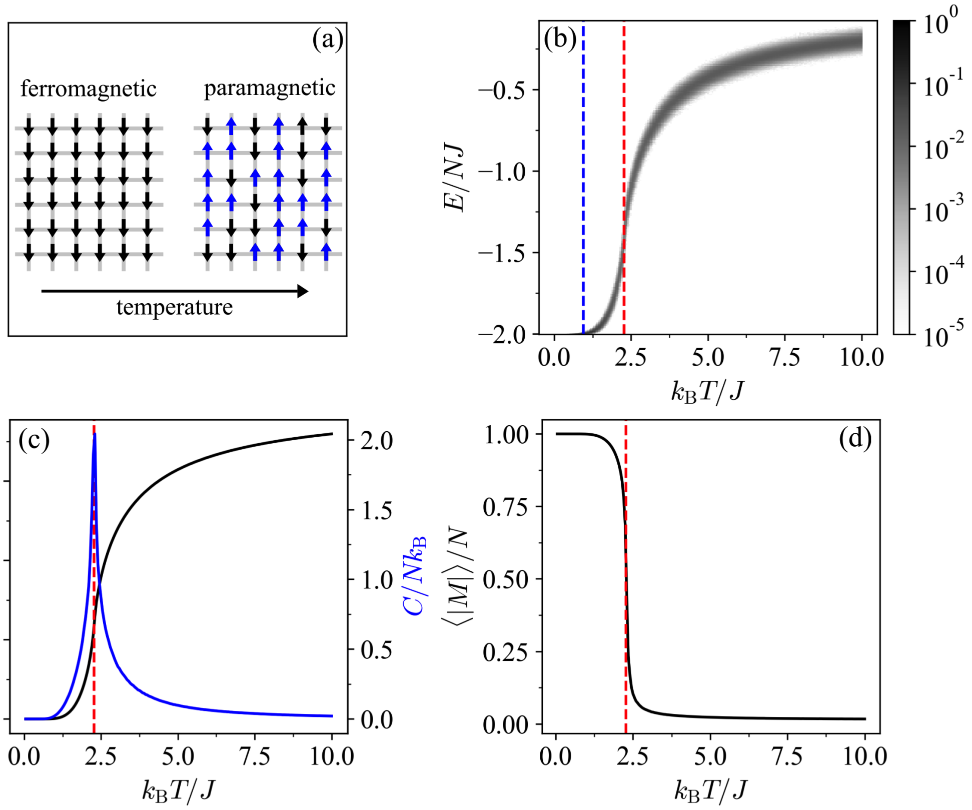

FIGURE 3.4: (a) Illustration of the symmetry-breaking phase transition in the Ising model. (b) Probability distributions that govern the input data (here the energy) as a function of the tuning parameter, where $N = L^2$. The color scale denotes the probability. The blue dashed line highlights the predicted critical temperature based on the optimal indicators of SL and PBM. (c) Average energy per site (black) and associated heat capacity (blue) as a function of temperature. (d) Average magnetization per site as a function of temperature. Here, we consider an Ising model with $L = 60$ and the dimensionless temperature as a tuning parameter $\gamma = k_BT/J$, where $\gamma_1 = 0.05$, $\gamma_K = 10$, and $\Delta\gamma = 0.05$. The critical temperature [Equation (2.3)] is highlighted by a red dashed line.

model,[8] we will utilize them to compute Bayes-optimal predictions and indicators. In all other cases, we utilize a large number of samples, resulting in empirically optimal predictions and indicators that are expected to closely match the Bayes-optimal ones.[9] For SL, we will typically choose $\Gamma_{\rm I} = \{\gamma_1\}$ and $\Gamma_{\rm II} = \{\gamma_K\}$ as training regions, i.e., the two points deepest within the two phases.[10] We make this generic choice given that it requires minimal information about the location of the underlying critical point.

3.6.1 Ising model

For a description of the Ising model and the data generation process, see Section 2.3.1 and Figure 3.4. The results for the Ising model are shown in Figure 3.5. Interestingly,

[8] In the case of the many-body localization transition within the Bose-Hubbard model, we randomly sample a large number of disorder realizations.

[9] In the case of the classical equilibrium models, working in energy space rather than configuration space is key for obtaining well-converged empirical distributions.

[10] See Appendix C for a discussion on how the choice of training regions influences the results.

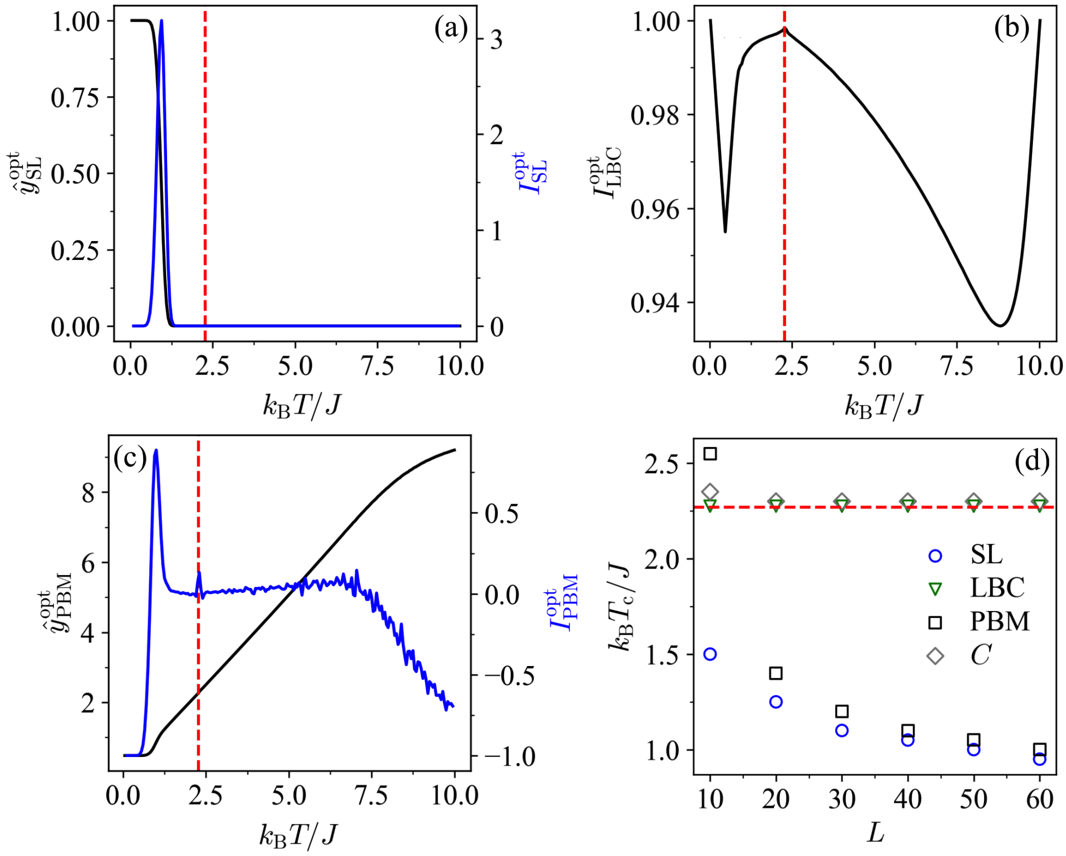

FIGURE 3.5: ML results for the Ising model ($L = 60$) with the dimensionless temperature as a tuning parameter $\gamma = k_{\mathrm{B}}T/J$, where $\gamma_1 = 0.05$, $\gamma_K = 10$, and $\Delta\gamma = 0.05$. (a) Mean optimal prediction $\hat{y}^{\mathrm{opt}}_{\mathrm{SL}}$ in SL (black) and the corresponding indicator $I^{\mathrm{opt}}_{\mathrm{SL}}$ (blue). In SL, the data obtained at γ_1 and γ_K constitutes our training set, i.e., $r_{\mathrm{I}} = 1$ and $l_{\mathrm{II}} = K$. (b) Optimal indicator of LBC, $I^{\mathrm{opt}}_{\mathrm{LBC}}$ (black). (c) Mean optimal prediction $\hat{y}^{\mathrm{opt}}_{\mathrm{PBM}}$ in PBM (black) and the corresponding indicator $I^{\mathrm{opt}}_{\mathrm{PBM}}$ (blue). (d) Estimated critical temperatures based on $I^{\mathrm{opt}}_{\mathrm{SL}}$ (SL), $I^{\mathrm{opt}}_{\mathrm{LBC}}$ (LBC), $I^{\mathrm{opt}}_{\mathrm{PBM}}$ (PBM), and heat capacity (C) as a function of the lattice size L. The estimated critical temperature based on the heat capacity corresponds to the location of its maximum. The critical temperature [Equation (2.3)] is highlighted by a red dashed line.

optimal SL fails to predict the correct critical temperature even for large lattices [see Figures 3.5(a) and (d)]. In fact, we can further analyze the special case when the inputs are governed by Boltzmann distributions [Equation (3.46)]: For training data obtained at $T_1 = 0$ (region I) and $T_K > 0$ (region II), the mean Bayes-optimal prediction of SL at an intermediate temperature T is

$$\hat{y}^{\mathrm{opt}}_{\mathrm{SL}}(T) = \frac{P(E_{\mathrm{gs}}|T)}{1 + P(E_{\mathrm{gs}}|T_K)} \propto P(E_{\mathrm{gs}}|T), \tag{3.65}$$

which approaches $P(E_{\mathrm{gs}}|T)$ in the thermodynamic limit as $T_K \to \infty$. Here, E_{gs} denotes the ground-state energy.

Proof

We take region I to be composed of a single point T_1. Let $T_1 \to 0$ such that

$$P(E|T_1) = \begin{cases} 1 \text{ if } E = E_{\text{gs}}, \\ 0 \text{ otherwise.} \end{cases} \tag{3.66}$$

Plugging into Equation (3.1) yields

$$\hat{y}_{\text{SL}}^{\text{opt}}(E) = \begin{cases} \frac{1}{1+P_{\text{II}}(E_{\text{gs}})} \text{ if } E = E_{\text{gs}}, \\ 0 \text{ otherwise.} \end{cases} \tag{3.67}$$

We calculate the mean Bayes-optimal prediction at a given temperature as

$$\hat{y}_{\text{SL}}^{\text{opt}}(T) = \sum_{E \in \mathcal{X}_E} P(E|T)\hat{y}_{\text{SL}}^{\text{opt}}(E). \tag{3.68}$$

Using Equation (3.67), this results in

$$\hat{y}_{\text{SL}}^{\text{opt}}(T) = \frac{P(E_{\text{gs}}|T)}{1 + P_{\text{II}}(E_{\text{gs}})}. \tag{3.69}$$

Assuming region II is composed of a single point T_K, we have $P_{\text{II}}(E_{\text{gs}}) = P(E_{\text{gs}}|T_K)$ and recover Equation (3.65). For $T_K \to \infty$, we have $P(E_{\text{gs}}|T_K) = g(E_{\text{gs}})/|\mathcal{X}|$. For the two-dimensional Ising model, for example, $|\mathcal{X}| = 2^{L\times L}$. Approaching the thermodynamic limit, this yields $\hat{y}_{\text{SL}}^{\text{opt}}(T) \to P(E_{\text{gs}}|T)$.

Therefore, in this case, the optimal indicator in SL peaks at the temperature at which the probability of drawing the ground state changes most [see the blue dashed line in Figure 3.4(b)]. The location of the peak tends to zero as one approaches the thermodynamic limit, see Figure 3.5(d).

The optimal indicator of PBM shows two distinct peaks: One coincides with the peak of the optimal indicator in SL, whereas the other coincides with the critical temperature of the Ising model [see Figure 3.5(c)]. A similar indicator signal (with two distinct peaks) was observed in [Schäfer and Lörch, 2019] with NNs after a sufficiently large number of training epochs. In principle, the finite-size scaling analysis allows one to identify the dominant peak as erroneous without prior knowledge of T_c, because it shifts toward $T = 0$ as the lattice size is increased, whereas the small peak remains stable. In the same fashion, the output of SL may be identified to be erroneous.

In [Carrasquilla and Melko, 2017], SL with NNs was able to predict the critical temperature of the Ising model for various lattice sizes correctly. In this case, small NNs with restricted expressive power in combination with ℓ_2 regularization were used. Similarly, using PBM in [Schäfer and Lörch, 2019] a single, distinct peak around T_c was observed after a small number of training epochs with a second peak emerging after longer training. Training time, NN size, and explicit ℓ_2 regularization are all factors that influence the effective capacity of the resulting model and thus determine its ability to approximate the optimal predictive model [Goodfellow *et al.*, 2016; Hu *et al.*, 2021], i.e., to realize the global minimum of the loss function corresponding to the optimal predictions and indicators. We recover the same behavior using NNs as [Carrasquilla and Melko, 2017; Schäfer and Lörch, 2019] by restricting the model capacity, e.g., by choosing a small NN, stopping the training early, or using strong ℓ_2 regularization, see Section 3.7.4 and recall Figures 2.6(a) and (c) in Chapter 2. As these restrictions are lifted, i.e., by choosing a larger NN, training for longer, or reducing the regularization strength, the NN-based predictions and indicators approach the corresponding optimal predictions and indicators displayed in Figure 3.5. Thus, our analysis demonstrates that SL and PBM necessarily rely on models with restricted capacity and hyperparameter tuning to correctly predict the critical temperature of the Ising model.

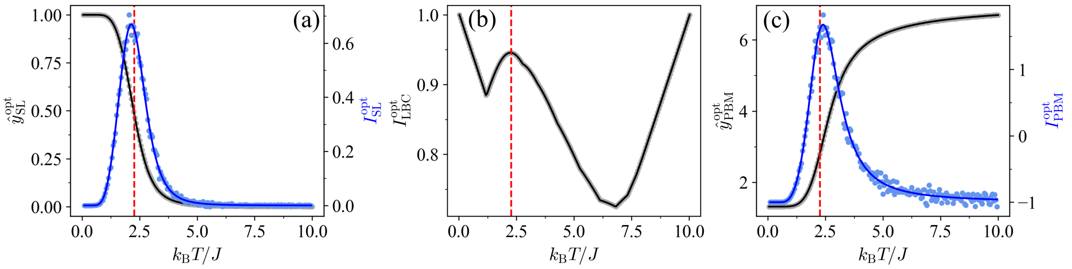

FIGURE 3.6: ML results for the Ising model ($L = 4$) with the dimensionless temperature as a tuning parameter $\gamma = k_{\mathrm{B}}T/J$, where $\gamma_1 = 0.05$, $\gamma_K = 10$, and $\Delta\gamma = 0.05$. The critical temperature [Equation (2.3)] is highlighted by a red dashed line. In SL, the data obtained at γ_1 and γ_K constitutes our training set, i.e., $r_{\mathrm{I}} = 1$ and $l_{\mathrm{II}} = K$. The input energies are computed based on spin configurations obtained through exact enumeration (lines) or Monte Carlo sampling (points). (a) Mean optimal prediction $\hat{y}_{\mathrm{SL}}^{\mathrm{opt}}$ in SL (black) and the corresponding indicator $I_{\mathrm{SL}}^{\mathrm{opt}}$ (blue). (b) Optimal indicator of LBC, $I_{\mathrm{LBC}}^{\mathrm{opt}}$ (black). (c) Mean optimal prediction $\hat{y}_{\mathrm{PBM}}^{\mathrm{opt}}$ in PBM (black) and the corresponding indicator $I_{\mathrm{PBM}}^{\mathrm{opt}}$ (blue).

Finally, the optimal indicator of LBC correctly highlights the critical temperature of the Ising model for various lattice sizes matching the results of Van Nieuwenburg *et al.* [2017], see Figures 3.5(b) and (d). Overall, the optimal indicators of all three methods show peaks at temperatures where the probability distribution underlying the data varies strongly. Recall the finding from Section 3.2 that all three methods gauge changes in the probability distributions underlying the data. We have confirmed that the results shown in Figure 3.5 are stable against small perturbations of the chosen parameter range, including regions I and II in SL, see Appendix C.

Influence of finite-sample statistics

Recall that we use the energy from Monte Carlo sampling as input, where 10^5 spin configurations are drawn per temperature. In Figure 3.6, we compare the Bayes-optimal predictions and indicators for the Ising model on a 4×4 lattice when enumerating all $2^{16} = 65536$ spin configurations explicitly or using Monte Carlo sampling with 10^5 number of configurations per sampled value of the tuning parameter, resulting in empirically optimal predictions and indicators. The results obtained based on the two distinct data sets are in good agreement. This is to be expected given that there are only 15 unique energies. The noise present in the indicator signals of SL and PBM when using Monte Carlo samples is absent when using exact enumeration. In the latter case, both indicators vary smoothly as a function of temperature. As such, this noise can be attributed to finite-sample statistics. Similarly, the fluctuations present in the optimal indicator signal of PBM in Figure 3.5 can be attributed to finite-sample statistics. Here, the analytical expression for the optimal indicator signal allows us to disentangle the stochasticity inherent to the NN training from other sources of noise, which was not rigorously possible in previous works.

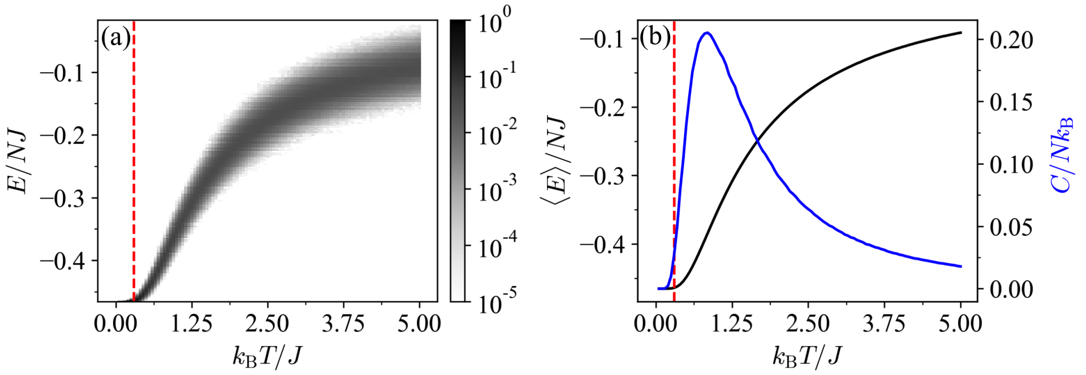

FIGURE 3.7: Results for the IGT ($L = 28$) with the dimensionless temperature as a tuning parameter $\gamma = k_BT/J$. The crossover temperature is highlighted by a red dashed line and scales as $k_BT_c/J \propto 1/\ln(2L^2)$ [Castelnovo and Chamon, 2007]. (a) Probability distributions governing the input data (here the energy) as a function of the tuning parameter, where $N = 2L^2$. The color scale depicts the probability. (b) Average energy per site (black) and associated heat capacity (blue) as a function of temperature. Note that the heat capacity does not peak at the crossover temperature.

In general, for both the classical and quantum systems we observe that the overlap in the underlying probability distributions leading to a peak in the indicator signals decreases as the number of samples is decreased. However, meaningful results can already be obtained when only a fraction of the total state space is covered. In the case of the Ising model on a 60 × 60 lattice, for example, we observe that the optimal predictions and indicators are already well converged for $|\mathcal{D}_\gamma| = 10^2$, i.e., matching the results obtained with $|\mathcal{D}_\gamma| = 10^5$. In particular, the key features in the indicators, i.e., the peak locations, can already be identified for $|\mathcal{D}_\gamma| = 10$. Compare this to the unique number of energies given by $|\mathcal{X}_E| = 3599$.

3.6.2 Ising gauge theory

For a description of the model and data generation process, see Section 2.3.2 and Figure 3.7. Recall that SL, LBC, and PBM are *a priori* sensitive to both phase transitions and crossovers. The results for the crossover in the IGT are shown in Figure 3.8. The optimal indicator of SL [Figure 3.8(a)] shows an appropriate scaling behavior. Moreover, the corresponding estimated critical temperature highlights the first appearance of violated local constraints, see Figure 3.7. This can be confirmed explicitly as SL can be shown to measure changes in the probability of drawing the ground state (cf. Section 3.6.1). Observe that the underlying probability distribution undergoes a large change at the crossover temperature, see Figure 3.7(a). SL and PBM were found to correctly highlight the crossover temperature of the IGT using NNs in [Carrasquilla and Melko, 2017] and [Greplova *et al.*, 2020], respectively. In fact, the optimal model underlying PBM for the IGT coincides with the physically motivated density-of-states-based model proposed in [Greplova *et al.*, 2020]: Greplova *et al.* empirically found that the NN-based predictions of PBM agree well with a physical model based on the underlying density of states, which was proposed in an *ad hoc* fashion guided by physical intuition. Here, we have explicitly confirmed this physical intuition on what the NN learns by proving that the optimal prediction of PBM for a given configuration in the IGT corresponds to the most likely tuning parameter value based on the underlying Boltzmann distribution.

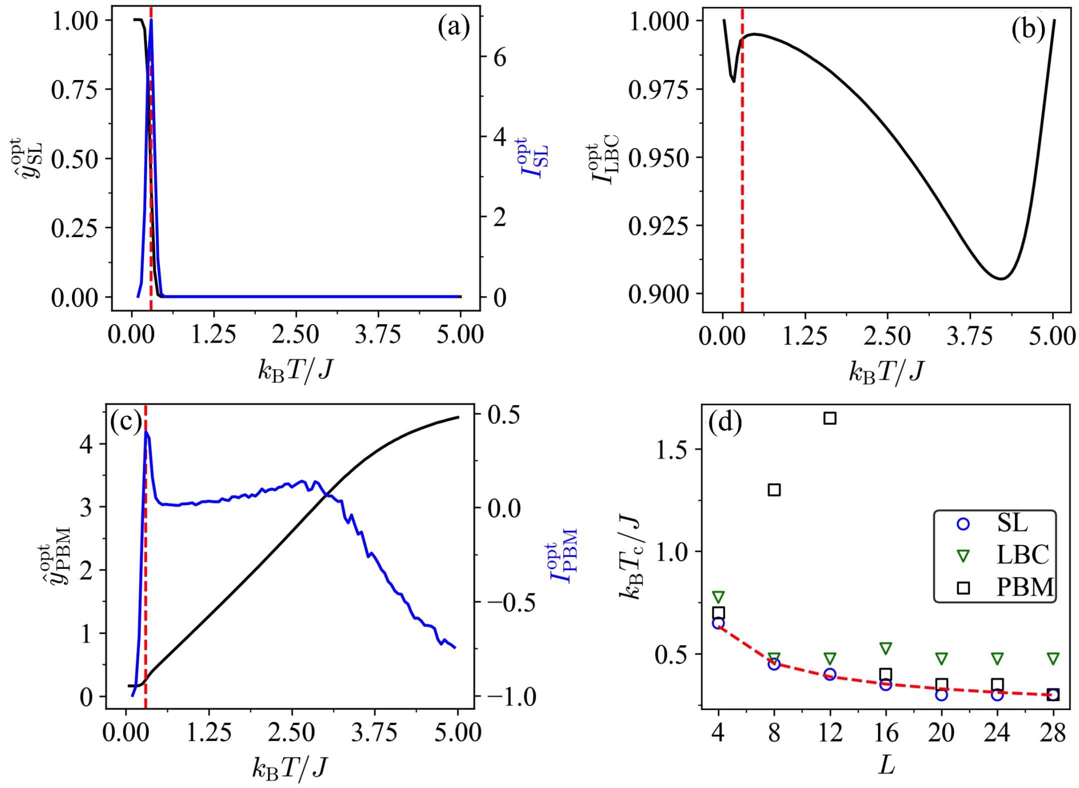

FIGURE 3.8: ML results for the IGT ($L = 28$) with the dimensionless temperature as a tuning parameter $\gamma = k_{\mathrm{B}}T/J$, where $\gamma_1 = 0.05$, $\gamma_K = 5$, and $\Delta\gamma = 0.05$. The crossover temperature is highlighted by a red dashed line and scales as $k_{\mathrm{B}}T_c/J \propto 1/\ln(2L^2)$ [Castelnovo and Chamon, 2007]. (a) Mean optimal prediction $\hat{y}^{\mathrm{opt}}_{\mathrm{SL}}$ in SL (black) and the corresponding indicator $I^{\mathrm{opt}}_{\mathrm{SL}}$ (blue). In SL, the data obtained at γ_1 and γ_K constitutes our training set, i.e., $r_{\mathrm{I}} = 1$ and $l_{\mathrm{II}} = K$. (b) Optimal indicator of LBC, $I^{\mathrm{opt}}_{\mathrm{LBC}}$ (black). (c) Mean optimal prediction $\hat{y}^{\mathrm{opt}}_{\mathrm{PBM}}$ in PBM (black) and the corresponding indicator $I^{\mathrm{opt}}_{\mathrm{PBM}}$ (blue). (d) Estimated critical temperature based on $I^{\mathrm{opt}}_{\mathrm{SL}}$ (SL), $I^{\mathrm{opt}}_{\mathrm{LBC}}$ (LBC), $I^{\mathrm{opt}}_{\mathrm{PBM}}$ (PBM) as a function of the lattice size L.

We find that the optimal indicator of PBM correctly marks the crossover temperature of the IGT except at small lattice sizes. As for the Ising model, the optimal indicator of PBM exhibits two peaks in this case. The peak located at the crossover temperature dominates for large lattice sizes. Note that for the IGT it is not beneficial to reduce the model capacity when using PBM or SL given that the corresponding optimal indicators correctly highlight the crossover temperature. In fact, this leads to a peak closely matching the heat capacity. The heat capacity, however, fails to identify the crossover, see Figure 3.7(b).

The optimal indicator of LBC correctly highlights the crossover temperature via its local maximum at small lattice sizes, but shows slight deviations from the appropriate scaling behavior for large lattices. In [Greplova *et al.*, 2020] difficulties were observed to identify the crossover temperature using LBC due to a distorted W-shape of its indicator. Choosing the same range for the tuning parameter, we can qualitatively reproduce their results using our analytical expression for the optimal indicator of LBC, see Figure 3.9.

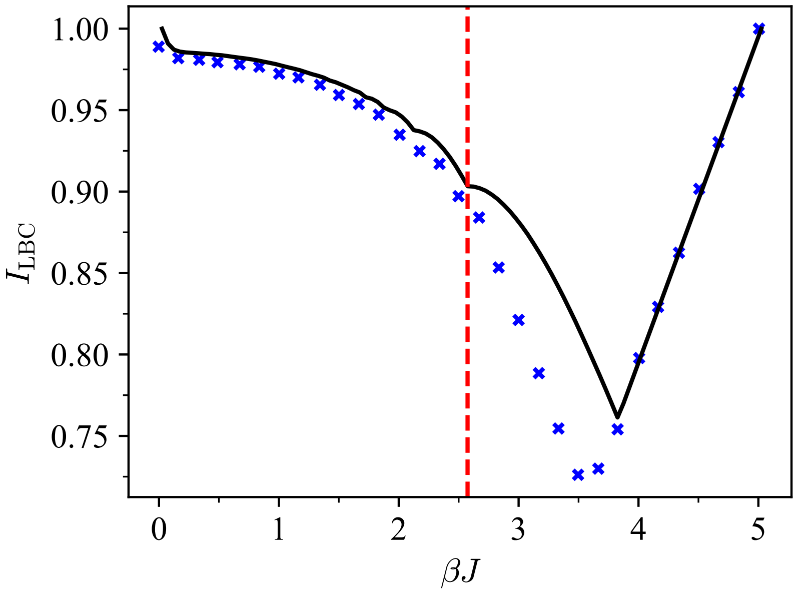

FIGURE 3.9: Indicator of LBC for the IGT ($L = 12$) with dimensionless inverse temperature $\gamma = \beta J$ as a tuning parameter, where $\gamma_1 = 0.05$, $\gamma_K = 5$, and $\Delta\gamma = 0.05$. The crossover temperature is highlighted by a red dashed line. The optimal indicator $I_{\text{LBC}}^{\text{opt}}$ we compute is shown in black. The blue crosses mark the NN-based indicator $I_{\text{LBC}}^{\text{NN}}$ obtained in Figure C1 of [Greplova *et al.*, 2020] using raw spin configurations as input. As expected, the NN-based indicator lies on or below the optimal indicator curve.

Using NNs, it is difficult to make concrete statements on whether a method succeeds or fails at identifying a given phase transition due to the inherent stochasticity arising during NN training and the choice of hyperparameters, such as the NN size. Our theoretical analysis allows for rigorous statements to be made about the optimal outcome when applying ML methods for detecting phase transitions to a given system (i.e., data set). In this particular example, the analytical expressions allow us to determine that when training highly expressive NNs for sufficiently long, the indicator signal of LBC is indeed ambiguous (as reported in [Greplova *et al.*, 2020]). Restricting the model capacity is not found to resolve this issue.

3.6.3 XY model

Next, we consider the two-dimensional classical XY model that exhibits a Berezinskii–Kosterlitz–Thouless (BKT) transition driven by the emergence of topological defects [Kosterlitz and Thouless, 1973; Kosterlitz, 1974]. The model is described by the following Hamiltonian

$$H = -J \sum_{\langle ij \rangle} \cos(\theta_i - \theta_j), \tag{3.70}$$

where $\langle ij \rangle$ denotes the sum over nearest neighbors (with periodic boundary conditions) of a square lattice of linear size L. The angle $\theta_i \in [0, 2\pi)$ corresponds to the orientation of the spin at site i. Once again, we use the Metropolis-Hastings algorithm [Metropolis *et al.*, 1953] to sample spin configurations from the thermal distribution at a given temperature T. The lattice is initialized in a random spin configuration. The lattice is updated by drawing a random spin to which we add a perturbation $\Delta\theta \in [-\pi, \pi]$ drawn uniformly at random. This perturbation is accepted

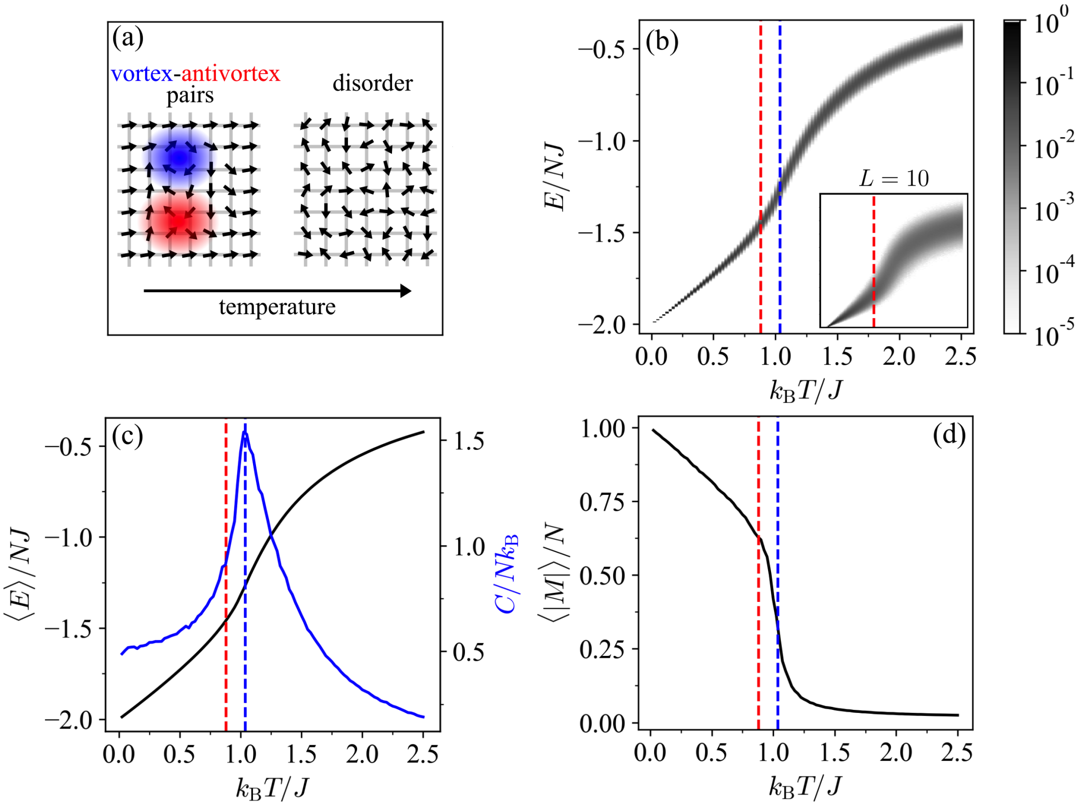

FIGURE 3.10: Results for the XY model ($L = 60$) with the dimensionless temperature as a tuning parameter $\gamma = k_{\mathrm{B}}T/J$. The BKT transition temperature $k_{\mathrm{B}}T_{\mathrm{c}}/J \approx 0.8935$ [Hsieh *et al.*, 2013] is highlighted by a red dashed line. The blue dashed line highlights the estimated critical temperature using LBC. (a) Illustration of the BKT phase transition in the XY model. (b) Probability distributions governing the input data (here the energy) as a function of the tuning parameter, where $N = L^2$. The color scale denotes the probability. The inset shows the probability distributions for $L = 10$. (c) Average energy per site (black) and associated heat capacity (blue) as a function of temperature. (d) Average magnetization per site as a function of temperature.

with probability $\min\{1, e^{-\Delta E/k_{\mathrm{B}}T}\}$, where ΔE is the energy difference resulting from the perturbation. To ensure that the systems are sufficiently thermalized, we sweep the complete lattice 10^5 times, where each lattice site is updated once per sweep. After the thermalization period, we collect 10^5 samples, which we find to be sufficient for achieving convergence. We start at a high temperature and decrease it gradually.

The formation of topological defects (i.e., vortices and antivortices) results in a quasi-long-range-ordered phase. The transition between the quasi-long-range-ordered phase at low temperature and a disordered phase at high temperature is a BKT transition, and the associated critical temperature is $k_{\mathrm{B}}T_{\mathrm{c}}/J \approx 0.8935$ [Hsieh *et al.*, 2013]. Below T_{c}, vortex-antivortex pairs form due to thermal fluctuations, but they remain bound to minimize their total free energy [see Figure 3.10(a)]. At T_{c}, the entropic contribution to the free energy equals the binding energy of a pair, which triggers vortex unbinding. These unbinding events drive the BKT phase transition. Note that the heat capacity has a peak at $T > T_{\mathrm{c}}$ which is associated with the entropy released when most vortex pairs unbind [Chaikin and Lubensky, 1995; Van Himbergen and Chakravarty, 1981], see Figure 3.10(c). Moreover, while the XY model has strictly

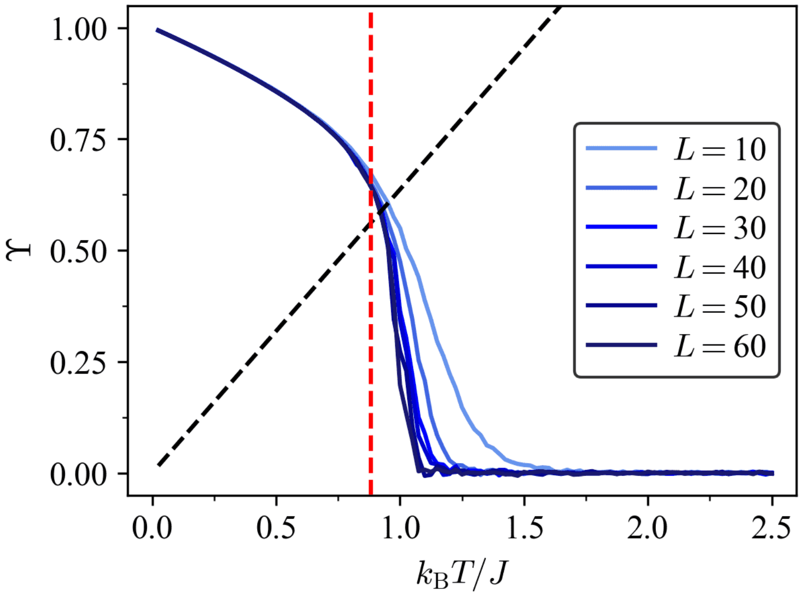

FIGURE 3.11: Helicity modulus Υ as a function of the tuning parameter $\gamma = k_\mathrm{B}T/J$ for the two-dimensional XY model for various lattice sizes. The value of the BKT transition point from literature $k_\mathrm{B}T_\mathrm{c}/J \approx 0.8935$ [Hsieh *et al.*, 2013] is highlighted by a red dashed line. The estimated transition point based on our Monte Carlo samples at finite size corresponds to the point at which the helicity modulus crosses the line given by $\frac{2k_\mathrm{B}T}{J\pi}$ (black dashed line).

zero magnetization for all $T > 0$ in the thermodynamic limit, a non-zero value is found for systems of finite size [Chung, 1999], see Figure 3.10(d). Instead, the critical temperature can, for example, be estimated based on the helicity modulus [Van Himbergen and Chakravarty, 1981; Minnhagen and Kim, 2003]. At the critical point, the helicity modulus Υ crosses $2T/\pi$ [Van Himbergen and Chakravarty, 1981; Minnhagen and Kim, 2003]. The helicity modulus is also referred to as spin stiffness or spin rigidity and measures the response of the system to an in-plane twist of the spins. We find that the estimated BKT transition point based on our samples matches the literature value well, see Figure 3.11.

Note that in the XY model, the angle of each spin can take on any value $\theta \in [0, 2\pi]$. This results in a continuum of states. Hence, we discretize the energy in practice, which serves as an input for the ML methods. This discretization eases computation and, more crucially, results in overlapping probability distributions given finite-sample statistics (recall case 3 in Section 3.3). The discretization is performed through simple histogram-binning using 1000 bins of equal size. The number of bins was increased systematically until a convergence of the optimal indicator signals was observed.

The ML results for the XY model are shown in Figure 3.12. Here, SL fails to predict the critical temperature correctly. This failure is linked to the fact that the optimal indicator of SL highlights changes in the probability of obtaining the ground state (cf. Section 3.6.1), which quickly vanishes with increasing temperature, see Figure 3.10(b). In a similar spirit, in [Beach *et al.*, 2018] it was found that "naive" SL (without engineering the features or NN architecture) fails to yield accurate estimates of the critical temperature. Here, we explicitly confirm that a classification based on detecting vortices does not correspond to the most optimal strategy. The peak in the optimal indicator of LBC matches the peak in the heat capacity at $k_\mathrm{B}T/J \approx 1$, cf.

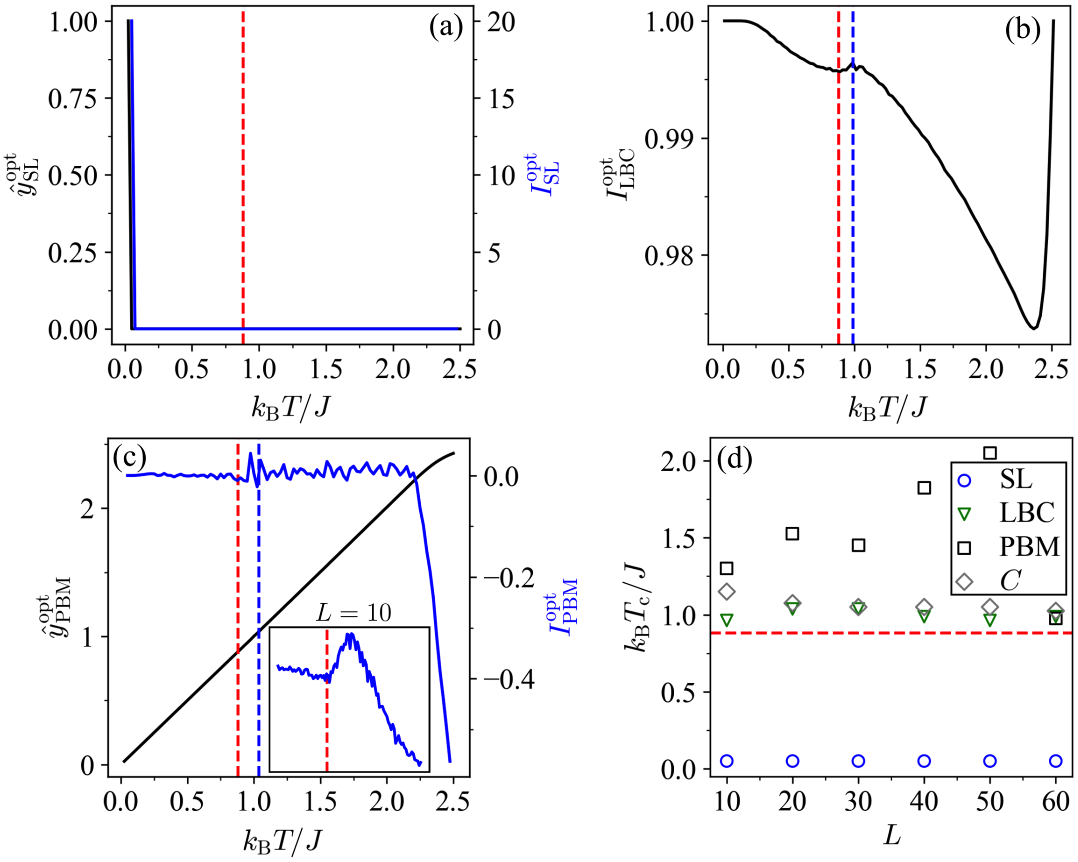

FIGURE 3.12: ML results for the XY model ($L = 60$) with the dimensionless temperature as a tuning parameter $\gamma = k_{\mathrm{B}}T/J$, where $\gamma_1 = 0.025$, $\gamma_K = 2.5$, and $\Delta\gamma = 0.025$. The BKT transition temperature $k_{\mathrm{B}}T_{\mathrm{c}}/J \approx 0.8935$ [Hsieh *et al.*, 2013] is highlighted by a red dashed line. The blue dashed line highlights the estimated critical temperature using LBC. (a) Mean optimal prediction $\hat{y}_{\mathrm{SL}}^{\mathrm{opt}}$ in SL (black) and the corresponding indicator $I_{\mathrm{SL}}^{\mathrm{opt}}$ (blue). In SL, the data obtained at γ_1 and γ_K constitutes our training set, i.e., $r_{\mathrm{I}} = 1$ and $l_{\mathrm{II}} = K$. (b) Optimal indicator of LBC, $I_{\mathrm{LBC}}^{\mathrm{opt}}$ (black). The blue dashed line highlights the predicted critical temperature of LBC. (c) Mean optimal prediction $\hat{y}_{\mathrm{PBM}}^{\mathrm{opt}}$ in PBM (black) and the corresponding indicator $I_{\mathrm{PBM}}^{\mathrm{opt}}$ (blue). The inset shows the optimal indicator signal of PBM for $L = 10$, which exhibits a peak near the location of the maximum in the heat capacity. (d) Estimated critical temperature based on $I_{\mathrm{SL}}^{\mathrm{opt}}$ (SL), $I_{\mathrm{LBC}}^{\mathrm{opt}}$ (LBC), $I_{\mathrm{PBM}}^{\mathrm{opt}}$ (PBM), and heat capacity (C) as a function of the lattice size L. The estimated critical temperature of the heat capacity corresponds to the location of its maximum.

Figures 3.10(c) and 3.12(b), and thus overestimates the critical temperature of the XY model. In [Beach *et al.*, 2018] indicator signals of similar shape were obtained using LBC with NNs for the XY model. The rapid decrease in the optimal indicator of LBC for $k_{\mathrm{B}}T/J \gtrsim 1$ can be attributed to the increase in the overlap of the underlying probability distributions [Figure 3.10(b)], which results in a higher classification error. Note that the overlap of the probability distributions decreases with increasing lattice size, see Figure 3.10(b). Hence, the indicator of PBM [Figure 3.12(c)] shows a clear peak close to the location of the peak in the heat capacity for small lattice sizes. For systems of increasing size, the optimal predictions of PBM start to closely match the underlying tuning parameter, resulting in an increasingly linear behavior [see black line in Figure 3.12(c)]. This corresponds to an optimal indicator signal close to zero, where the variations in the predicted critical value of the tuning parameter [Figure 3.12(d)] are due to small local fluctuations.

Overall, the behavior of the optimal indicators of all three methods closely resembles our previous example regarding perfectly distinguishable input data (case 3 in Section 3.3). This can be traced back to the small overlap of the underlying probability distributions, see Figure 3.10(b). The increase in the overlap with increasing temperature results in a decrease in the mean classification accuracy of LBC, i.e., its indicator [see Figure 3.12(b)]. Evidently, in the case of the XY model, NNs with restricted expressive power and other phase-classification methods based on the similarity of input data [Rodriguez-Nieva and Scheurer, 2019] may provide more valuable insights. In particular, we find that the indicators peak close to the transition temperature, i.e., near the location of the peak in the heat capacity and drop in the magnetization, when restricting the model capacity, e.g., by stopping the NN training early (see Section 3.7). Recall that this was also observed in the case of the Ising model (see Section 3.6.1).

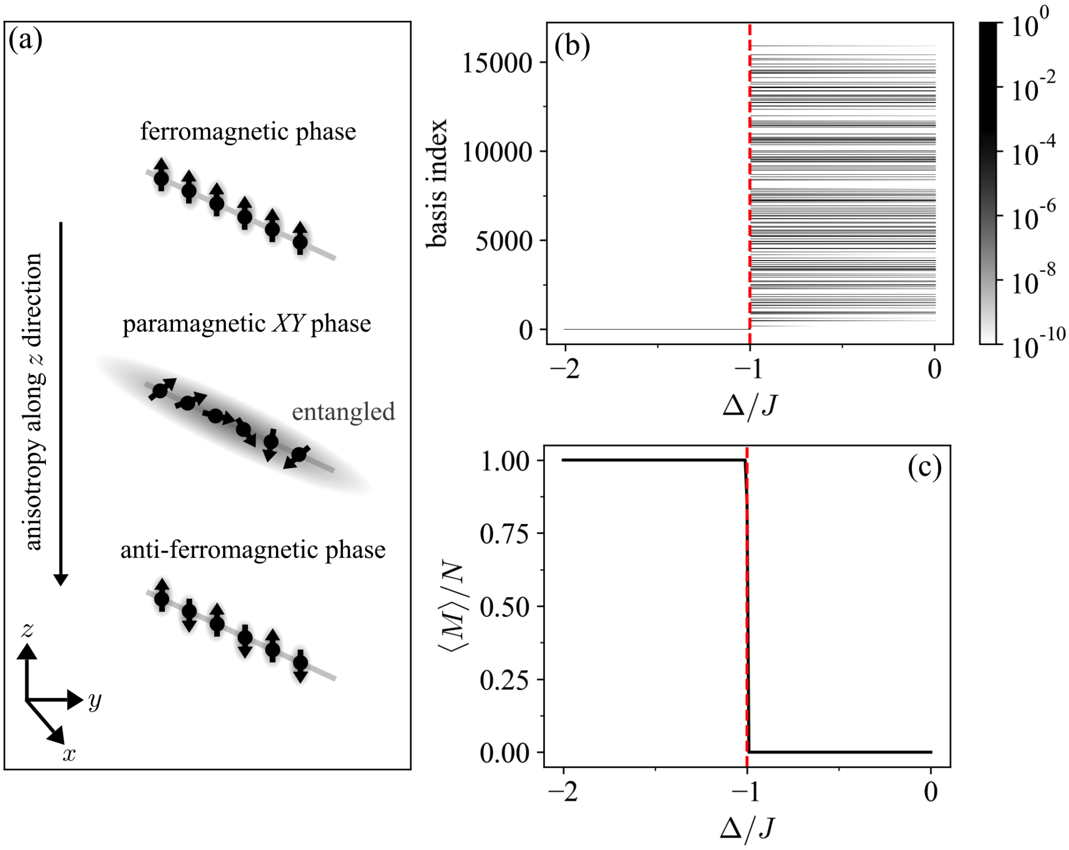

FIGURE 3.13: Results for the XXZ chain ($L = 14$) with the dimensionless anisotropy strength along the z-direction as the tuning parameter $\gamma = \Delta/J$. The critical value of the tuning parameter $\Delta/J = -1$ at which the phase transition between the ferromagnetic phase and paramagnetic XY phase occurs is highlighted by a red dashed line. (a) Illustration of the quantum phase transitions of the XXZ chain. (b) Probability distributions governing the input data (indices of S^z basis states) as a function of tuning parameter, where the color scale denotes the probability. The color scale is cut off at 10^{-10} to improve visual clarity. (c) Average magnetization per site (black), where $N = L$.

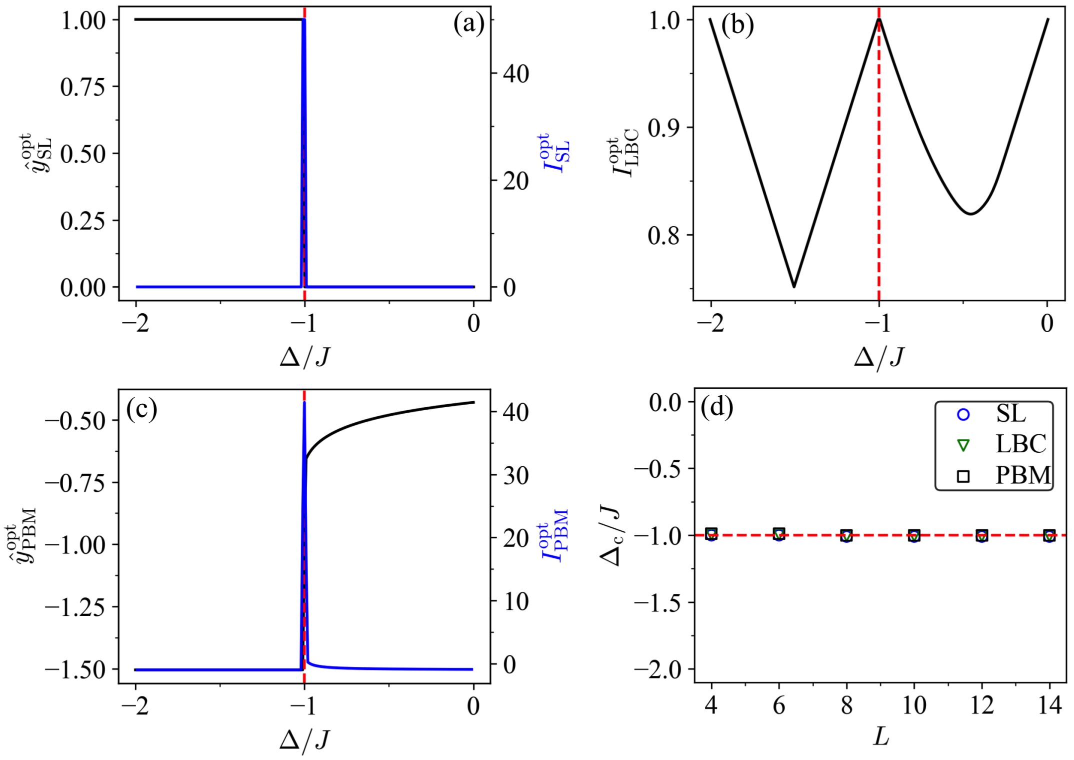

FIGURE 3.14: ML results for the XXZ chain ($L = 14$) with the dimensionless anisotropy strength along the z-direction as the tuning parameter $\gamma = \Delta/J$, where $\gamma_1 = -2$, $\gamma_K = 0$, and $\Delta\gamma = 0.01$. The critical value of the tuning parameter $\Delta/J = -1$ at which the phase transition between the ferromagnetic phase and paramagnetic XY phase occurs is highlighted by a red dashed line. (a) Mean optimal prediction $\hat{y}_{\mathrm{SL}}^{\mathrm{opt}}$ in SL (black) and the corresponding indicator $I_{\mathrm{SL}}^{\mathrm{opt}}$ (blue). In SL, the data obtained at γ_1 and γ_K constitutes our training set, i.e., $r_{\mathrm{I}} = 1$ and $l_{\mathrm{II}} = K$. (b) Optimal indicator of LBC, $I_{\mathrm{LBC}}^{\mathrm{opt}}$ (black). (c) Mean optimal prediction $\hat{y}_{\mathrm{PBM}}^{\mathrm{opt}}$ in PBM (black) and the corresponding indicator $I_{\mathrm{PBM}}^{\mathrm{opt}}$ (blue). (d) Estimated critical value of the tuning parameter based on $I_{\mathrm{SL}}^{\mathrm{opt}}$ (SL), $I_{\mathrm{LBC}}^{\mathrm{opt}}$ (LBC), $I_{\mathrm{PBM}}^{\mathrm{opt}}$ (PBM) as a function of the chain length L.

3.6.4 XXZ model

Having discussed classical models, we move on to the quantum case. First, we consider the spin$-\frac{1}{2}$ XXZ chain [Schollwöck *et al.*, 2008; Franchini, 2017] with open boundary conditions whose Hamiltonian is given by

$$H = \sum_{i=1}^{L-1} J(S_{i+1}^x S_i^x + S_{i+1}^y S_i^y) + \Delta S_{i+1}^z S_i^z, \tag{3.71}$$

where J is the coupling strength along the x- and y-direction and Δ is the coupling strength in the z-direction. For $\Delta/J < -1$, the XXZ chain is in the ferromagnetic phase, see Figure 3.13(a). The ground state is spanned by the two product states where all spins point either in the z or $-z$ direction which have a magnetization of $\langle M \rangle = 2\langle S_{\mathrm{tot}}^z \rangle = \pm L$. The ferromagnetic phase exhibits a broken symmetry: these states do not exhibit the discrete symmetry of spin reflection $S_i^z \to -S_i^z$ under which the Hamiltonian is invariant. For $\Delta/J > 1$, the XXZ chain is in the antiferromagnetic phase with broken symmetry and two degenerate ground states. These are product states with vanishing magnetization. For $-1 < \Delta/J < 1$ the XXZ chain is in the paramagnetic XY phase characterized by uni-axial symmetry of the easy-plane type and vanishing magnetization.

Here, we restrict our analysis to the transition between the ferromagnetic and paramagnetic XY phases. The ground states are obtained through exact diagonalization. Figures 3.13 and 3.14 shows the case when the ground state with $\langle S^z_{\text{tot}}\rangle = +L/2$ is selected in the ferromagnetic phase and S^z is chosen as a measurement basis. The quantum phase transition can be revealed by looking at the magnetization, see Figure 3.13(c). The optimal indicators of all three methods correctly highlight the phase transition, see Figure 3.14. Looking at the underlying probability distributions [see Figure 3.13(b)], the problem closely resembles the prototypical case of a bipartitioned data set (case 2 in Section 3.3). Thus, the optimal predictions and indicators also qualitatively match the results obtained in this case. In particular, the Bayes-optimal predictions of SL can be described by Equation (3.65), where the ferromagnetic ground state takes the role of the ground state energy.

Proof

We take region I to be composed of a single point γ_1. Assuming that

$$P(\boldsymbol{x}|\gamma_1) = \begin{cases} 1 \text{ if } \boldsymbol{x} = \boldsymbol{x}^*, \\ 0 \text{ otherwise}, \end{cases} \tag{3.72}$$

and following the same procedure as for a Boltzmann distribution in Section 3.6.1, we have

$$\hat{y}^{\text{opt}}_{\text{SL}}(\gamma) = \frac{P(\boldsymbol{x}^*|\gamma)}{1 + P_{\text{II}}(\boldsymbol{x}^*)}. \tag{3.73}$$

Equation (3.73) can be used to explain the optimal indicator signals of SL in the XXZ chain, where $\boldsymbol{x}^*$ corresponds to a ground state which is one of the chosen basis states.

We verified that the optimal indicators also mark the phase transition when other states from the ground state manifold, such as equal superpositions of product states with maximal magnetization along the z-direction, are selected in the ferromagnetic phase. Similarly, the phase transition is also highlighted when measurements are performed in the S^x or S^y basis instead of the S^z basis.

3.6.5 Kitaev model

The Kitaev chain is a one-dimensional model based on L spinless fermions, which undergoes a quantum phase transition between a topologically trivial and non-trivial phase [Kitaev, 2001; Alicea, 2012]. The Kitaev Hamiltonian is given by

$$H = \sum_{i=1}^{L-1} (\Delta c_{i+1} c_i - t c^\dagger_{i+1} c_i + \text{h.c.}) - \mu \sum_{i=1}^{L} n_i, \tag{3.74}$$

where we consider open boundary conditions, μ is the chemical potential, t is the hopping amplitude, and Δ is the induced superconducting gap. In the following, we set $\Delta = -t$. The ground state of this model features a quantum phase transition from a topologically trivial ($|\mu/t| > 2$) to a non-trivial state ($|\mu/t| < 2$), see Figure 3.15(a). In the topological phase, Majorana zero modes [Wilczek, 2009] are present. Here we

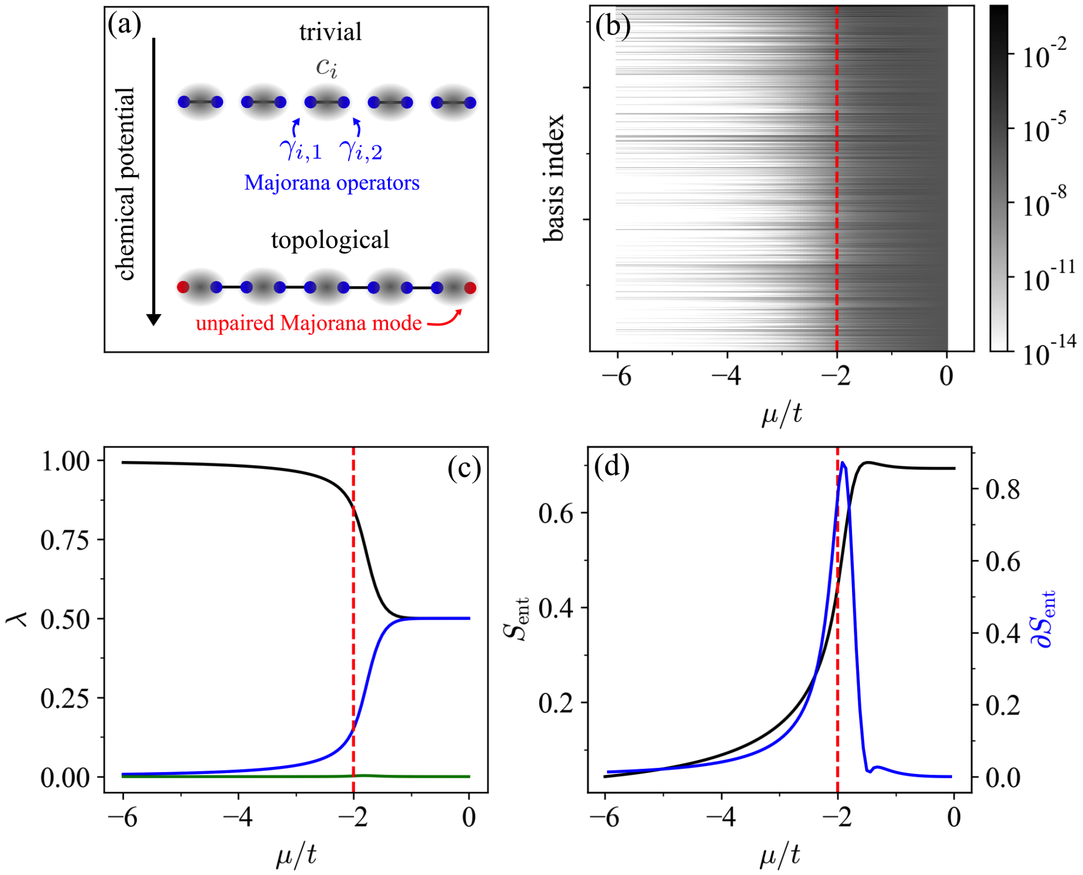

FIGURE 3.15: Results for the Kitaev chain ($L = 20$) with the dimensionless chemical potential as a tuning parameter $\gamma = \mu/t$. The critical value $\mu_c/t = -2$ is highlighted by a red dashed line. (a) Illustration of the phase transition in the Kitaev chain between a topological and trivial phase, where the Majorana operators $\gamma_{i,1}$ and $\gamma_{i,2}$ are defined by $c_i = (\gamma_{i,1} + i\gamma_{i,2})/\sqrt{2}$, $c_i^\dagger = (\gamma_{i,1} - i\gamma_{i,2})/\sqrt{2}$. (b) Probability distributions governing the input data (indices of Fock basis states) as a function of the tuning parameter, where the color scale denotes the probability. The color scale is cut off at 10^{-14} to improve visual clarity. (c) The three largest eigenvalues of ρ_{A} [Equation (3.75)] as a function of the tuning parameter. (d) Entanglement entropy S_{ent} [Equation (3.76)] (black) and its derivative with respect to the tuning parameter $\partial S_{\mathrm{ent}}/\partial\gamma$ (blue).

restrict ourselves to $\mu/t \leq 0$. We compute the ground states through exact diagonalization. We restrict ourselves to the even-particle sector whose corresponding ground state has a lower energy within the topologically trivial phase. In the topological phase, the ground state is doubly degenerate, and the two states can be distinguished by their fermionic parity. This is because of the presence of the pairing term in the Kitaev chain Hamiltonian [Equation (3.74)]. As a consequence, H does not conserve the total fermion number $N_f = \sum_{i=1}^{L} n_i$, i.e., $[H, N_f] \neq 0$. The fermion number modulo 2, however, is conserved [Katsura *et al.*, 2015].

The topologically trivial and non-trivial phase can be distinguished through entanglement spectra and the corresponding entanglement entropy [Amico *et al.*, 2008]. Consider the reduced density matrix ρ_{A} of a system in the pure state $|\Psi\rangle$ obtained by subdividing the Hilbert space $\mathcal{H}$ into two parts, A and B, and tracing out the degrees

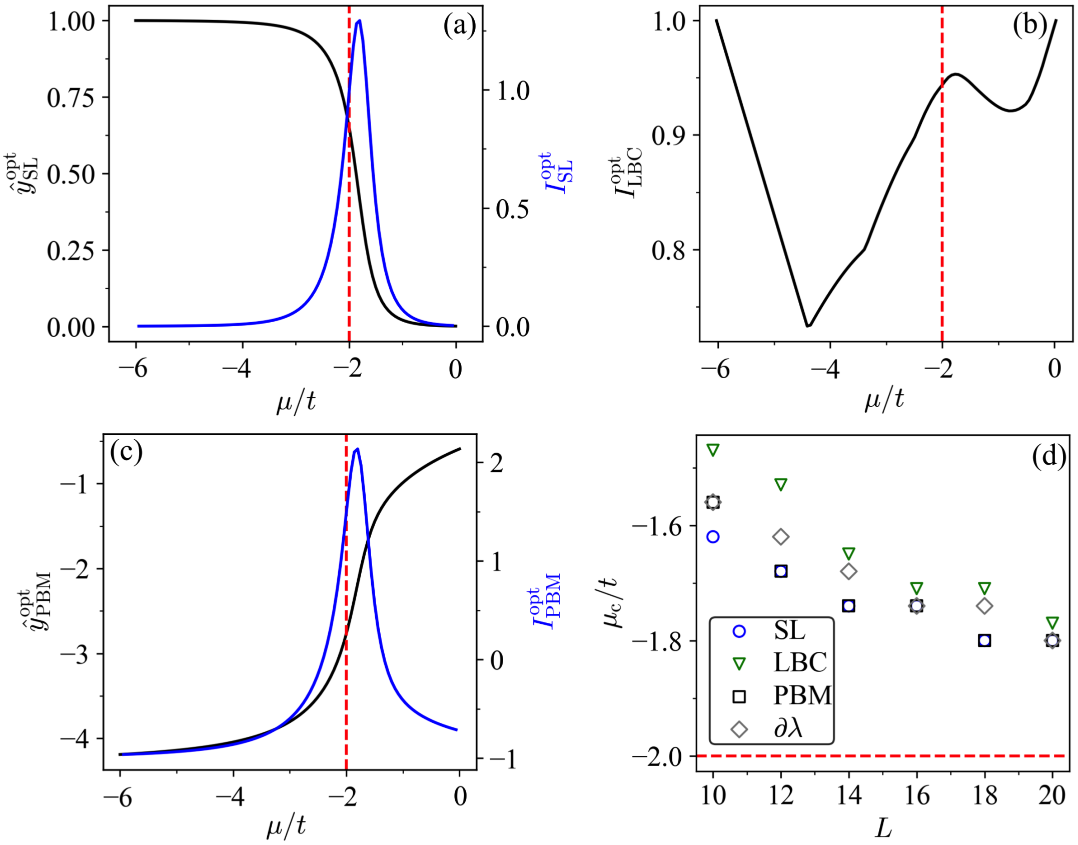

FIGURE 3.16: ML results for the Kitaev chain ($L = 20$) with the dimensionless chemical potential as a tuning parameter $\gamma = \mu/t$, where $\gamma_1 = -6$, $\gamma_K = 0$, and $\Delta\gamma = 0.06$. The critical value $\mu_c/t = -2$ is highlighted by a red dashed line. (a) Mean optimal prediction $\hat{y}^{\text{opt}}_{\text{SL}}$ in SL (black) and the corresponding indicator $I^{\text{opt}}_{\text{SL}}$ (blue). In SL, the data obtained at γ_1 and γ_K constitutes our training set, i.e., $r_{\text{I}} = 1$ and $l_{\text{II}} = K$. (b) Optimal indicator of LBC, $I^{\text{opt}}_{\text{LBC}}$ (black). (c) Mean optimal prediction $\hat{y}^{\text{opt}}_{\text{PBM}}$ in PBM (black) and the corresponding indicator $I^{\text{opt}}_{\text{PBM}}$ (blue). (d) Estimated critical value of the tuning parameter based on $I^{\text{opt}}_{\text{SL}}$ (SL), $I^{\text{opt}}_{\text{LBC}}$ (LBC), $I^{\text{opt}}_{\text{PBM}}$ (PBM), and the derivative of the largest eigenvalue of the reduced density matrix [see black line in Figure 3.15(c)] given by $\partial\lambda/\partial\gamma$ ($\partial\lambda$), as a function of the chain length L. The estimated critical value of the tuning parameter denoted by $\partial\lambda$ corresponds to the location of the maximum in $\partial\lambda/\partial\gamma$.

of freedom of B

$$\rho_{\text{A}} = \text{tr}_{\text{B}}\left[|\Psi\rangle\langle\Psi|\right], \tag{3.75}$$

with $\{\lambda_i\}_i$ the spectrum of ρ_{A} and $\{-\ln(\lambda_i)\}_i$ the entanglement spectrum. Here, we consider the bipartition of the chain into left and right halves with $L_{\text{A}} = L_{\text{B}} = L/2$. The entanglement entropy can then be computed as

$$S_{\text{ent}}(\rho_{\text{A}}) = -\sum_i \lambda_i \ln(\lambda_i). \tag{3.76}$$

The three largest eigenvalues of ρ_A are shown in Figure 3.15(c) and the resulting entanglement entropy is shown in Figure 3.15(d). Both the spectrum and entanglement entropy exhibit the largest change close to the critical value $\mu_c/t = -2$. The entanglement entropy approaches zero deep within the topologically trivial phase, signaling that the two halves of the ground state of the chain are not entangled. In the topological phase, the entanglement entropy approaches a value of ln(2) characteristic of the entangled ground state.

Figure 3.16 shows the results of SL, LBC, and PBM. The location of the local maxima of the optimal indicators based on all three methods converges to the critical value of $\mu_c/t = -2$ with increasing chain length. Considering the probability distributions governing the input data [see Figure 3.15(b)], we observe that almost all basis states become occupied with non-negligible probability as the tuning parameter μ/t is tuned across its critical value. Note that in [Van Nieuwenburg *et al.*, 2017], the phase transition in the Kitaev model was successfully revealed using LBC with NNs where the entanglement spectrum of the ground state served as an input. The scaling behavior of the estimated critical value of the tuning parameter based on the optimal indicators of SL, LBC, and PBM is comparable to standard physical indicators, such as the eigenvalues of the reduced density matrix or the entanglement entropy [see Figure 3.16(d)]. In the limit $\mu/t \to -\infty$, the ground state of the Kitaev chain corresponds to the Fock state with each site being occupied. Thus, in the limit $\mu_1/t \to -\infty$, the optimal predictions of SL follow Equation (3.65), where the aforementioned Fock state takes the role of the ground state energy (same argument as in Section 3.6.4).

Influence of finite-sample statistics

Figure 3.17 shows the optimal predictions and indicators of SL, LBC, and PBM for the Kitaev chain of length $L = 20$ given various values of $|\mathcal{D}_\gamma|$. Recall that the results for the quantum systems displayed in Section 3.6 are generally obtained based on the "ground-truth" probability distributions obtained, e.g., from exact diagonalization. Here, we explicitly sample these probability distributions, i.e., perform projective measurements and infer the probability distribution based on the measurement results. In SL and PBM, accurate estimates for the critical value of the tuning parameter can be obtained based on $|\mathcal{D}_\gamma| = 10^3$ samples, whereas $|\mathcal{D}_\gamma| = 10^4$ samples are required for a local maximum to emerge in LBC. This only covers a fraction of the total state space comprised of $|\mathcal{X}| = 524288$ states. Notice that the indicator of LBC shows a plateau close to one in the topological phase for a small number of samples, which signifies the absence of "confusion" inherent to the data. Similarly, the optimal prediction of PBM is approximately linear in the topological phase for a small number of samples, corresponding to a model that can perfectly resolve the value of the tuning parameter associated with the input. This demonstrates the fact that while the ground-truth probability distributions may have substantial overlap, estimated probabilities based on a drawn data set may not.

The high level of uncertainty in the indicator of SL and PBM compared to LBC can be attributed to the symmetric difference quotient used to approximate the derivative. Moreover, in LBC we associate a distinct optimal predictive model to each bipartition point, whereas the optimal indicator is extracted from a single optimal model in the case of SL and PBM. This leads to an additional suppression of fluctuations in the case of LBC. In the future, it will be of interest to enhance the quality of the optimal predictions and indicators based on finite data through improved derivative computations in the case of SL and PBM [Chartrand, 2011].

3.6.6 Bose-Hubbard model

Finally, we consider the many-body localization (MBL) phase transition in the one-dimensional Bose-Hubbard model (with open boundary conditions) following [Lukin

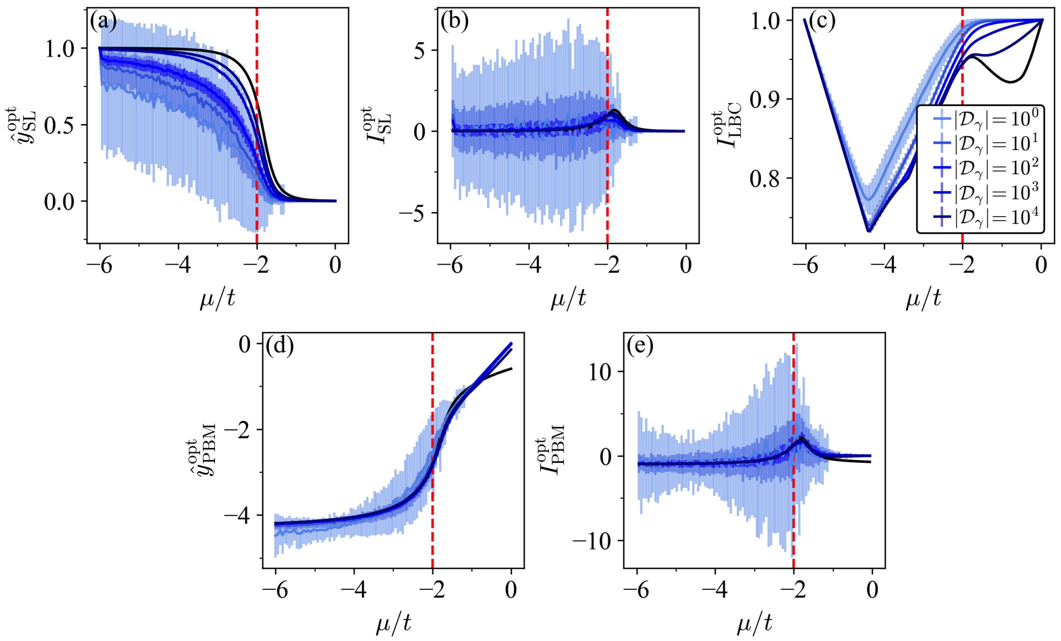

FIGURE 3.17: Optimal predictions and indicators of SL, LBC, and PBM for the Kitaev chain ($L = 20$) given various number of data points $|\mathcal{D}_\gamma|$ per sampled value of the tuning parameter $\gamma = \mu/t$, where $\gamma_1 = -6$, $\gamma_K = 0$, and $\Delta\gamma = 0.06$. The critical value $\mu_c/t = -2$ is highlighted by a red dashed line. The optimal predictions and indicators obtained based on the ground-truth probability distributions from exact diagonalization are shown in black. (a) Mean optimal prediction $\hat{y}^{\mathrm{opt}}_{\mathrm{SL}}$ in SL (the data obtained at γ_1 and γ_K constitutes our training set, i.e., $r_{\mathrm{I}} = 1$ and $l_{\mathrm{II}} = K$) and (b) the corresponding indicator $I^{\mathrm{opt}}_{\mathrm{SL}}$. (c) Optimal indicator of LBC, $I^{\mathrm{opt}}_{\mathrm{LBC}}$. (d) Mean optimal prediction $\hat{y}^{\mathrm{opt}}_{\mathrm{PBM}}$ in PBM and (f) the corresponding indicator $I^{\mathrm{opt}}_{\mathrm{PBM}}$. Here, we report results averaged over 100 independent data sets, where the error bars correspond to the standard deviation.

et al., 2019; Rispoli *et al.*, 2019; Bohrdt *et al.*, 2021]. The system is described by the Hamiltonian

$$H = -J \sum_{i=1}^{L-1} (b^\dagger_{i+1} b_i + \text{h.c.}) + \sum_{i=1}^{L} \frac{U}{2} n_i (n_i - 1) + W h_i n_i, \tag{3.77}$$

where J is the hopping strength and U is the on-site interaction strength [see top panel in Figure 3.18(a)]. Here, we fix $U/J = 2.9$. The last term in Equation (3.77) corresponds to a quasiperiodic potential $h_i = \cos(2\pi\beta i + \phi)$ mimicking on-site disorder with amplitude W, where we fix $1/\beta = 1.618$. This system transitions to the MBL phase, where thermalization breaks down as the disorder strength is increased beyond a critical value W_c/J, see the bottom panel in Figure 3.18(a). We analyze the system in the long-time limit $tJ = 100$ after unitary time-evolution starting from a Mott-insulating state with one particle per site by solving the Schrödinger equation numerically. We average over different disorder realizations obtained by sampling the phase $\phi \in [0, 2\pi)$ of the potential uniformly.

A popular way to differentiate between the thermalizing and MBL regimes relies on the study of spectral statistics using tools from random matrix theory [Pal and Huse,

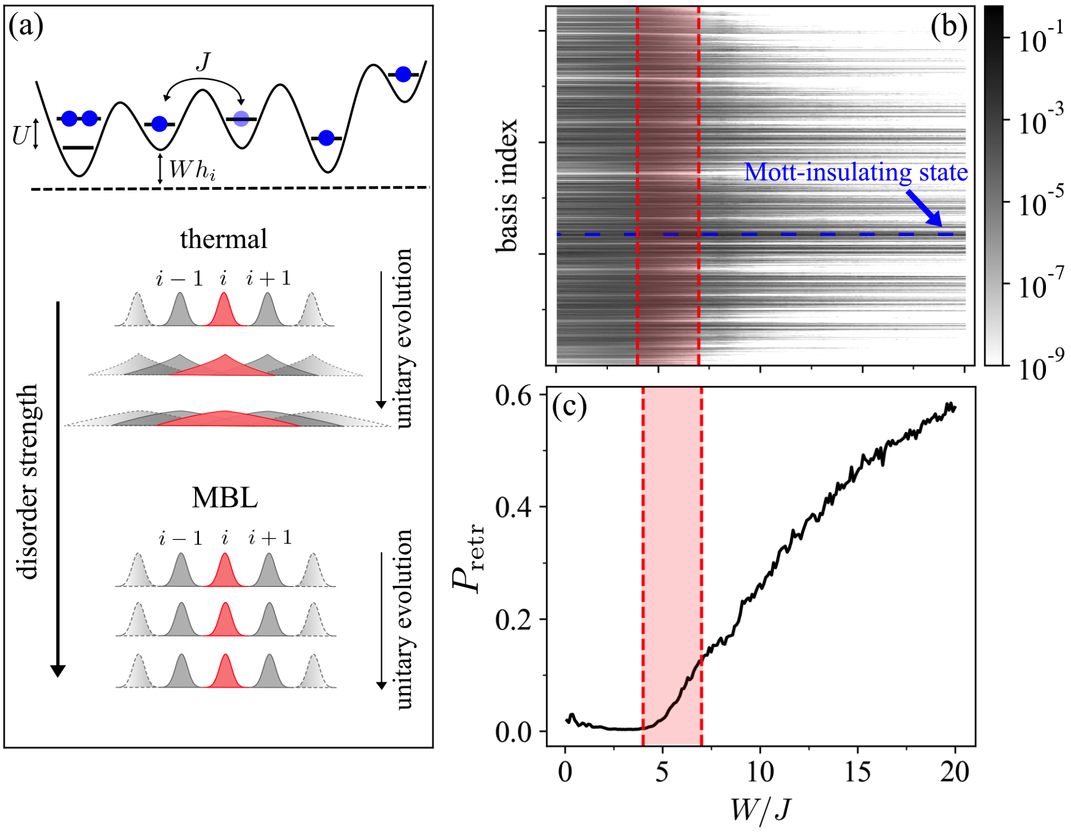

FIGURE 3.18: Results for the MBL phase transition in the one-dimensional Bose-Hubbard model ($L = 8$) with the dimensionless disorder strength as a tuning parameter $\gamma = W/J$. Here, $1.1\cdot10^3$ different disorder realizations were considered. The reference range for the critical value of the tuning parameter $W_c/J \approx 4-7$ [Rispoli *et al.*, 2019; Bohrdt *et al.*, 2021] at which the phase transition between the thermalizing and MBL phase occurs is highlighted in red. (a) Illustration of the one-dimensional Bose-Hubbard model [Equation (3.77)] (top) and the MBL phase transition (bottom), where the system is initialized in a Mott-insulating state. (b) Probability distributions governing the input data (indices of Fock basis states with $N_b = 8$ particles) as a function of the tuning parameter, where the color scale denotes the probability. The color scale is cut off at 10^{-9} to improve visual clarity. The blue dashed line highlights the initial Mott-insulating state. (c) Disorder-averaged retrieval probability P_{retr} as a function of the tuning parameter corresponding to the line-cut marked in panel (b).

2010; Khemani *et al.*, 2017; Alet and Laflorencie, 2018]. In the thermal regime, the statistical distribution of level spacings is given by a Gaussian orthogonal ensemble (GOE), while a Poisson distribution is expected for localized states. The ratio of consecutive level spacings is

$$r_i = \frac{\min\{\delta_i, \delta_{i+1}\}}{\max\{\delta_i, \delta_{i+1}\}}, \tag{3.78}$$

with $\delta_i = E_i - E_{i-1}$ at a given eigenenergy E_i. Averaging over the spectrum and multiple disorder realizations yields $\langle r\rangle$, which varies from $r_{\mathrm{GOE}} = 0.5307$ within the thermalizing phase to $r_{\mathrm{Poisson}} = 2\ln(2) - 1 \approx 0.3863$ within the MBL phase, see Figure 3.19(d).

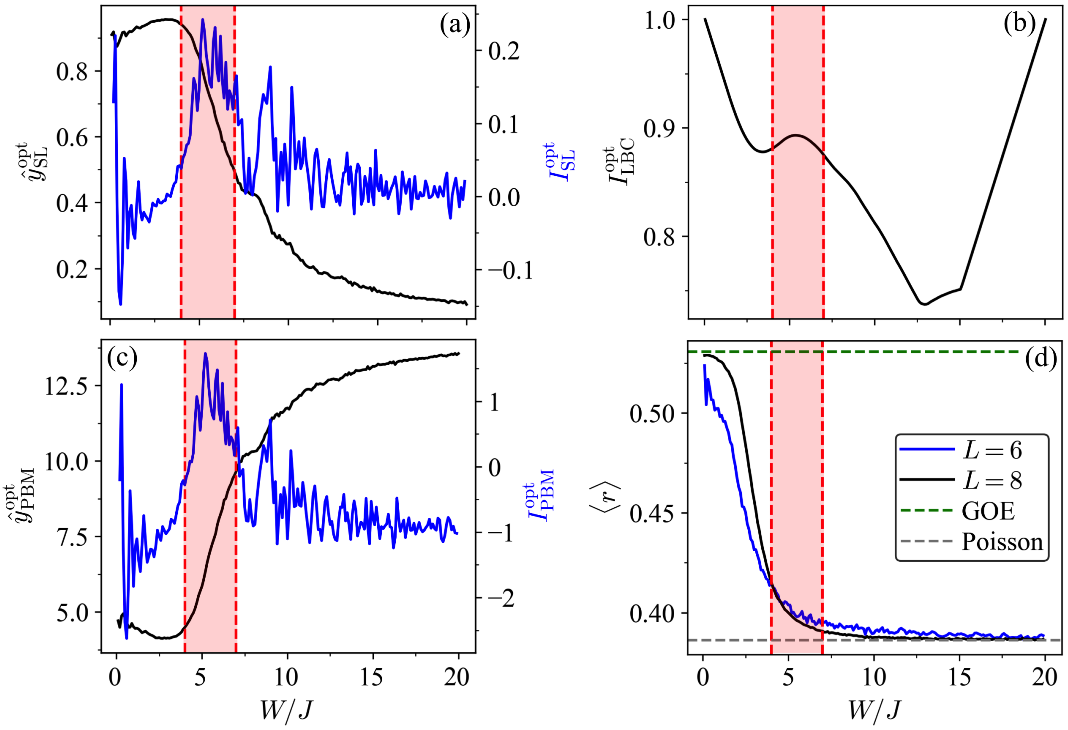

FIGURE 3.19: ML results for the MBL phase transition in the one-dimensional Bose-Hubbard model ($L = 8$) with the dimensionless disorder strength as a tuning parameter $\gamma = W/J$ ranging from $\gamma_1 = 0.1$ to $\gamma_K = 20$ in steps of $\Delta\gamma = 0.1$. Here, $1.1 \cdot 10^3$ different disorder realizations were considered. The reference range for the critical value of the tuning parameter $W_c/J \approx 4-7$ [Rispoli *et al.*, 2019; Bohrdt *et al.*, 2021] at which the phase transition between the thermalizing and MBL phase occurs is highlighted in red. (a) Mean optimal prediction $\hat{y}_{\mathrm{SL}}^{\mathrm{opt}}$ in SL (black) and the corresponding indicator $I_{\mathrm{SL}}^{\mathrm{opt}}$ (blue). In SL, the data obtained at γ_1 and γ_K constitutes our training set, i.e., $r_{\mathrm{I}} = 1$ and $l_{\mathrm{II}} = K$. (b) Optimal indicator of LBC $I_{\mathrm{LBC}}^{\mathrm{opt}}$ (black). (c) Mean optimal prediction $\hat{y}_{\mathrm{PBM}}^{\mathrm{opt}}$ in PBM (black) and the corresponding indicator $I_{\mathrm{PBM}}^{\mathrm{opt}}$ (blue). (d) Average ratio of consecutive level spacings $\langle r \rangle$ for a chain of length $L = 6$ (blue) and $L = 8$ (black) with reference values $r_{\mathrm{GOE}} = 0.5307$ (green, dashed) and $r_{\mathrm{Poisson}} = 2\ln(2) - 1 \approx 0.3863$ (grey, dashed). For averaging, we consider all eigenstates located in the middle one-third of the spectrum [Pal and Huse, 2010; Rispoli *et al.*, 2019] restricted to the subspace with $N_b = L$ particles. Additionally, we average over multiple disorder realizations (10^4 for $L = 6$ and $1.1 \cdot 10^3$ for $L = 8$).

The ML results are shown in Figure 3.19. All three methods correctly identify the MBL phase boundary, where we take $W_c/J \approx 4-7$ from [Rispoli *et al.*, 2019; Bohrdt *et al.*, 2021] as a reference. This is in agreement with the spectral analysis: the crossover between the average ratio of consecutive level spacings for systems of size $L = 6$ and $L = 8$ is located at $W_c/J \approx 4$, see Figure 3.19(d). Moreover, the phase boundary marks the range of the tuning parameter in which the most significant change in the underlying probability distribution occurs [see Figure 3.18(b)]. A line-cut along the index corresponding to the initial Mott-insulating state is shown in Figure 3.18(c). It corresponds to the disorder-averaged probability of retrieving the initial state after unitary time evolution. The MBL phase boundary is marked by the sudden increase in P_{retr} [Lukin *et al.*, 2019] which is correctly picked up by SL, LBC, and PBM.

Our results are also in agreement with [Bohrdt *et al.*, 2021], which examined the MBL phase transition within the same model using SL, PBM, and LBC with NNs on numerical and experimental data. As such, this example highlights the possibility of calculating optimal indicators directly from experimental data.

3.7 Comparison to computation using neural networks

In this section, we discuss the application of SL, LBC, and PBM with NNs to the six physical systems analyzed in Section 3.6. First, we show that one can accurately approximate the optimal predictions and indicators constructed based on probability distributions by training NNs. Next, we discuss the computational cost associated with training NNs compared to constructing and evaluating optimal predictive models. Finally, we investigate the influence of NN size, early stopping, regularization, and finite-sample statistics on the results.

3.7.1 Implementation details

For the classical systems (Ising model, IGT, and XY model), the energy $H(\boldsymbol{\sigma})$ of the spin configurations $\boldsymbol{\sigma}$ sampled from Boltzmann distributions at various temperatures serves as an input. To counteract the effect of finite-sample statistics on the predictions in the case of SL due to inputs not contained in the training set $\boldsymbol{x}^* \notin \bar{\mathcal{T}}$, i.e., $\bar{\mathcal{D}} \neq \bar{\mathcal{T}}$, we modify the corresponding probability $P(\boldsymbol{x}^*|\gamma_K)$ from $0 \mapsto 1/\left(|\mathcal{D}_{\gamma_K}| + |\bar{\mathcal{D}} \setminus \bar{\mathcal{T}}|\right)$. Here, $|\bar{\mathcal{D}} \setminus \bar{\mathcal{T}}|$ denotes the number of such unique inputs at γ_K. That is, we add a single instance of each sample that does not appear at the boundary point γ_K to the corresponding data set $\mathcal{D}_{\gamma_K}$. This makes use of our knowledge that the Boltzmann distribution becomes uniform as $T \to \infty$. Alternatively, we could set these predictions to zero as discussed in Appendix B. Both options ensure that the empirically optimal predictions can be recovered using NNs.[11]

For the quantum systems (XXZ chain, Kitaev chain, and Bose-Hubbard model), the index of the corresponding basis states serves as input where we use a one-hot encoding. The S^z eigenstate given by $|\uparrow\downarrow \ldots \uparrow\rangle$ and the Fock state $|10\ldots1\rangle$, for example, are encoded as a bit-string $\boldsymbol{x} = (10\ldots1)$. Before NN training, we standardize the input, see Section 2.5.4. Note that this bijective mapping does not change the probability associated with each input. Therefore, the optimal predictions and indicators remain unchanged.

The NNs used in this chapter consist of a series of fully-connected layers with ReLUs as activation functions, as described in Section 2.5.4. For the prototypical probability distributions discussed in Section 3.3, we use a single hidden layer with 64 nodes. The number of hidden layers and nodes utilized for analyzing all other physical systems from Section 3.6 are reported in the corresponding figure captions.

We use Flux in `Julia` to implement and train the NNs. The weights and biases are optimized via gradient descent with Adam [Kingma and Ba, 2014] to minimize the loss function over a series of training epochs. In SL and LBC, we train on a CE loss function [Equation (2.7) and (2.10), respectively], whereas in PBM we train on an MSE loss function [Equation (2.12)]. Gradients are calculated using backpropagation [Rumelhart *et al.*, 1986; Goodfellow *et al.*, 2016; Baydin *et al.*, 2018]. For

[11] While the NN-based indicator can change if no such modifications are performed, this does not resolve the instances where the optimal indicator of SL fails to locate the phase transition (such as in the Ising model or XY model).

the prototypical probability distributions discussed in Section 3.3, we train for 10000 epochs with a learning rate of 0.001. The number of training epochs and learning rate for all other models is reported in the corresponding figure captions.

3.7.2 Reproducing optimal predictions

The predictions and indicators of the three methods obtained using NNs after long training for all six physical systems considered in Section 3.6 are shown in Appendix D for completeness. We chose the smallest system sizes for convenience. Overall, they are in excellent agreement with the corresponding optimal predictions and indicators. As the system size is increased, it becomes increasingly difficult to approximate the corresponding optimal predictions and indicators with high accuracy because the NN size has to be increased systematically and other hyperparameters need to be adjusted more carefully. However, even for the largest system sizes considered in this chapter, qualitative agreement can still be achieved with moderate NN sizes, see Figure 3.21 for an explicit example.

3.7.3 Computation time

The measured computation times associated with calculating the optimal indicators of phase transitions of SL, LBC, and PBM, for all six physical systems discussed in Section 3.6 are reported in Table 3.1. The corresponding `Julia` code is open source [Arnold and Schäfer, 2022a]. We do not consider the computational cost associated with generating samples and estimating the underlying probability distributions.

Overall, the computation times are remarkably low. For all systems, the optimal indicator of SL and PBM can be obtained in under a second, and the optimal indicator of LBC in under a minute. We observe that the computation times of SL and PBM are comparable, with PBM being slightly slower than SL. In contrast, the computation times of LBC are two orders of magnitude larger. Note that these are the evaluation times corresponding to the largest system sizes under consideration. The computation times qualitatively agree with the complexity analysis described in Section 3.4. An additional speed-up can be gained through parallel execution. In particular, it is straightforward to compute optimal predictions (in the case of SL and PBM) and optimal indicators (in the case of LBC) at discrete values of the tuning parameter in parallel, e.g., via multithreading (which is implemented in [Arnold and Schäfer, 2022a]).

	Ising (Sec. 3.6.1)	IGT (Sec. 3.6.2)	XY (Sec. 3.6.3)	XXZ (Sec. 3.6.4)	Kitaev (Sec. 3.6.5)	Bose-Hubbard (Sec. 3.6.6)
$t_{\text{SL}}^{\text{opt}}$	0.0007 ± 0.0002	0.00007 ± 0.00002	0.00012 ± 0.00003	0.0049 ± 0.0009	0.17 ± 0.02	0.0044 ± 0.0009
$t_{\text{SL}}^{\text{NN}}$	0.00060 ± 0.00005	0.00030 ± 0.00002	0.00048 ± 0.00003	0.0060 ± 0.0009	0.14 ± 0.02	0.0023 ± 0.0003
$t_{\text{SL}}^{\text{NN}}/t_{\text{SL}}^{\text{opt}}$	0.9 ± 0.3	4.9 ± 1.3	4.0 ± 0.8	1.2 ± 0.3	0.9 ± 0.2	0.5 ± 0.1
$t_{\text{PBM}}^{\text{opt}}$	0.0016 ± 0.0004	0.00014 ± 0.00006	0.00021 ± 0.00008	0.019 ± 0.003	0.42 ± 0.05	0.009 ± 0.002
$t_{\text{PBM}}^{\text{NN}}$	0.0042 ± 0.0007	0.0005 ± 0.0001	0.00084 ± 0.00004	0.080 ± 0.006	1.2 ± 0.1	0.026 ± 0.004
$t_{\text{PBM}}^{\text{NN}}/t_{\text{PBM}}^{\text{opt}}$	2.7 ± 0.8	4.0 ± 2.1	4.0 ± 1.5	4.2 ± 0.8	2.8 ± 0.4	2.7 ± 0.6
$t_{\text{LBC}}^{\text{opt}}$	0.8 ± 0.1	0.042 ± 0.001	0.041 ± 0.004	3.7 ± 0.4	32.0 ± 1.7	1.4 ± 0.2
$t_{\text{LBC}}^{\text{NN}}$	1.11 ± 0.06	0.09 ± 0.01	0.12 ± 0.01	12.2 ± 1.2	93.9 ± 3.8	3.2 ± 0.4
$t_{\text{LBC}}^{\text{NN}}/t_{\text{LBC}}^{\text{opt}}$	1.3 ± 0.2	2.1 ± 0.2	2.8 ± 0.4	3.3 ± 0.5	3.0 ± 0.2	2.4 ± 0.5
$t_{\text{PBM}}^{\text{opt}}/t_{\text{SL}}^{\text{opt}}$	2.3 ± 0.9	2.0 ± 1.1	1.8 ± 0.7	3.8 ± 1.0	2.7 ± 0.5	2.1 ± 0.6
$t_{\text{LBC}}^{\text{opt}}/t_{\text{SL}}^{\text{opt}}$	1231 ± 374	629 ± 164	346 ± 77	751 ± 158	204 ± 33	308 ± 78
L	60	28	60	14	20	8
$\lvert\bar{\mathcal{D}}\rvert$	1711	353	1000	16384	524288	6435
$\lvert\Gamma\rvert$	200	100	100	201	101	200

TABLE 3.1: Measured computations times in seconds associated with constructing and evaluating optimal models, t^{opt}, or training an NN of minimal size (one hidden layer with a single node) for a single epoch, t^{NN}, for all three methods and six systems discussed in Section 3.6. The linear system size L, the total number of unique observed samples $|\bar{\mathcal{D}}|$, as well as the number of sampled values of the tuning parameter $|\Gamma|$ for each system are also reported. The construction and evaluation of the optimal models yields the optimal predictions, optimal indicator, and optimal loss value. A training epoch is comprised of evaluating the NN at all $|\bar{\mathcal{D}}|$ unique samples, calculating the loss function, obtaining the gradient via backpropagation, and performing a single gradient step. For details on the NN architecture and training, see Section 3.7.1. Note that in LBC, $t_{\text{LBC}}^{\text{NN}}$ corresponds to $|\Gamma| + 1$ times the computation time of a training epoch for a single NN. All computation times were measured on a single CPU [Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz] and garbage collection times were subtracted from the total runtime. To gather statistics, for each method and system, computations were ran for 20 hours. If 10^5 independent runs were completed in less than 20 hours, the computations were stopped prematurely. The error corresponds to the observed standard deviation.

Next, let us touch upon the computational cost of training NNs. Table 3.1 reports the measured computation times associated with training an NN with one hidden layer composed of a single node for one epoch. A training epoch is comprised of evaluating the NN (or NNs in the case of LBC) at all $|\bar{\mathcal{D}}|$ unique samples contained in the dataset (see Table 3.1), calculating the loss function, obtaining the gradient via backpropagation, and performing a single gradient step. This represents a lower bound for the total computation time associated with obtaining NN-based predictions and indicators. In a typical application, larger NNs need to be used, the NNs need to be trained for multiple epochs, the NN parameters (or the corresponding predictions and indicator) need to be cached at regular intervals, hyperparameters need to be tuned, and finally the indicator needs to be computed based on the NN predictions. The computation time for a single epoch is also expected to increase if the data is processed in a batch-wise fashion (albeit likely at the benefit of requiring fewer training epochs overall). We find that this lower bound on the training time is comparable with the evaluation time of the corresponding optimal predictions and indicators (and optimal loss) and the two times differ by less than an order of magnitude across all six physical systems studied in this chapter. This empirical finding can be explained as follows: To construct the optimal model, the probability of all inputs needs to be evaluated. Similarly, in each training epoch, the NN is evaluated at all inputs contained in the training data set. The computation time associated with evaluating a small NN for a given input is comparable with evaluating the corresponding optimal model prediction, and the overhead associated with the gradient computation via backpropagation is of the same order of magnitude as the NN forward pass [Blayo *et al.*, 2014].

Suppose one is interested in the predictions and indicators of SL, PBM, and LBC, in the limit of a perfectly trained, highly expressive NNs. Evidently, based on the discussion above, the evaluation of the analytical expressions is generally more efficient in that case. The precise timings will depend on the particular implementation, as well as the choice of hyperparameters. However, even in the case where small NNs are trained for short times the computation time associated with constructing and evaluating an optimal model is *at worst* comparable.

Here, we have neglected any overhead associated with constructing probability distributions based on drawn samples. In principle, when using NN one does not rely on the estimated probability distributions, i.e., one can directly work with the unprocessed dataset. Note, however, that in many scenarios (including this chapter) the overhead of estimated probability distributions from the dataset is negligible. When studying quantum systems using exact diagonalization, one has direct access to the underlying probability distributions. Similarly, when performing Monte Carlo studies the energy statistics are readily available.

3.7.4 Controlling model capacity

In the following, we investigate the effect of NN size, training time, and ℓ_2 regularization on the NN-based predictions and indicators and compare them with the corresponding optimal predictions and indicators. All three factors influence the capacity of the resulting model and thus determine its ability to approximate the optimal predictive model realizing the global minimum of the loss function, i.e., the optimal predictions and indicators, see Figure 3.20. As pointed out in Section 3.6.1, there are instances where the optimal model does not correctly highlight a phase transition whereas simpler models do.

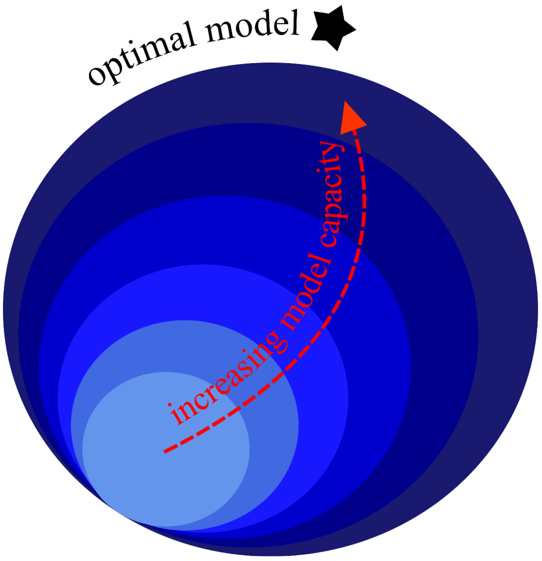

FIGURE 3.20: Illustration of how models with increasing capacity more closely approximate the optimal model, i.e., the predictions that minimize the corresponding loss function. Here, circles of distinct colors denote predictive models in function space that can be approximated accurately with an NN of a given capacity.

As an example, let us consider the application of PBM to the Ising model. Figure 3.21 shows the results for a 60 × 60 lattice obtained with NNs composed of a single hidden layer with a variable number of hidden nodes ranging from 2 to 2048. Figures 3.21(a) and (d) show the corresponding NN-based predictions and indicators after training for 10000 epochs. For NNs with 2 and 8 nodes, the indicator shows a clear peak at the critical value of the tuning parameter. As the number of nodes increases, the NN results start to resemble the optimal predictions and indicators (black) more closely. This reflects the fact that the expressivity of an NN increases as the number of nodes is increased. A similar behavior is also visible in Figure 3.21(g) which shows the loss over time, where NNs with more than 8 nodes achieve values close to the optimal loss (black), i.e., the global minimum.

Figures 3.21(b) and (e) show the predictions and indicators for the smallest NN (2 hidden nodes) evaluated at various training epochs. Here, the indicator gradually converges toward its final form, which exhibits a peak at the critical value of the tuning parameter. Similarly, Figures 3.21(c) and (f) show the results for the largest NN (2048 hidden nodes). Here, early on during training, the indicator is sharply peaked near the critical value of the tuning parameter. As the training progresses, the indicator signal starts to wash out and converge to the optimal indicator signal. The evolution of the global maximum of the indicator signal as a function of the training epoch for the various NN sizes is shown in Figure 3.21(h). These results quantify how accurately the estimated critical value of the tuning parameter based on the optimal indicator (black) is reproduced for a given NN size and training time.

Figure 3.21(f) shows that even for the large NNs there seems to be an intermediate period during training where the indicator peaks near the critical value of the tuning parameter, correctly highlighting the phase transition. Looking at Figure 3.21(g), during these intermediate periods the corresponding loss function starts to saturate and displays a kink. This suggests a procedure for early stopping, where the training is stopped once a kink in the loss function is observed. Early stopping based on

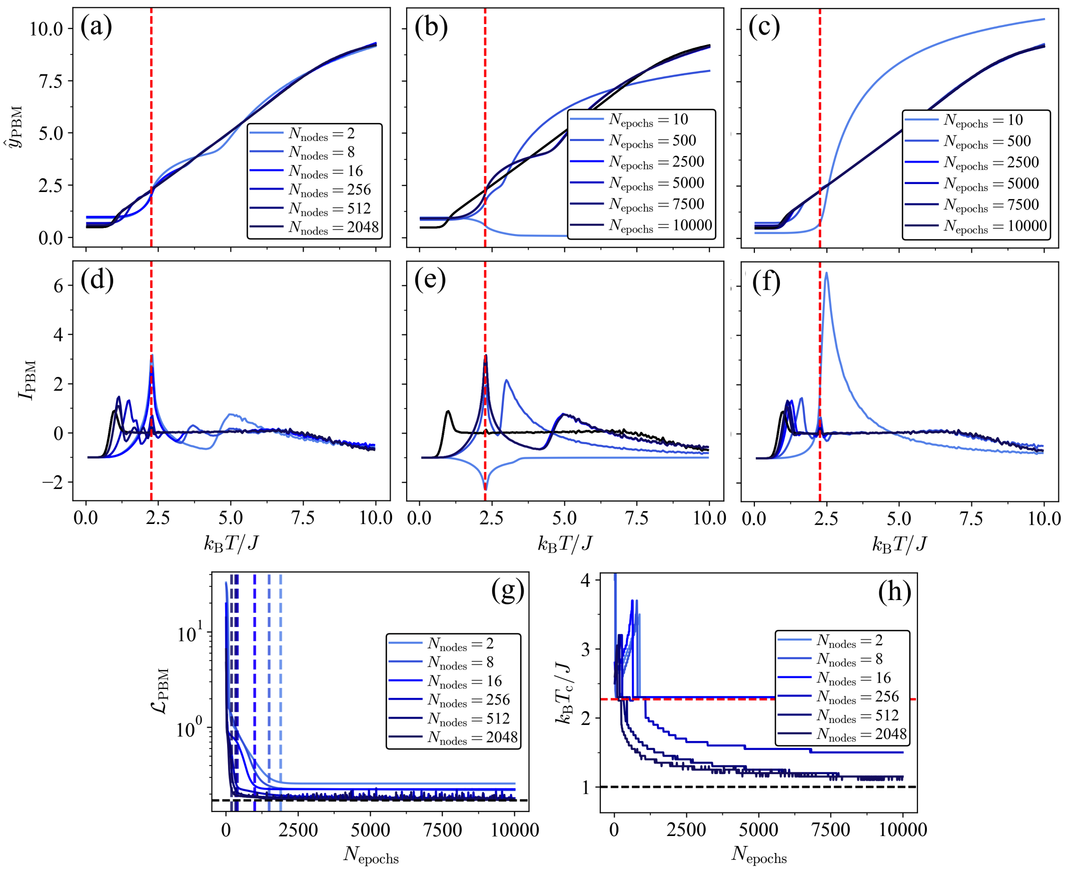

FIGURE 3.21: Results for the Ising model ($L = 60$) of PBM using NNs with a single hidden layer composed of different number of hidden nodes $N_{\rm nodes}$. The learning rate is set to 0.01. The tuning parameter ranges from $\gamma_1 = 0.05$ to $\gamma_K = 10$ with $\Delta\gamma = 0.05$. The critical value of the tuning parameter $\gamma_{\rm c} = k_{\rm B}T_{\rm c}/J$ is highlighted in red. The optimal predictions, optimal indicator, optimal loss, and corresponding estimated critical value of the tuning parameter are highlighted in black. (a),(d) Mean prediction $\hat{y}_{\rm PBM}(\gamma)$ of PBM obtained using NNs after training for 10000 epochs, as well as the corresponding indicator $I_{\rm PBM}(\gamma)$. (b),(e) Mean prediction $\hat{y}_{\rm PBM}(\gamma)$ of PBM obtained using an NN with $N_{\rm nodes} = 2$ at various stages during training, as well as the corresponding indicator $I_{\rm PBM}(\gamma)$. (c),(f) Mean prediction $\hat{y}_{\rm PBM}(\gamma)$ of PBM obtained using an NN with $N_{\rm nodes} = 2048$ at various stages during training, as well as the corresponding indicator $I_{\rm PBM}(\gamma)$. (g) Loss $\mathcal{L}_{\rm PBM}$ as a function of the number of training epochs $N_{\rm epochs}$. The location of kinks in the loss is marked by vertical dashed lines. (h) Estimated critical value of the tuning parameter as a function of the number of training epochs.

the validation loss will be discussed in the subsequent section (see Section 3.7.5). During training, the model capacity increases as visible by the steady decrease in the corresponding loss: initially, the model cannot resolve anything, in the intermediate stages it can resolve between the two phases leading to the sharp peak, and eventually it approaches the optimal predictive model (which, in this case, does not correctly highlight the phase transition). By stopping the training at the intermediate stage (i.e., selecting the corresponding NN parameters after the training is complete) a model of intermediate resolution can be obtained. Thus, early stopping acts as an implicit regularization. In the case of PBM, stopping the training early yields an NN whose indicator peaks near the critical temperature of the Ising model. However, this is not always the case. In LBC, for example, the estimated critical temperature gradually improves during training, i.e., as the model capacity increases. Recall that the optimal indicator of LBC correctly highlights the phase transition. Qualitatively similar results can be obtained for the other methods and systems. In particular, in the Ising model and XY model, we find that the indicators of SL and PBM both show a clear peak near the critical transition temperature early on during training around the epochs marked by a kink in the loss function. The peak locations of the corresponding NN-based indicator signals coincide with the signals of physical indicators, such as the heat capacity or magnetization.

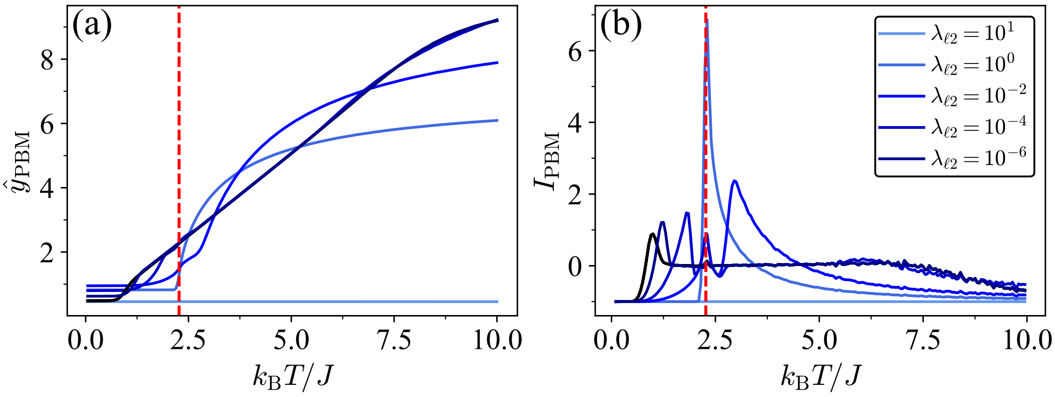

FIGURE 3.22: (a) Mean prediction $\hat{y}_{\rm PBM}(\gamma)$ and (b) the corresponding indicator $I_{\rm PBM}(\gamma)$ of PBM for the Ising model ($L = 60$) using NNs obtained after long training for various regularization strengths $\lambda_{\ell 2}$. The tuning parameter ranges from $\gamma_1 = 0.05$ to $\gamma_K = 10$ with $\Delta\gamma = 0.05$. The critical value of the tuning parameter $\gamma_{\rm c} = k_{\rm B}T_{\rm c}/J$ is highlighted in red. The optimal predictions and indicator are highlighted in black. Each NN has a single hidden layer with 2048 nodes and is trained for 10000 epochs with a learning rate of 0.01.

Lastly, we can also control the capacity of our model through explicit ℓ_2 regularization [Goodfellow *et al.*, 2016]

$$\mathcal{L} \to \mathcal{L} + \lambda_{\ell 2} \sum_i \theta_i^2, \tag{3.79}$$

where the sum runs over all tunable parameters θ_i of the NN and $\lambda_{\ell 2}$ is the regularization strength. Figure 3.22 shows the NN-based predictions and indicators of PBM for the Ising model after training with various regularization strengths. At large regularization strength, the resulting model cannot resolve any structure leading to a flat indicator signal. At an intermediate regularization strength, the resulting model can distinguish between the two phases leading to a clear peak in the indicator signal at the critical temperature of the Ising model. As the regularization strength is decreased further, the resulting model becomes more complex and converges toward the optimal model that minimizes the loss function in the absence of regularization. Consequently, the predictions and indicators converge toward the optimal predictions and indicators. In the Ising model, we thus find that explicit regularization helps to construct a model of intermediate resolution whose indicator correctly highlights the critical temperature (similarly for SL). However, as mentioned above, models with restricted capacity may not always highlight the critical value of the tuning parameter correctly. In the IGT, for example, the indicator of regularized NNs tends to display an erroneous peak similar to the heat capacity, see Figures 3.7 and 3.8.

3.7.5 Finite-sample statistics and splitting data into distinct sets

Here, we investigate NN-based predictions and indicators in the case where only a limited amount of data is available. In particular, we discuss the effect of splitting the data into a training, validation, and test set. Recall that in the limit of infinite data, the training, validation, and test set will coincide as they are all sampled independently from the same probability distribution underlying the physical system. Therefore, in the limit of sufficient data the training, validation, and test losses will decrease in lockstep during training. This is illustrated in Figure 3.23 which shows the training and validation loss of PBM for the Kitaev chain for different data set sizes.[12] For small data sets, the training, validation, and test sets can differ, resulting in differing training, validation, and test losses. In particular, one can observe a characteristic increase of the validation loss after a certain time period attributed to overfitting [Goodfellow *et al.*, 2016]. This allows one to perform early stopping such that the minimum in the validation loss is realized. Note that the location of the minima in the validation loss coincides with the kink in the corresponding training loss. The sharp local minimum in the validation loss fades as the data set size is increased further, leaving only the corresponding kinks in the training loss as a signal for early stopping. The latter situation has been discussed in Section 3.7.4. Therefore, a splitting into training, validation, and test set may allow for a clearer signal to perform early stopping given a small data set.

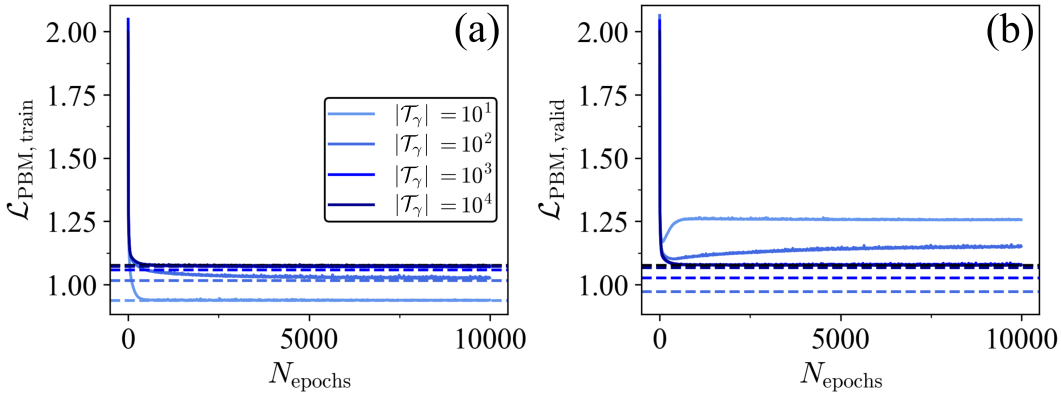

FIGURE 3.23: (a) Training loss and (b) validation loss as a function of the number of training epochs of PBM for the Kitaev chain ($L = 14$) using an NN composed of a single hidden layer with 128 nodes for various numbers of training samples $|\mathcal{T}_\gamma|$ per parameter value, where $|\mathcal{V}_\gamma| = |\mathcal{T}_\gamma|/5$. The corresponding optimal loss based on the training or validation data set is highlighted by a colored dashed line. The optimal loss based on the ground-truth probability distributions is highlighted in black. The test loss shows the same behavior as the validation loss. Each NN is trained for 10000 epochs with a learning rate of 0.01. The results averaged over 10 independent data sets.

[12] The training, validation, or test loss refers to the loss function [here Equation (2.12)] evaluated using the corresponding dataset [i.e., replacing $\mathcal{T}$ by $\mathcal{V}$ or $\mathcal{E}$ in Equation (2.12)].

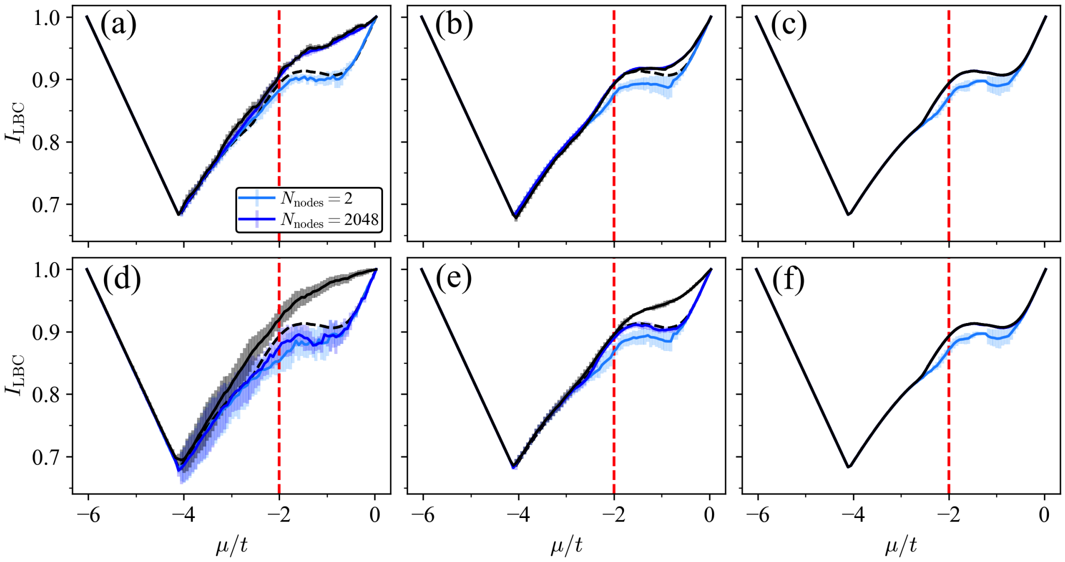

FIGURE 3.24: Results of LBC for the Kitaev chain ($L = 10$) using NNs composed of a single hidden layer with 2 or 2048 nodes for various numbers of training samples $|\mathcal{T}_\gamma|$ per parameter value. The tuning parameter $\gamma = \mu/t$ ranges from $\gamma_1 = -6$ to $\gamma_K = 0$ with $\Delta\gamma = 0.06$. The critical value $\mu_c/t = -2$ is highlighted by a red dashed line. The optimal indicator obtained based on the corresponding data set or the ground-truth probability distributions is highlighted by a black solid or dashed line, respectively. (a)-(c) Indicator I_{LBC} of LBC evaluated on the training set, $\mathcal{E} = \mathcal{T}$, for (a) $|\mathcal{T}_\gamma| = 10$, (b) $|\mathcal{T}_\gamma| = 10^2$, and (c) $|\mathcal{T}_\gamma| = 10^5$, where the NN-based predictions are obtained after complete training (without a validation set). (d)-(f) Indicator I_{LBC} of LBC evaluated on a separate test set, $|\mathcal{E}| = |\mathcal{T}|/5$ (Here, for the optimal indicator shown as a dashed line, the "test set" is used to construct the per-sample predictions as well as for evaluating mean predictions at a given parameter value.), for (d) $|\mathcal{T}_\gamma| = 10$, (e) $|\mathcal{T}_\gamma| = 10^2$, and (f) $|\mathcal{T}_\gamma| = 10^5$, where early stopping is performed by minimizing the validation loss ($|\mathcal{V}| = |\mathcal{T}|/5$). Similar results are obtained when evaluating the NNs at the end of training instead. Each NN is trained for 10000 epochs with a learning rate of 0.005. The results averaged over 10 independent data sets and the error bars are given by the standard deviation.

Another effect arising when a limited amount of data is available and finite-sample statistics play a role is best illustrated by investigating the Kitaev chain using LBC. Figure 3.24 shows the NN-based indicator signal of LBC obtained for training, validation, and test sets of various sizes. For small data set sizes [see Figure 3.24(a) and (d)] the optimal indicator (black, solid) shows no local maximum due to the negligible overlap in the inferred probability distribution. The NN-based indicator of a sufficiently large NN closely matches the optimal indicator on the training set after training [Figure 3.24(a)], whereas a small NN is incapable of approximating the optimal indicator on the training set. However, interestingly the indicator signal of the small NN qualitatively matches the optimal indicator signal based on the ground-truth probability distributions. In particular, it features a local maximum allowing for an estimate of the critical value of the tuning parameter to be obtained. This is another example illustrating how simple models can lead to sharp indicator signals. While the inferred probability distribution only has a marginal overlap in the topological phase resulting in the absence of a local maximum in the optimal indicator signal (black), the data may be partially indistinguishable to a simple model. This illustrates how "confusion" can also arise due to models with restricted expressivity.

The same phenomenon can also be observed for the indicator signal of the large NN evaluated on the test set (or validation set), see Figure 3.24(d). Here, the confusion arises because the predictions for the unseen data within the validation and test set are sub-optimal. In the future, it will be of interest to investigate whether this effect can be mimicked through appropriate interpolation of the optimal predictions [Jacot *et al.*, 2018; Greplova *et al.*, 2020; Huang *et al.*, 2022b]. Figures 3.24(b) and (e) as well as Figures 3.24(c) and (f) show how the discrepancy between the optimal indicator signal based on a finite data set and the NN-based indicator vanishes for the large NN as the data set size increases. This arises because eventually, the training, validation, and test sets become indistinguishable. The discrepancy does, however, persist for the small NN.

3.8 Discussion

In Section 3.6, we have demonstrated that the optimal indicators of SL, LBC, and PBM successfully detect phase transitions and crossovers in a variety of different classical and quantum systems based on numerical data. Recall that the optimal predictions correspond to an optimal model that reaches the global minimum of the loss function. *A priori*, it is unclear if such optimal predictors can be recovered in practice when training NNs, because the employed NNs are of finite size and local optimization techniques are used. In Section 3.7.2, we demonstrate that the optimal predictions and indicators of all six systems studied in Section 3.6 can indeed be recovered by training NNs. This reachability further underpins the practical relevance of our analysis for the case when using SL, LBC, and PBM with NNs.

In a traditional NN-based approach, one searches for the optimal model by iteratively updating the parameters of an NN to minimize a loss function [see step 2) in Figure 3.1]. In contrast, our numerical routine based on the derived analytical expressions allows for the optimal model to be constructed directly from data [see step 2*) in Figure 3.1]. As such, evaluating the analytical predictors also compares favorably to the NN-based approach in terms of computation time. For each of the three methods and across all six studied physical systems, we find that the time needed to train an NN of *minimal size* (one hidden layer with a single node) for a *single epoch* is of the same order of magnitude as the time needed to compute the optimal predictions, optimal indicator, and optimal loss (see Table 3.1). Therefore, the computation time associated with constructing and evaluating an optimal model is *at worst* comparable with training and evaluating an NN-based model. In practice, however, the latter approach typically requires significantly more computation time because larger NNs need to be used, the training takes many epochs, and hyperparameters need to be adjusted. In particular, as the system size increases and the associated state space grows, converging to the global minimum of the loss function can become increasingly difficult. The convergence of the optimal model, on the other hand, is guaranteed *by construction*.

We have observed that the optimal indicator of a given method may fail to correctly highlight a phase transition. A failure can, for example, occur if only a limited amount of data is available and finite-sample statistics dominate. In this case, while the ground-truth probability distributions underlying the data show a significant overlap resulting in a peak in the Bayes-optimal indicator signal, the inferred probability distributions do not. However, even if the data set is sufficiently large, i.e., the ground-truth probability distributions are well approximated, the empirically optimal model can fail (see classical systems in Section 3.6 for examples). Both instances of failure can often be resolved by employing non-optimal models. Such a model can be realized by an NN whose capacity, i.e., its ability to fit a wide variety of functions, is restricted. This can be achieved, e.g., by reducing the NN size, performing early stopping, or the explicit addition of ℓ_2 regularization. In these instances, other phase-classification methods that are inherently based on notions of similarity of input data [Wang, 2016; Rodriguez-Nieva and Scheurer, 2019; Kottmann *et al.*, 2020; Arnold *et al.*, 2021] are also expected to provide valuable insights. These methods stand in contrast to the optimal predictors of SL, LBC, and PBM, which are not explicitly based on learning order parameters, i.e., recognizing prevalent patterns or orderings. Instead, the optimal predictors gauge changes in the probability distributions governing the data.

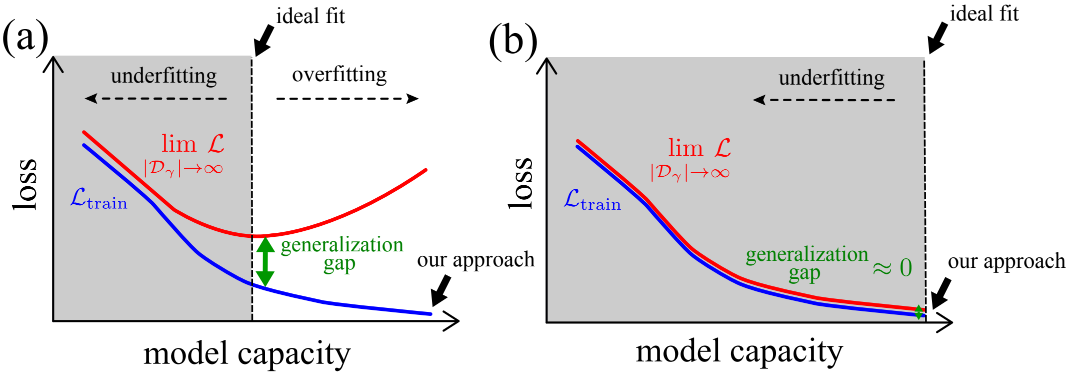
(a)
ideal fit
underfitting
overfitting
loss
$\lim_{|\mathcal{D}_\gamma|\to\infty} \mathcal{L}$
$\mathcal{L}_{\text{train}}$
generalization gap
our approach
model capacity
(b)
ideal fit
underfitting
loss
$\lim_{|\mathcal{D}_\gamma|\to\infty} \mathcal{L}$
$\mathcal{L}_{\text{train}}$
generalization gap ≈ 0
our approach
model capacity

FIGURE 3.25: Sketch of how the loss changes as a function of the model capacity in a scenario with (a) few data and (b) lots of data. The ideal fit corresponds to the minimum of the true loss $\mathcal{L}$, i.e., the loss obtained in the infinite-data limit. Our approach of constructing predictions that minimize the training loss $\mathcal{L}_{\text{train}}$ yields an empirically optimal model. If little data is available, globally minimizing the training loss generally yields a model that is far from achieving a minimum of the true loss. Given lots of data, however, the training loss is close to the true loss. Thus, the empirically optimal model constructed by our approach is close to being Bayes optimal.

Contrary to popular opinion, the failure of optimal models, or equivalently high-capacity NNs, does not always correspond to overfitting in the traditional sense [Goodfellow *et al.*, 2016]: the gap between training and test loss[13] vanishes in the limit of a sufficiently large data set (which is available for the examples discussed in Section 3.5). Therefore, sub-optimal models, such as NNs with insufficient capacity, are in fact *underfitting* the data. Figure 3.25 illustrates this scenario. This fact signals a fundamental mismatch between the classification or regression task underlying a particular ML method, i.e., the corresponding loss function, and the goal of detecting phase transitions. In particular, it raises the intriguing question of whether one can adjust the learning task in SL, PBM, and LBC such that the corresponding optimal models also correctly highlight the phase transition in these problematic cases, e.g., through an appropriate modification of the underlying loss functions, by enforcing

[13] This is a proxy for the generalization gap, i.e., the difference between the training loss and the true loss in the limit of infinite data.

explicit constraints, or by altering the indicator computation. We will explore such modifications in Chapter 4.

3.9 Summary

The ML methods for detecting phase transitions from data given by SL, LBC, and PBM can be viewed under a unifying light: all three approaches have predictive models, such as NNs, at their heart which are trained to solve a given classification or regression task. Analyzing their predictions allows us to compute a scalar indicator that highlights phase boundaries. The power and success of these methods are largely attributed to the universal function approximation capabilities of their underlying NNs. However, the more expressive an ML model, such as an NN, the more resources are needed to train it, and the more difficult it is to interpret the underlying functional dependence of its predictions on the input. In the past, expressivity has often been sacrificed to regain interpretability [Ponte and Melko, 2017; Wetzel and Scherzer, 2017; Zhang *et al.*, 2019a; Greitemann *et al.*, 2019a,b; Liu *et al.*, 2019; Zhang *et al.*, 2020]. Recall from Section 2.6 how previous works restricted the expressive power of the underlying models, for example by reducing the receptive field size of CNNs or by employing explicit regularization, to extract learned order parameters.

Here, we took an alternative approach to deal with the *interpretability-expressivity tradeoff*: By analyzing the class of predictive models that solve the classification and regression tasks underlying SL, LBC, and PBM optimally, we have derived analytical expressions for the indicators of phase transitions of these three methods. This establishes a solid theoretical foundation for SL, LBC, and PBM, based on which we were able to explain and understand the results of a variety of previous studies [Carrasquilla and Melko, 2017; Van Nieuwenburg *et al.*, 2017; Beach *et al.*, 2018; Schäfer and Lörch, 2019; Greplova *et al.*, 2020; Bohrdt *et al.*, 2021]. The analytical expressions not only enable our understanding of the phase-transition-detection methods under consideration – they also allow for the direct computation of their optimal predictions and indicators based on the input data *without* explicitly training NNs. We have demonstrated that this novel procedure can successfully reveal a broad range of different phase transitions in a numerical setting and is favorable in terms of computation time.

3.10 Outlook

We anticipate that similar analyses will be useful to gain an understanding of other NN-based methods for identifying phase transitions [Huembeli *et al.*, 2018, 2019; Kottmann *et al.*, 2020, 2021; Patel *et al.*, 2022; Guo and He, 2023] and other classification tasks in condensed matter physics more broadly [Bohrdt *et al.*, 2019; Zhang *et al.*, 2019b; Pilati and Pieri, 2019; Ghosh *et al.*, 2020; Miles *et al.*, 2021; Szołdra *et al.*, 2021; Gavreev *et al.*, 2022]. In these cases, the optimal models can also serve as benchmark solutions that enable future studies aimed at investigating the learning process of NNs and improving their design and update routines [McClean *et al.*, 2018; Vieijra *et al.*, 2020; Miles *et al.*, 2021; Bukov *et al.*, 2021; Valenti *et al.*, 2022; Miles *et al.*, 2023]. For example, in [Carrasquilla and Melko, 2017; Ch'ng *et al.*, 2017; Broecker *et al.*, 2017] it was shown that an NN trained to predict the phase transition in a given physical model using SL can successfully classify configurations generated from an entirely different Hamiltonian. An exciting prospect is to explore whether the success of this "transfer learning" can be rigorously explained based on our results.

In this chapter, we have restricted ourselves to a single tuning parameter along which a single phase transition occurs. How one appropriately extends SL and LBC to tackle phase diagrams in higher-dimensional parameter spaces remains an open question. The inclusion of multiple tuning parameters also brings up the question of whether the energy is still the only relevant quantity for describing a classical equilibrium system. Can one still utilize this insight to construct accurate empirical distributions and compute optimal indicators from a few samples?

When dealing with quantum systems, we solved the underlying Schrödinger equation exactly, giving us direct access to the probability distributions governing our measurement statistics. However, this prevents us from analyzing systems of larger size. How could one extend our numerical procedure based on evaluating the analytical expressions to utilize approximate descriptions of quantum states, such as neural or tensor networks?

Similarly, could one utilize techniques for density estimation to construct more accurate distributions from a few samples in the classical case as well? Can we quantify how many samples are needed to obtain accurate indicators of phase transitions? In addition, in Section 3.7.5, we have seen that NNs may be beneficial for constructing indicators when few samples are available. When to choose NN-based indicators over indicators constructed from distributions remains to be investigated in full generality, i.e., for a range of systems and available number of samples.

In the quantum case, we were faced with the choice of an appropriate measurement basis. In this chapter, we restricted ourselves to simple projective measurements. Strictly speaking, this choice does require some prior knowledge of the underlying systems which goes against the spirit of these methods in being able to detect novel phase transitions with little to no prior system knowledge.[14] Can we understand the influence of the choice of measurement basis on the results of the ML methods more clearly? And can the methods work with IC-POVMs?

Throughout this chapter, we have encountered several failures of SL, PBM, and LBC in detecting phase transitions. By explicitly constructing optimal indicators we ruled out the possibility that these failures are connected to a lack of model capacity. This result suggests that it may be possible to modify the core objective and indicators of these methods to remedy these failures, i.e., to make these methods work in conjunction with high-capacity predictive models.

Through the analytical expressions for the optimal predictions and indicators it became clear that the three methods are based on measuring changes in the probability distributions underlying the system. Can this statement be made more rigorous, i.e., can we link the indicators of these methods to known statistical distances and other notions quantifying statistical changes?

We will start to tackle these questions in Chapter 4.

The results and figures presented in this chapter have been in parts published in [Arnold and Schäfer, 2022b]. The corresponding code is open source [Arnold and Schäfer, 2022a].

[14] After all, it is simple to construct a mixed state that yields the same measurement statistics as a pure state when performing measurements in a single basis.

Chapter 4

Machine Learning Phase Transitions 2.0: On Generative Models, Higher-Dimensional Parameter Spaces, and Beyond

The results presented in this chapter are based on the following publication:

Mapping out phase diagrams with generative classifiers,
J. Arnold, F. Schäfer, A. Edelman, and C. Bruder,
Phys. Rev. Lett. **132**, 207301 (2024).

4.1 Motivation

A classification task can be approached in two fundamentally distinct ways [Ng and Jordan, 2001]. Typically, a classifier is constructed by modeling the conditional probability $P(y|\boldsymbol{x})$ of label y given a sample $\boldsymbol{x}$ directly [see Figure 4.1(a)]. Models of this type are called *discriminative* and the corresponding classifier predicts the label y with maximal $P(y|\boldsymbol{x})$. Because of the power of neural networks (NNs), particularly convolutional NNs [Krizhevsky *et al.*, 2012], discriminative classifiers have had tremendous success in fields such as image classification. Moreover, by solving classification tasks in a data-driven manner using discriminative classifiers, the phase diagrams of a large variety of physical systems have successfully been revealed in both simulation [Carrasquilla and Melko, 2017; Van Nieuwenburg *et al.*, 2017; Wetzel and Scherzer, 2017; Schäfer and Lörch, 2019; Liu and van Nieuwenburg, 2018; Beach *et al.*, 2018; Suchsland and Wessel, 2018; Lee and Kim, 2019; Kharkov *et al.*, 2020; Greplova *et al.*, 2020; Arnold *et al.*, 2021; Gavreev *et al.*, 2022; Zvyagintseva *et al.*, 2022; Tibaldi *et al.*, 2023; Guo and He, 2023; Guo *et al.*, 2023; Schlömer and Bohrdt, 2023] and experiment [Rem *et al.*, 2019; Bohrdt *et al.*, 2021; Miles *et al.*, 2023] – this is how we have utilized NNs in SL, LBC, and PBM so far in Chapters 2 and 3.[1] Because generic data, such as spin configurations or energy spectra, can be used as input, this constitutes a promising route toward the discovery of novel phases of matter and phase transitions with little to no human supervision.

The alternative is to model the conditional probability of a sample given a label $P(\boldsymbol{x}|y)$ [see Figure 4.1(b)]. Such models are called *generative* models as they describe

[1] Up until now we viewed PBM as a regression task. However, one may equally view it as a classification task with the classes corresponding to the sampled values of the tuning parameter Γ. By weighing each sampled value of the tuning parameter with the corresponding class probability, an estimate for the tuning parameter at which a sample was generated can be obtained, see Section 4.2.3.

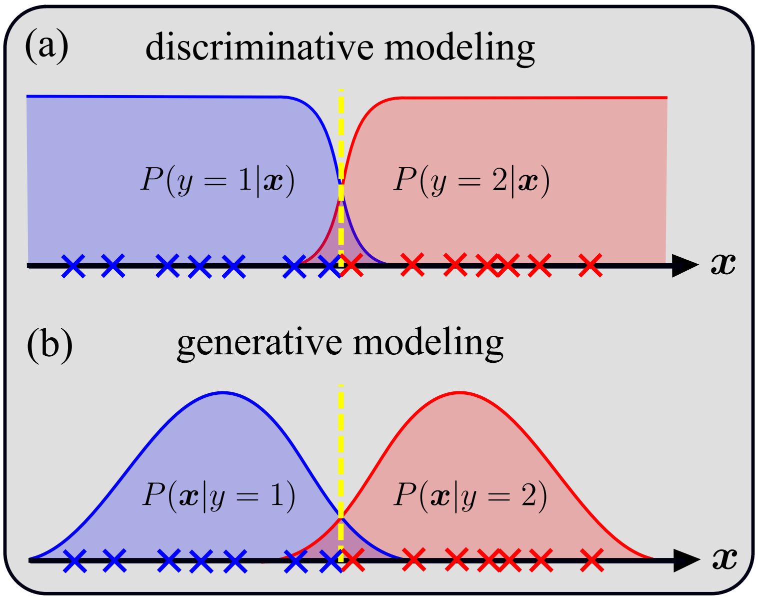

Figure 4.1: Schematic illustration of the probability distributions modeled in (a) the discriminative and (b) generative approach to a binary classification problem with labels $y \in \{1,2\}$. In the discriminative approach, $P(y|\boldsymbol{x})$ is learned directly from data (crosses). In the generative approach, the class-conditional distributions $P(\boldsymbol{x}|y)$ are learned instead.

how to generate samples $\boldsymbol{x}$ conditioned on the class label y.[2] Applying Bayes' rule,

$$P(y|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|y)P(y)}{P(\boldsymbol{x})} = \frac{P(\boldsymbol{x}|y)P(y)}{\sum_{y'} P(\boldsymbol{x}|y')P(y')}, \tag{4.1}$$

one can use a generative model to construct a so-called generative classifier. We have been constructing generative classifiers in Chapter 3 all along, albeit unknowingly.

Generative models have played a pivotal role in recent advances in ML, enabling applications such as the generation of images, composition of music, or translation of text [Vaswani *et al.*, 2017]. Moreover, generative models have been used extensively to describe many-body systems. In statistical physics, Boltzmann distributions have been represented by generative models ranging from mean-field *ansätze* to autoregressive networks [Wu *et al.*, 2019]. Similarly, various approaches have been developed to express quantum states, including mean-field ansätze and tensor networks [Schollwöck, 2011], as well as novel ML-inspired generative models, such as restricted Boltzmann machines [Carleo and Troyer, 2017], recurrent NNs [Hibat-Allah *et al.*, 2020], or transformers [Cha *et al.*, 2021; Melko and Carrasquilla, 2024].

In this chapter, we will formulate the three phase-transition-detection methods in a fully probabilistic fashion, highlighting the key underlying classification tasks. These can be solved using either discriminative or generative classifiers. The procedure based on generative classifiers generalizes the numerical procedure proposed in Chapter 3, which was restricted to generative classifiers constructed from empirical estimates of the underlying distributions and quantum states obtained from exact numerical

[2] Most generally, a generative model is defined as a statistical model of the joint probability distribution $P(\boldsymbol{x}, y)$. Here, we stick with the former definition given that $P(y)$ is chosen beforehand in phase-transition-detection tasks.

solutions of the Schrödinger equation. The first core contribution of this chapter is to showcase how generative models native to the realm of many-body physics can be used to construct classifiers that solve phase-transition-detection tasks more efficiently compared to discriminative classifiers. In a nutshell: In numerical investigations, one often has direct access to a description of the system in terms of generative models. The generative approach can use this information to create a classifier, bypassing an explicit data-driven construction as in the discriminative approach.

Formulating the methods probabilistically also helps us identify prior assumptions that can be lifted to generalize the methods. Here, we modify the methods to more accurately detect phase transitions and extend their application domain to arbitrary-dimensional parameter spaces possibly featuring multiple phases. This constitutes the second core contribution of this chapter. The probabilistic formulation also paves the way for subsequent analyses in Chapter 6.

4.2 Machine learning phase transitions: A probabilistic formulation

In this section, we introduce SL, LBC, and PBM in a fully probabilistic manner. At the core of these methods lay classification tasks. In these tasks, the label $y \in \mathcal{Y}$ specifies a set of points Γ_y in the space of tuning parameters. Without loss of generality, we choose a uniform distribution over the set of parameters associated with each class, i.e., $P(\boldsymbol{\gamma}|y) = 1/|\Gamma_y|$ for $\boldsymbol{\gamma} \in \Gamma_y$ and zero otherwise. We will justify this choice in Section 4.6.1. With this choice, the coarse-grained measurement probability is

$$P(\boldsymbol{x}|y) = \sum_{\boldsymbol{\gamma}\in\Gamma} P(\boldsymbol{x}|\boldsymbol{\gamma})P(\boldsymbol{\gamma}|y) = \frac{1}{|\Gamma_y|}\sum_{\boldsymbol{\gamma}\in\Gamma_y} P(\boldsymbol{x}|\boldsymbol{\gamma}). \tag{4.2}$$

Using Equation (4.1), we have

$$P(y|\boldsymbol{x}) = \frac{\frac{1}{|\Gamma_y|}\sum_{\boldsymbol{\gamma}\in\Gamma_y} P(\boldsymbol{x}|\boldsymbol{\gamma})}{\sum_{y'\in\mathcal{Y}}\frac{1}{|\Gamma_{y'}|}\sum_{\boldsymbol{\gamma}'\in\Gamma_{y'}} P(\boldsymbol{x}|\boldsymbol{\gamma}')}, \tag{4.3}$$

where, *a priori*, we consider each label to be equally likely, i.e., $P(y) = 1/|\mathcal{Y}|$. In [Van Nieuwenburg *et al.*, 2017], i.e., the original formulation of LBC, the prior probability $P(y)$ was not uniform and instead chosen as

$$P(y) = \frac{|\Gamma_y|}{\sum_{y'\in\mathcal{Y}}|\Gamma_{y'}|}. \tag{4.4}$$

In this case, class imbalances are not accounted for and can lead to an unfavorable signal for detecting phase boundaries. We will discuss such an example in detail in Section 4.6.2.

The three methods of SL, LBC, and PBM for mapping out phase diagrams differ in how the parameter space is labeled, i.e., in the choice of $\{\Gamma_y\}_{y\in\mathcal{Y}}$.

4.2.1 Supervised learning

In SL, we assume that we have some prior knowledge of the phase diagram. In particular, assuming we know the number of distinct phases G and their rough location in parameter space, we can choose $\mathcal{Y} = \{0, 1, \ldots, G-1\}$ and Γ_y to be composed of a set of points characteristic of each phase y. Next, we compute the posterior probability $P(y|\boldsymbol{\gamma}) = \mathbb{E}_{\boldsymbol{x}\sim P(\cdot|\boldsymbol{\gamma})}\left[P(y|\boldsymbol{x})\right]$ associated with each phase across a parameter region of interest [see Equation (4.3)]. Rapid changes in the posterior probability are characteristic of phase boundaries. We capture these by the following scalar indicator of phase transitions

$$\begin{aligned} I_{\mathrm{SL}}(\boldsymbol{\gamma}) &= \frac{1}{|\mathcal{Y}|}\sum_{y\in\mathcal{Y}} ||\nabla_{\boldsymbol{\gamma}} P(y|\boldsymbol{\gamma})||_2\,, \\ &= \frac{1}{|\mathcal{Y}|}\sum_{y\in\mathcal{Y}} \left|\left|\mathbb{E}_{\boldsymbol{x}\sim P(\cdot|\boldsymbol{\gamma})}\left[P(y|\boldsymbol{x})\nabla_{\boldsymbol{\gamma}}\ln\Big(P(\boldsymbol{x}|\boldsymbol{\gamma})\Big)\right]\right|\right|_2\,, \end{aligned} \tag{4.5}$$

where $\|\cdot\|_2$ denotes the Euclidean norm and we used the log-derivative trick. This generalizes the SL method as introduced in Chapter 2 and analyzed in Chapter 3. In particular, if the tuning parameter is one-dimensional, there are two distinct phases, and $|\Gamma_0| = |\Gamma_1|$, the indicator in Equation (4.5) reduces to the Bayes-optimal indicator in Equation (3.5), where we chose to replace minus signs by absolute values.

4.2.2 Learning by confusion

SL requires partial knowledge of the phase diagram which may be unavailable. To get around this, [Van Nieuwenburg *et al.*, 2017] have proposed LBC corresponding to a different labeling strategy for one-dimensional parameter spaces that is phase agnostic. Let $\mathcal{Y} = \{0,1\}$ and for each point $\gamma\in\Gamma$ partition the parameter space into two sets $\Gamma_0 = \{\gamma'\in\Gamma|\gamma'\leq\gamma\}$ and $\Gamma_1 = \{\gamma'\in\Gamma|\gamma'>\gamma\}$.[3,4] Each parameter γ defines a bipartition and in turn a classification task. The associated (optimal) average error probability can be computed as

$$p_{\mathrm{err}}(\gamma) = \frac{1}{2}\sum_{y\in\{0,1\}}\frac{1}{|\Gamma_y|}\sum_{\gamma'\in\Gamma_y}\mathbb{E}_{\boldsymbol{x}\sim P(\cdot|\gamma')}\left[p_{\mathrm{err}}(\boldsymbol{x})\right], \tag{4.6}$$

where $p_{\mathrm{err}}(\boldsymbol{x}) = \min\{P(0|\boldsymbol{x}), P(1|\boldsymbol{x})\}$ is the Bayes-optimal average error probability when predicting the label of sample $\boldsymbol{x}$ (see Section 4.5).[5] Intuitively, one expects $p_{\mathrm{err}}(\gamma)$ to be lowest at a phase boundary where the data is partitioned according to the underlying phase structure. Thus, phase boundaries can be detected as local maxima in the indicator $I(\gamma) = 1-2p_{\mathrm{err}}(\gamma)$ which takes on values between 0 and 1 given that a classifier *at worst* achieves $p_{\mathrm{err}} = 1/2$.

This formulation of LBC contains two distinct modifications compared to how it was originally proposed, i.e., compared to how we introduced it in Chapter 2 and analyzed it in Chapter 3. First, we treat each of the two bipartitions equally irrespective of their size in terms of the number of sampled points in parameter space [see Equation (4.6)]. This is crucial for unbiasing the LBC indicator and avoiding trivial peaks at the edges of the sampled interval, i.e., to convert the characteristic V-shaped and W-shaped indicators into a flat signal and a signal with a single peak, respectively. Second, we define the indicator as one minus two times the average error probability instead of the accuracy (i.e., one minus the average error probability). This makes it

[3] In principle, Γ does not need to be sampled on a grid.

[4] Here, we change our convention compared to Chapters 2 and 3 and choose the bipartition point to be the rightmost point of the first region (i.e., region I/0) instead of the central point between the two regions.

[5] Alternatively to $p_{\mathrm{err}}(\boldsymbol{x}) = \min\{P(0|\boldsymbol{x}), P(1|\boldsymbol{x})\}$, we may use $p_{\mathrm{err}}(\boldsymbol{x}, y) = |y - \mathrm{argmax}_{y'} P(y'|\boldsymbol{x})|$ in Equation (4.6). In Section 4.5, we prove that these two choices result in equivalent overall error rates $p_{\mathrm{err}}(\gamma)$.

such that for indistinguishable inputs, i.e., systems with a single phase, we obtain a flat indicator signal at zero, instead of 0.5.

Extension to higher-dimensional parameter spaces

Liu and van Nieuwenburg [2018] extended the LBC method to two-dimensional parameter spaces by considering the predicted phase boundary as a parametrized curve that partitions the parameter space locally and is driven via internal forces (e.g., preventing bending and stretching), as well as external forces aiming to minimize the overall classification error. We find this approach to be unreliable and computationally costly without some partial prior knowledge of the phase diagram, see Appendix E for a detailed discussion. Instead, we propose an alternative robust generalization that applies to parameter spaces of arbitrary dimension building upon the idea of a local bipartition introduced in [Liu and van Nieuwenburg, 2018]:

At each sampled point $\boldsymbol{\gamma}^{\mathrm{bp}} = \left(\gamma_1^{\mathrm{bp}}, \gamma_2^{\mathrm{bp}}, \dots, \gamma_d^{\mathrm{bp}}\right) \in \Gamma$, the parameter space is split along each direction. For a given direction $1 \leq i \leq d$, this yields two sets, $\Gamma_0^{(i)}(\boldsymbol{\gamma}^{\mathrm{bp}})$ and $\Gamma_1^{(i)}(\boldsymbol{\gamma}^{\mathrm{bp}})$, each comprised of the l points closest to $\boldsymbol{\gamma}^{\mathrm{bp}}$ in part I and II of the split parameter space, respectively. All samples drawn at parameter points in $\Gamma_0^{(i)}(\boldsymbol{\gamma}^{\mathrm{bp}})$ and $\Gamma_1^{(i)}(\boldsymbol{\gamma}^{\mathrm{bp}})$ are assigned the label 0 or 1, respectively. Using different norms for measuring this distance results in different local sets. Figure 4.2 illustrates two distinct strategies. In this chapter, we choose the l points $\boldsymbol{\gamma}$ closest to $\boldsymbol{\gamma}^{\mathrm{bp}}$ in each part of the split parameter space with respect to $|\gamma_i - \gamma_i^{\mathrm{bp}}|$, see Figure 4.2(a). This is suitable for parameter spaces sampled on a uniform grid. In the case of non-uniformly sampled parameter spaces, one may instead consider a Euclidean distance $\left|\left|\boldsymbol{\gamma} - \boldsymbol{\gamma}^{\mathrm{bp}}\right|\right|_2$, see Figure 4.2(b). Note that in both cases, sets with a number of points lower than l may arise at the edges of the parameter space. The resulting class imbalance is corrected for, see Section 4.6.2.

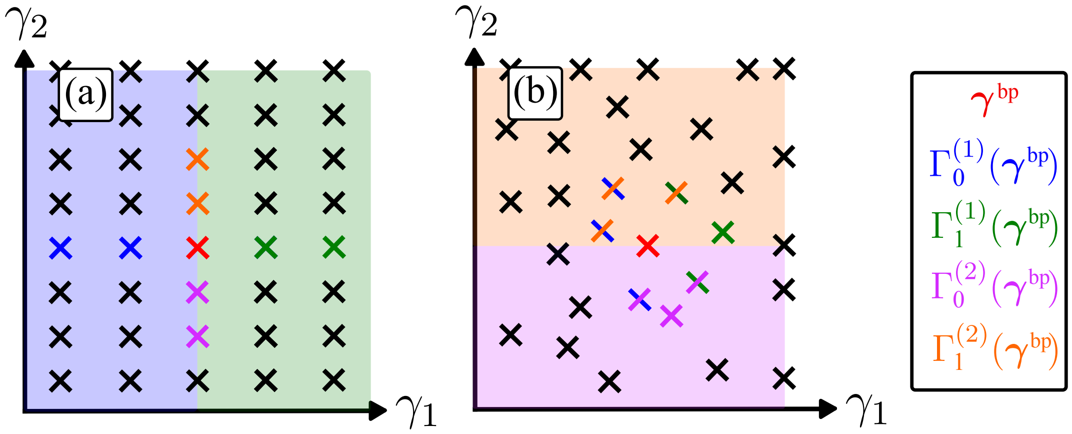
γ_2
(a)
γ_1
γ_2
(b)
γ_1
$\boldsymbol{\gamma}^{\mathrm{bp}}$
$\Gamma_0^{(1)}(\boldsymbol{\gamma}^{\mathrm{bp}})$
$\Gamma_1^{(1)}(\boldsymbol{\gamma}^{\mathrm{bp}})$
$\Gamma_0^{(2)}(\boldsymbol{\gamma}^{\mathrm{bp}})$
$\Gamma_1^{(2)}(\boldsymbol{\gamma}^{\mathrm{bp}})$

FIGURE 4.2: Illustration for constructing local bipartitions for LBC [Equation (4.7)] of a two-dimensional parameter space ($d = 2$) that is sampled (a) uniformly on a grid ($l = 2$) or (b) non-uniformly ($l = 3$). Crosses denote sampled points in parameter space $\boldsymbol{\gamma} \in \Gamma$. Sampled points belonging to one of the four sets $\Gamma_0^{(1)}(\boldsymbol{\gamma}^{\mathrm{bp}})$, $\Gamma_1^{(1)}(\boldsymbol{\gamma}^{\mathrm{bp}})$, $\Gamma_0^{(2)}(\boldsymbol{\gamma}^{\mathrm{bp}})$, $\Gamma_1^{(2)}(\boldsymbol{\gamma}^{\mathrm{bp}})$ are colored blue, green, pink, and orange, respectively. We each show one of the resulting splits of the parameter space by shaded colored regions.

For each sampled point $\boldsymbol{\gamma} \in \Gamma$ and the resulting two sets $\Gamma_0^{(i)}(\boldsymbol{\gamma})$ and $\Gamma_1^{(i)}(\boldsymbol{\gamma})$, we then compute an indicator component $I_{\mathrm{LBC}}^{(i)}(\boldsymbol{\gamma}) = 1 - 2p_{\mathrm{err}}^{(i)}(\boldsymbol{\gamma})$ according to Equation (4.6). The overall LBC indicator is then given as

$$I_{\mathrm{LBC}}(\boldsymbol{\gamma}) = \sqrt{\sum_{i=1}^{d} \left[I_{\mathrm{LBC}}^{(i)}(\boldsymbol{\gamma})\right]^2}. \tag{4.7}$$

Together with the spacing between sampled parameter points, the hyperparameter l sets the natural length scale on which changes in the distributions underlying the system's measurement statistics are detected. In this chapter, we typically set $l = 1$, i.e., we aim to detect local changes. In Chapter 6, we will justify this choice rigorously by connecting the resulting indicator to a well-known information-theoretic quantity. When the underlying data is noisy or few samples are available, we find $l = \mathcal{O}(1) > 1$ to produce better results compared to $l = 1$ as it averages over small-scale fluctuations. While this was not necessary in this chapter, we will utilize this setting in Chapter 8. In Chapter 9, we will detect more large-scale changes by setting l to be $\mathcal{O}(10)$–$\mathcal{O}(100)$.

Note that if the underlying parameter space is one-dimensional and $l > |\Gamma|$, the resulting sets Γ_0 and Γ_1 match the ones introduced in the original LBC scheme in Chapters 2-3: the entire range is split into two distinct sets. Throughout the rest of this thesis, we will often set $l = \infty$ to denote this case.

4.2.3 Prediction-based method

While LBC does not require partial knowledge of the phase diagram, it requires solving a multitude of classification problems – one for each bipartition. Schäfer and Lörch [2019] addressed this by an alternative phase-agnostic labeling strategy in PBM, where each sampled value of the tuning parameter can be considered its own class $\Gamma_y = \{\boldsymbol{\gamma}_y\}$, $y \in \mathcal{Y} = \{0, 1, \dots, |\Gamma| - 1\}$. The mean predicted value of the tuning parameter

$$\hat{\boldsymbol{\gamma}}(\boldsymbol{\gamma}) = \mathbb{E}_{\boldsymbol{x}\sim P(\cdot|\boldsymbol{\gamma})}\left[\hat{\boldsymbol{\gamma}}(\boldsymbol{x})\right] = \mathbb{E}_{\boldsymbol{x}\sim P(\cdot|\boldsymbol{\gamma})}\left[\sum_{y\in\mathcal{Y}} P(y|\boldsymbol{x})\boldsymbol{\gamma}_y\right] \tag{4.8}$$

is expected to be most sensitive at phase boundaries. Here, $\hat{\boldsymbol{\gamma}}(\boldsymbol{x})$ corresponds to the optimal prediction of the tuning parameter at which the sample $\boldsymbol{x}$ has been generated. We capture the susceptibility of the mean predicted values by the following indicator

$$\begin{aligned} I_{\mathrm{PBM}}(\boldsymbol{\gamma}) &= \sqrt{\sum_{i=1}^{d} \left(\frac{\partial \hat{\gamma}_i(\boldsymbol{\gamma})/\partial \gamma_i}{\mathrm{std}_i(\boldsymbol{\gamma})}\right)^2}, \\ &= \left|\left|\mathbb{E}_{\boldsymbol{x}\sim P(\cdot|\boldsymbol{\gamma})}\left[\frac{\hat{\boldsymbol{\gamma}}(\boldsymbol{x})}{\mathbf{std}(\boldsymbol{\gamma})}\nabla_{\boldsymbol{\gamma}} \ln\Big(P(\boldsymbol{x}|\boldsymbol{\gamma})\Big)\right]\right|\right|_2, \end{aligned} \tag{4.9}$$

where $\mathbf{std}(\boldsymbol{\gamma}) = \sqrt{\mathbb{E}_{\boldsymbol{x}\sim P(\cdot|\boldsymbol{\gamma})}\left[\hat{\boldsymbol{\gamma}}(\boldsymbol{x})^2\right] - \left(\mathbb{E}_{\boldsymbol{x}\sim P(\cdot|\boldsymbol{\gamma})}\left[\hat{\boldsymbol{\gamma}}(\boldsymbol{x})\right]\right)^2}$ is the associated standard deviation. Here, operations are carried out elementwise and we used the log-derivative trick.

Compared to the original formulation of PBM in [Schäfer and Lörch, 2019] and our introductions in Chapters 2 and 3, the main modification is the division of the signal $\partial\hat{\gamma}_i/\partial\gamma_i$ by the standard deviation of $\hat{\gamma}_i$, denoted by std_i. This is found to yield more reliable predicted phase diagrams and alleviate problems encountered in previous studies [Schäfer and Lörch, 2019; Arnold and Schäfer, 2022b], including problems we encountered in Chapter 3. We will discuss this in more detail in Section 4.6.3. Besides this major change, we introduced norms to ensure a positive indicator value and removed constant offsets. Previously, the PBM indicator was defined via the divergence of the field given by the predictions $\hat{\boldsymbol{\gamma}}(\boldsymbol{\gamma})$ and took on a characteristic value of $-d$ deep within a phase in a d-dimensional parameter space [Arnold *et al.*, 2021]. The modified PBM indicator now takes on a value of zero deep within a phase.

4.3 Discriminative vs. generative modeling

By casting the determination of a phase diagram, i.e., the detection of the system's phase boundaries, as classification tasks, we have reduced the problem to the computation of a scalar indicator of phase transitions $I(\boldsymbol{\gamma})$ across the region of interest [see Equations (4.5), (4.7), and (4.45)]. Up to now, this computation has typically been approached in a discriminative way:

Given a set of samples $\mathcal{D}_{\boldsymbol{\gamma}}$ drawn from $P(\cdot|\boldsymbol{\gamma})$ for each $\boldsymbol{\gamma} \in \Gamma_y$ and $y \in \mathcal{Y}$, a (parametric) model $\tilde{P}(y|\boldsymbol{x})$ of $P(y|\boldsymbol{x})$ is constructed. Typically, $\tilde{P}(y|\boldsymbol{x})$ is represented as an NN whose parameters are optimized in a supervised fashion to solve the respective classification task. We will discuss this discriminative approach in full detail in Section 4.5. An estimate of the indicator can be computed by substituting $P(y|\boldsymbol{x})$ with $\tilde{P}(y|\boldsymbol{x})$ and replacing expected values with a sample mean $\mathbb{E}_{\boldsymbol{x}\sim P(\cdot|\boldsymbol{\gamma})} \rightarrow \frac{1}{|\mathcal{D}_{\boldsymbol{\gamma}}|}\sum_{\boldsymbol{x}\in\mathcal{D}_{\boldsymbol{\gamma}}}$. The main idea we want to convey in this chapter is how to approach the problem of detecting phase transitions in a generative manner:

Given a model of the probability distributions underlying the measurement statistics at various discrete points in parameter space $\{\tilde{P}(\cdot|\boldsymbol{\gamma})\}_{\boldsymbol{\gamma}\in\Gamma}$ from which one can efficiently sample, an estimate of an indicator [Equations (4.5), (4.7), and (4.45)] can be computed by substituting $P(\cdot|\boldsymbol{\gamma})$ with $\tilde{P}(\cdot|\boldsymbol{\gamma})$ and replacing expected values with a sample mean $\mathbb{E}_{\boldsymbol{x}\sim P(\cdot|\boldsymbol{\gamma})} \rightarrow \frac{1}{|\tilde{\mathcal{D}}_{\boldsymbol{\gamma}}|}\sum_{\boldsymbol{x}\in\tilde{\mathcal{D}}_{\boldsymbol{\gamma}}}$, where $\tilde{\mathcal{D}}_{\boldsymbol{\gamma}}$ denotes the set of samples drawn from the model $\tilde{P}(\cdot|\boldsymbol{\gamma})$. In ML terms, one desires generative models with explicit, tractable densities [Goodfellow, 2016]. Popular examples belonging to this class are autoregressive networks, such as fully visible belief networks or recurrent NNs, normalizing flows, or tensor networks. However, one can also consider nonparametric models, e.g., based on histogram binning, as well as numerically exact (or even analytical) probability distributions if available.

Note that $\tilde{P}(\cdot|\boldsymbol{\gamma})$ models the measurement statistics underlying the physical system. Because the distribution underlying the measurement statistics of a physical system contains strictly more information than a corresponding classifier, $\tilde{P}(\cdot|\boldsymbol{\gamma})$ is more fundamental compared to $\tilde{P}(y|\boldsymbol{x})$. In particular, you can use $\tilde{P}(\boldsymbol{x}|\boldsymbol{\gamma})$ to construct a model $\tilde{P}(y|\boldsymbol{x})$, but the reverse is not true. More generally, a model $\tilde{P}(\boldsymbol{x}|\boldsymbol{\gamma})$ can enable various downstream tasks, such as the computation of distinct indicators of phase transitions or physical observables. In numerical investigations, one often has direct access to a description of the system in terms of a generative model that acts as the source of data. The generative approach to classification can use this information to directly construct a classifier instead of learning it iteratively from data, which is necessary in the discriminative approach. In particular, if $\tilde{P}(\cdot|\boldsymbol{\gamma}) = P(\cdot|\boldsymbol{\gamma})$ the generative classifier is Bayes optimal [Devroye *et al.*, 1996; Goodfellow *et al.*, 2016] *by construction*, meaning no other classifier can perform better (on average) on the classification task at hand. In contrast, the discriminative approach yields a Bayes-optimal classifier in the limit of infinite dataset size and model capacity [Goodfellow *et al.*, 2016] (e.g., utilizing sufficiently large, well-trained NNs), see proof in Section 4.5. In practice, this results in a large computational overhead compared to the generative approach. We will demonstrate this explicitly in Section 4.5.1.

4.4 Application of generative approach to physical systems

In this section, we show how to map out phase diagrams of classical equilibrium systems and quantum ground states in a generative manner using the modified indicators defined in Section 4.2.

4.4.1 Classical equilibrium systems

For a classical system at equilibrium with a large thermal reservoir, the probability to find the system in state $\boldsymbol{x} \in \mathcal{X}$ is given by

$$P(\boldsymbol{x}|\boldsymbol{\gamma}) = e^{-\mathcal{H}(\boldsymbol{x},\boldsymbol{\gamma})}/Z(\boldsymbol{\gamma}), \tag{4.10}$$

where $Z(\boldsymbol{\gamma})$ is the partition function. Here, we consider dimensionless Hamiltonians of the form

$$\mathcal{H} = H/k_{\mathrm{B}}T = \sum_{i=1}^{d} \gamma_i X_i(\boldsymbol{x}). \tag{4.11}$$

As the state space $\mathcal{X}$ is large, modeling these distributions is a hard task. However, such Boltzmann distributions belong to the exponential family and

$$\boldsymbol{X} = \big(X_1(\boldsymbol{x}), \dots, X_d(\boldsymbol{x})\big) \tag{4.12}$$

is a minimal sufficient statistic for $\boldsymbol{\gamma}$, i.e., the map $\boldsymbol{x} \mapsto \boldsymbol{X}$ corresponds to an optimal lossless compression with respect to $\boldsymbol{\gamma}$.[6] Thus, to map out phase diagrams by computing an indicator it suffices to model the distributions over the sufficient statistic $\{P(\boldsymbol{X}|\boldsymbol{\gamma})\}_{\boldsymbol{\gamma}\in\Gamma}$, see proof below. Crucially, the dimensionality of the minimal sufficient statistic $\boldsymbol{X}$, $\dim(\boldsymbol{X}) = d$, can be independent of the system size. This enables simple nonparametric modeling approaches that are asymptotically unbiased and fast to evaluate.

Let us prove that to map out phase diagrams of a large class of classical equilibrium systems by computing one of the three indicators discussed in Section 4.2, it suffices to model $\{P(\boldsymbol{X}|\boldsymbol{\gamma})\}_{\boldsymbol{\gamma}\in\Gamma}$, where $\boldsymbol{X}$ is the corresponding sufficient statistic.

Proof

Consider probability distributions of the exponential family

$$P_{\exp}(\boldsymbol{x}|\boldsymbol{\gamma}) = h(\boldsymbol{x}) \exp\left[\sum_{i=1}^{d} \eta_i(\boldsymbol{\gamma}) X_i(\boldsymbol{x}) - A(\boldsymbol{\gamma})\right], \tag{4.13}$$

where $h(\boldsymbol{x}) \geq 0$ is the carrier measure, $\boldsymbol{X}(\boldsymbol{x})$ is a sufficient statistic, $\boldsymbol{\eta}(\boldsymbol{\gamma})$ are the natural parameters, and $A(\boldsymbol{\gamma})$ is the log-partition function. The statistic $\boldsymbol{X}(\boldsymbol{x})$

[6] While any sufficient statistic would correspond to a lossless compression, a minimal sufficient statistic corresponds to a lossless compression that results in the lowest dimension. Any further compression would lead to an information loss.

is a *minimal* sufficient statistic for $\boldsymbol{\gamma}$ if the set of allowed natural parameters $\boldsymbol{\eta}(\boldsymbol{\gamma})$ spans a d-dimensional space [Casella and Berger, 2002]. The distribution over the sufficient statistic is given by

$$\begin{aligned} P_{\exp}(\boldsymbol{X}'|\gamma) &= \sum_{\boldsymbol{x}\in\mathcal{X}} P_{\exp}(\boldsymbol{x}|\gamma)\delta\left[\boldsymbol{X}(\boldsymbol{x}) - \boldsymbol{X}'\right] \\ &= g(\boldsymbol{X}')\exp\left[\boldsymbol{\eta}(\gamma)\cdot\boldsymbol{X}' - A(\gamma)\right], \end{aligned} \tag{4.14}$$

where $g(\boldsymbol{X}') = \sum_{\boldsymbol{x}\in\mathcal{X}} h(\boldsymbol{x})\delta\left[\boldsymbol{X}(\boldsymbol{x}) - \boldsymbol{X}'\right]$ and δ denotes the Kronecker delta. We have

$$P_{\exp}(\boldsymbol{x}|\gamma)/P_{\exp}(\boldsymbol{X}(\boldsymbol{x})|\gamma) = h(\boldsymbol{x})/g(\boldsymbol{X}(\boldsymbol{x})). \tag{4.15}$$

Crucially, this ratio is independent of $\boldsymbol{\gamma}$. Hence, using Equation (4.15), the conditional probability of a label in classification tasks arising when detecting phase transitions [Equation (4.3)] can be written as

$$\begin{aligned} P(y|\boldsymbol{x}) &= \frac{\frac{1}{|\Gamma_y|}\sum_{\gamma\in\Gamma_y} P_{\exp}(\boldsymbol{x}|\gamma)}{\sum_{y'\in\mathcal{Y}} \frac{1}{|\Gamma_{y'}|}\sum_{\gamma'\in\Gamma_{y'}} P_{\exp}(\boldsymbol{x}|\gamma')} \\ &= \frac{\frac{1}{|\Gamma_y|}\sum_{\gamma\in\Gamma_y} P_{\exp}(\boldsymbol{X}(\boldsymbol{x})|\gamma)}{\sum_{y'\in\mathcal{Y}} \frac{1}{|\Gamma_{y'}|}\sum_{\gamma'\in\Gamma_{y'}} P_{\exp}(\boldsymbol{X}(\boldsymbol{x})|\gamma')} \\ &= P(y|\boldsymbol{X}(\boldsymbol{x})). \end{aligned} \tag{4.16}$$

Thus,

$$\begin{aligned} P(y|\gamma) &= \mathbb{E}_{\boldsymbol{x}\sim P_{\exp}(\cdot|\gamma)}\left[P(y|\boldsymbol{x})\right] \\ &= \sum_{\boldsymbol{x}\in\mathcal{X}} P_{\exp}(\boldsymbol{x}|\gamma)P(y|\boldsymbol{x}) \\ &= \sum_{\boldsymbol{x}\in\mathcal{X}} P_{\exp}(\boldsymbol{X}(\boldsymbol{x})|\gamma)P(y|\boldsymbol{X}(\boldsymbol{x}))h(\boldsymbol{x})/g(\boldsymbol{X}(\boldsymbol{x})) \\ &= \sum_{\boldsymbol{X}'\in\mathcal{X}_{\text{suff}}}\sum_{\boldsymbol{x}\in\mathcal{X}} P_{\exp}(\boldsymbol{X}'|\gamma)P(y|\boldsymbol{X}')\delta\left[\boldsymbol{X}(\boldsymbol{x}) - \boldsymbol{X}'\right]h(\boldsymbol{x})/g(\boldsymbol{X}') \\ &= \sum_{\boldsymbol{X}\in\mathcal{X}_{\text{suff}}} P_{\exp}(\boldsymbol{X}|\gamma)P(y|\boldsymbol{X}) \\ &= \mathbb{E}_{\boldsymbol{X}\sim P_{\exp}(\cdot|\gamma)}\left[P(y|\boldsymbol{X})\right], \end{aligned} \tag{4.17}$$

where $\mathcal{X}_{\text{suff}}$ is the state space (without duplicates) associated with the sufficient statistic $\mathcal{X}_{\text{suff}} = \overline{\{\boldsymbol{X}(\boldsymbol{x})|\boldsymbol{x}\in\mathcal{X}\}}$. Similarly, using Equation (4.16), we have

$$\begin{aligned} \hat{\gamma}(\boldsymbol{x}) &= \sum_{y\in\mathcal{Y}} P(y|\boldsymbol{x})\gamma_y = \sum_{y\in\mathcal{Y}} P(y|\boldsymbol{X}(\boldsymbol{x}))\gamma_y \\ &= \hat{\gamma}\left(\boldsymbol{X}(\boldsymbol{x})\right), \end{aligned} \tag{4.18}$$

and

$$\begin{aligned}\hat{\gamma}(\boldsymbol{\gamma}) &= \mathbb{E}_{\boldsymbol{x}\sim P_{\text{exp}}(\cdot|\boldsymbol{\gamma})}\left[\hat{\gamma}(\boldsymbol{x})\right] \\ &= \sum_{\boldsymbol{x}\in\mathcal{X}} P_{\text{exp}}(\boldsymbol{x}|\boldsymbol{\gamma})\hat{\gamma}(\boldsymbol{x}) \\ &= \sum_{\boldsymbol{X}\in\mathcal{X}_{\text{suff}}} P_{\text{exp}}(\boldsymbol{X}|\boldsymbol{\gamma})\hat{\gamma}(\boldsymbol{X}) \\ &= \mathbb{E}_{\boldsymbol{X}\sim P_{\text{exp}}(\cdot|\boldsymbol{\gamma})}\left[\hat{\gamma}(\boldsymbol{X})\right]. \end{aligned} \tag{4.19}$$

The relevant indicators [Equations (4.5), (4.7), and (4.45)] of the three phase-transition-detection methods presented in Section 4.2 can be straightforwardly computed based on the quantities in Equations (4.17), and (4.19), which are expressed solely in terms of the distribution over the sufficient statistic $P(\boldsymbol{X}|\boldsymbol{\gamma})$ (instead of the full distribution over $\boldsymbol{x}$).

In Section 4.4.1, we consider classical systems with dimensionless Hamiltonians of the form given in Equation (4.11) at equilibrium with a large thermal reservoir. In this case, the probability of finding the system in state $\boldsymbol{x}\in\mathcal{X}$ is given by Equation (4.10). Such Boltzmann distributions belong to the exponential family [Equation (4.13)] with $h(\boldsymbol{x})=1$, $A(\boldsymbol{\gamma})=\ln Z(\boldsymbol{\gamma})$, and $\eta_i(\boldsymbol{\gamma})=\gamma_i$. Thus, as a special case of the above, it follows that phase diagrams of such systems can be mapped out given distributions over the sufficient statistic $\{P(\boldsymbol{X}|\boldsymbol{\gamma})\}_{\boldsymbol{\gamma}\in\Gamma}$.

Utilizing knowledge of symmetries

The sufficient statistic allows for an optimal lossless compression of the state space $\mathcal{X}$. However, we have seen that it requires knowledge of the underlying Hamiltonian. In the following, we demonstrate how one can achieve a lossless compression of the state space without explicitly knowing the underlying Hamiltonian, but only utilizing knowledge of its symmetries. Let us consider a symmetry operation $S:\mathcal{X}\rightarrow\mathcal{X}$ that leaves the energy of the system invariant, i.e., for any $\boldsymbol{x}\in\mathcal{X}$, we have $\mathcal{H}\left(S(\boldsymbol{x})\right)=\mathcal{H}(\boldsymbol{x})$. Considering dimensionless Hamiltonians of the form given in Equation (4.11), we have

$$\sum_{i=1}^{d}\gamma_i\left[X_i(\boldsymbol{x})-X_i\left(S(\boldsymbol{x})\right)\right]=0 \tag{4.20}$$

for all allowed parameters $\boldsymbol{\gamma}$. Assuming that this set spans a d-dimensional space, we have $\boldsymbol{X}(\boldsymbol{x})=\boldsymbol{X}\left(S(\boldsymbol{x})\right)$ for any $\boldsymbol{x}\in\mathcal{X}$. That is, the sufficient statistic is also invariant under any symmetry operation S. Thus, we can perform a lossless compression by adopting a representation that is unique for all samples related by symmetry operations. Let us denote the associated state space by $\mathcal{X}_{\text{symm}}$. Note that while this compression is lossless, it is not necessarily optimal, i.e., $|\mathcal{X}_{\text{symm}}|\geq|\mathcal{X}_{\text{suff}}|$.

Example: Anisotropic Ising Model

As a concrete example, we study the prototypical anisotropic Ising model on an $L\times L$ square lattice described by the Hamiltonian

$$H=-\sum_{j,k=1}^{L}\left(J_x\sigma_{j,k}\sigma_{j,k+1}+J_y\sigma_{j+1,k}\sigma_{j,k}\right), \tag{4.21}$$

where $\sigma_{j,k} \in \{\pm 1\}$ denote Ising spins and J_x and J_y are the coupling strengths in the horizontal and vertical direction, respectively. At low temperatures, there exist four ordered phases (related by symmetry) each of which undergoes a second-order phase transition to a paramagnetic phase as the temperature is increased [Figure 4.3(a)]. The two tuning parameters are $\boldsymbol{\gamma} = (J_x/k_{\mathrm{B}}T, J_y/k_{\mathrm{B}}T)$ and for a given spin configuration $\boldsymbol{\sigma}$ the minimal sufficient statistic is

$$\boldsymbol{X}(\boldsymbol{\sigma}) = \left(- \sum_{j,k=1}^{L} \sigma_{j,k}\sigma_{j,k+1} \, , \, - \sum_{j,k=1}^{L} \sigma_{j+1,k}\sigma_{j,k} \right) , \tag{4.22}$$

which corresponds to the nearest-neighbor correlation in x- and y-direction, respectively.

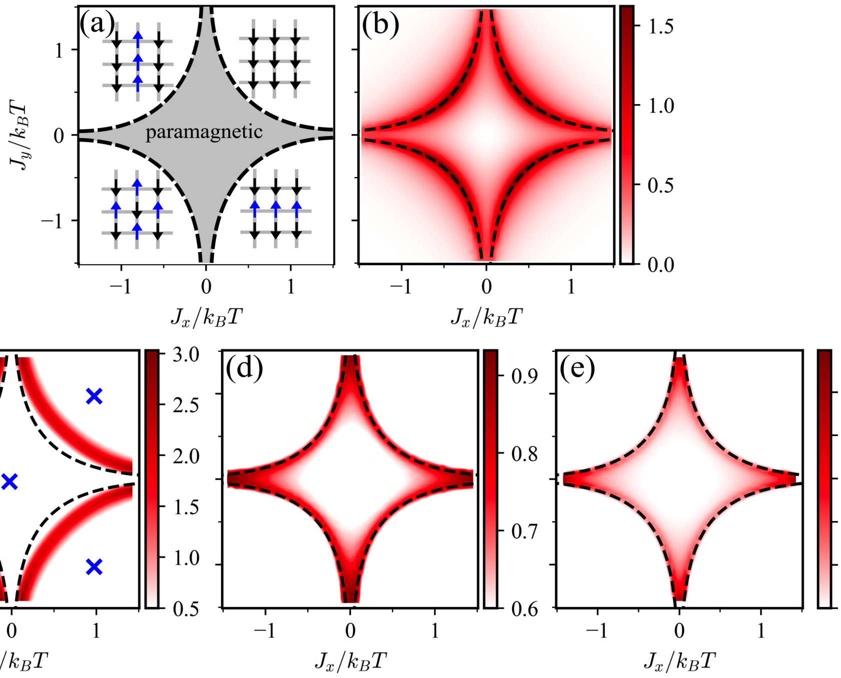

FIGURE 4.3: Results for the anisotropic Ising model on a square lattice [Equation (4.21), $L = 20$]. (a) Schematic illustration of the phase diagram with the characteristic spin configurations of each of the four ordered phases. (b) Heat capacity per spin $C(\boldsymbol{\gamma})/Nk_{\mathrm{B}} = \left(\langle E^2\rangle_{\boldsymbol{\gamma}} - \langle E\rangle_{\boldsymbol{\gamma}}^2\right)/Nk_{\mathrm{B}}^2T^2$, where $N = L^2$ is the total number of spins. (c) $I_{\mathrm{SL}}(\boldsymbol{\gamma})$ [Equation (4.5)] where the set of points $\{\Gamma_y\}_{y\in\mathcal{Y}}$ representative of each phase is marked by blue crosses ($\mathcal{Y} = \{0,1,2,3,4\}$). (d) $I_{\mathrm{LBC}}(\boldsymbol{\gamma})$ [Equation (4.7)] with $l = 1$. (e) $I_{\mathrm{PBM}}(\boldsymbol{\gamma})$ [Equation (4.45)] where $\hat{\boldsymbol{\gamma}}(\boldsymbol{x})$ is estimated elementwise from line scans. The set Γ is composed of a uniform grid with 60 points for each axis and $|\mathcal{D}_{\boldsymbol{\gamma}}| = 10^5$ for all $\boldsymbol{\gamma} \in \Gamma$. Onsager's analytical solution [Onsager, 1944] for the phase boundary is shown as a black dashed line ($J_y/k_{\mathrm{B}}T = -\ln[\tanh(J_x/k_{\mathrm{B}}T)]/2$ for $J_x, J_y > 0$; similarly for the other three sectors).

We draw spin configurations $\boldsymbol{\sigma}$ from Boltzmann distributions $\{P(\cdot|\boldsymbol{\gamma})\}_{\boldsymbol{\gamma}\in\Gamma}$ via Markov chain Monte Carlo. In particular, given a set of parameters $\boldsymbol{\gamma}$, we use the Metropolis-Hastings algorithm to sample spin configurations from the corresponding Boltzmann distribution. The lattice is updated by drawing a random spin, which is flipped with probability $\min\{1, e^{-\Delta E/k_{\mathrm{B}}T}\}$, where ΔE is the energy difference resulting from the considered flip. After a thermalization period of 10^5 lattice sweeps, we collect 10^5 samples. We treat each quadrant of the phase diagram separately. In each quadrant, we initialize the system in one of the two corresponding ground states and increase γ_1 at constant γ_2. When increasing γ_1, we use the spin configuration from the preceding value as an initial condition for the Markov chain. When increasing γ_2, we reset the system to the corresponding ground state.

Based on the sampled spin configurations we compute the sufficient statistic and construct empirical distributions $\{\tilde{P}(\boldsymbol{X}|\boldsymbol{\gamma})\}_{\boldsymbol{\gamma}\in\Gamma}$ using histogram binning $\tilde{P}(\boldsymbol{X}'|\boldsymbol{\gamma}) = 1/|\mathcal{D}_{\boldsymbol{\gamma}}| \sum_{\boldsymbol{\sigma}\in\mathcal{D}_{\boldsymbol{\gamma}}} \delta_{\boldsymbol{X}(\boldsymbol{\sigma}),\boldsymbol{X}'}$. Based on these models, we compute the three indicators of phase transitions [Equations (4.5), (4.7), and (4.45)] as described in Section 4.3, see Figures 4.3(c)-(e). The newly proposed indicators I_{LBC} and I_{PBM} reproduce the known phase diagram quantitatively and are consistent with physics-informed quantities, such as the heat capacity [cf. Figure 4.3(b)], which is remarkable given that these indicators are generic in nature and do not require prior knowledge of the phase diagram. The simple indicator I_{SL} only reproduces the phase diagram qualitatively and the result strongly depends on the choice of $\{\Gamma_y\}_{y\in\mathcal{Y}}$, i.e., on the amount of prior knowledge of the phase diagram being utilized.

4.4.2 Quantum ground states

For a quantum system subjected to a measurement described by a positive operator-valued measure (POVM), the probability to obtain the measurement outcome $\boldsymbol{x} \in \mathcal{X}$ associated with the POVM element $\Pi_{\boldsymbol{x}}$ is given by $P(\boldsymbol{x}|\boldsymbol{\gamma}) = \mathrm{tr}\,[\Pi_{\boldsymbol{x}}\rho(\boldsymbol{\gamma})]$. Modeling such distributions is a hard task due to the exponential growth of the underlying Hilbert space. However, numerous ansätze, ranging from mean-field to tensor networks [Schollwöck, 2011], as well as ML-inspired architectures based on autoregressive NNs [Sharir *et al.*, 2020; Hibat-Allah *et al.*, 2020] have been developed to approximate ground states on the basis of the variational principle.

Example: Cluster-Ising Model

As an example for utilizing generative classifiers in this context, we consider a spin$-\frac{1}{2}$ chain of length L (odd) governed by a cluster-Ising Hamiltonian [Smacchia *et al.*, 2011; Verresen *et al.*, 2017]

$$H = -\sum_{i=1}^{L} \left(JZ_{i-1}X_iZ_{i+1} + h_1X_i + h_2X_iX_{i+1}\right), \tag{4.23}$$

where $\{X_i, Y_i, Z_i\}$ are the Pauli operators acting on the spin at site i and we consider open boundary conditions defined by $Z_0 = Z_{L+1} = X_{L+1} = \mathbb{I}$. Here, the tuning parameters $\boldsymbol{\gamma} = (h_1/J, h_2/J)$ correspond to the external field strength and nearest-neighbor Ising-type coupling, respectively. The ground-state phase diagram of this model features three distinct phases [Figure 4.4(a)]: At finite h_1/J, for $h_2/J \to \infty$ the system is in a paramagnetic phase with all spins pointing in the x-direction, whereas for $h_2/J \to -\infty$ it is in a Néel-type antiferromagnetic phase. At $h_1/J = h_2/J = 0$, the ground state is a cluster state [Briegel and Raussendorf, 2001] giving rise to a symmetry-protected topological (SPT) quantum phase ($\mathbb{Z}_2 \times \mathbb{Z}_2$ symmetry) characterized by a nonzero expectation value of the string order parameter $\mathcal{S} = Z_1X_2X_4\cdots X_{L-3}X_{L-1}Z_L$. The cluster-Ising model is a prime candidate for studying topological order with quantum computers [Azses *et al.*, 2020; Herrmann et al., 2022; Smith et al., 2022; Zapletal et al., 2024] and has recently been investigated using both classical [Sadoune et al., 2023] and quantum discriminative classifiers [Herrmann et al., 2022; Zapletal et al., 2024].

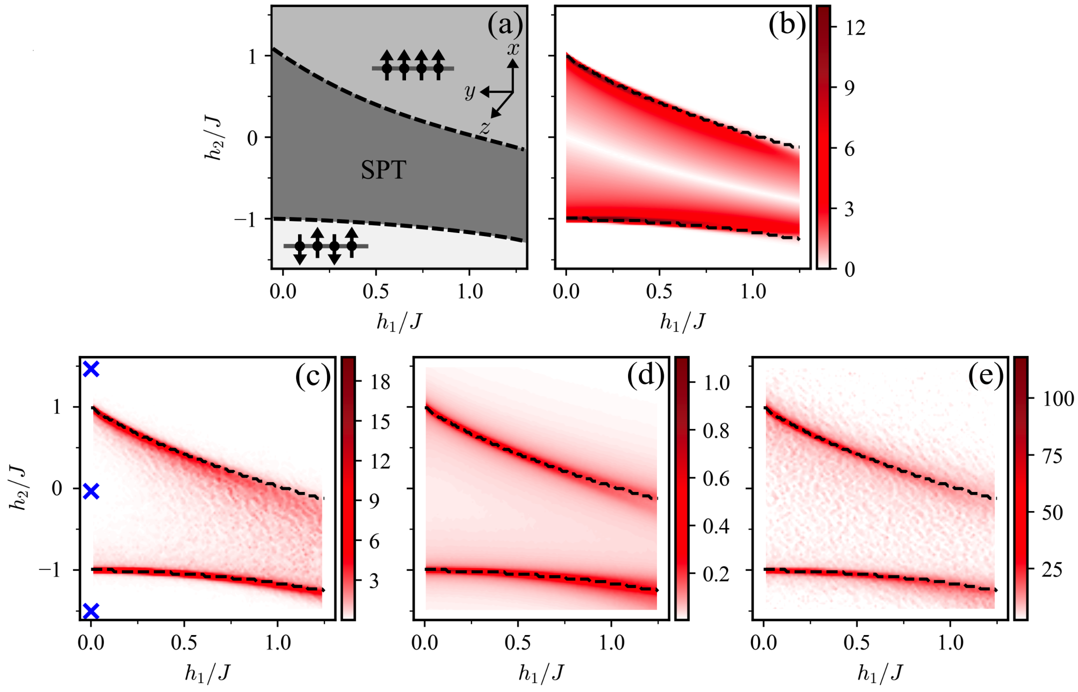

FIGURE 4.4: Results for the cluster-Ising model [Equation (4.23), $L = 71$]. (a) Schematic illustration of the phase diagram featuring three distinct phases: a Néel-type antiferromagnetic phase (bottom), an SPT phase (middle), and a paramagnetic phase (top). (b) Magnitude of the derivative of the string order parameter $|\partial\langle\mathcal{S}\rangle_{\boldsymbol{\gamma}}/\partial\gamma_2|$. (c) $I_{\mathrm{SL}}(\boldsymbol{\gamma})$ [Equation (4.5)] where the sets of points $\{\Gamma_y\}_{y\in\mathcal{Y}}$ representative of each phase are marked by blue crosses ($\mathcal{Y} = \{0, 1, 2\}$). (d) $I_{\mathrm{LBC}}(\boldsymbol{\gamma})$ [Equation (4.7)] with $l = 1$. (e) $I_{\mathrm{PBM}}(\boldsymbol{\gamma})$ [Equation (4.45)] where $\hat{\boldsymbol{\gamma}}(\boldsymbol{x})$ is estimated elementwise from line scans. The set Γ is composed of a uniform grid with 101 points for each axis and $|\tilde{\mathcal{D}}_{\boldsymbol{\gamma}}| = 10^3$ for all $\boldsymbol{\gamma} \in \Gamma$. Estimated phase boundaries (black dashed lines) are determined from maxima in $|\partial^2\langle H\rangle_{\boldsymbol{\gamma}}/\partial\gamma_2^2|$.

To avoid any information loss and for the sake of generality, we consider informationally complete POVMs, i.e., measurements whose statistics completely specify the quantum state at hand. For a single qubit, the Pauli-6 POVM is a simple and common choice. It consists of the six POVM elements

$$\left\{\frac{1}{3}|0\rangle\langle 0|, \frac{1}{3}|1\rangle\langle 1|, \frac{1}{3}|+\rangle\langle +|, \frac{1}{3}|-\rangle\langle -|, \frac{1}{3}|+i\rangle\langle +i|, \frac{1}{3}|-i\rangle\langle -i|\right\}, \tag{4.24}$$

where $\{|0\rangle, |1\rangle\}$, $\{|+\rangle, |-\rangle\}$, and $\{|+i\rangle, |-i\rangle\}$ denote the eigenstates of the Pauli operators Z, X, and Y, respectively. A POVM for the entire many-qubit Hilbert space can be constructed using tensor products $\Pi_{\boldsymbol{x}} = \Pi^{(1)}_{x_1} \otimes \Pi^{(2)}_{x_2} \otimes \cdots \otimes \Pi^{(L)}_{x_L}$, leading to $|\mathcal{X}| = 6^L$ possible measurement outcomes. We construct models for the measurement statistic $\{\tilde{P}(\cdot|\boldsymbol{\gamma})\}_{\boldsymbol{\gamma}\in\Gamma}$ using a matrix product state ansatz that is optimized via the finite-size density matrix renormalization group (DMRG) algorithm with a maximum bond dimension of 150. To this end, we utilize the ITensor package [Fishman et al., 2022] in `Julia`.

Based on these generative models, we compute indicators of phase transitions [Equations (4.5), (4.7), and (4.45)], see Figures 4.4(c)-(e). The signals correctly reproduce the phase diagram obtained from physics-informed quantities, such as the *nonlocal* string order parameter [cf. Figure 4.4(b)], demonstrating the applicability of our framework to systems featuring topological order.

Note that the problem of detecting a transition from a topological phase to a topologically trivial phase is significantly easier than properly recognizing the topological phase itself. It may be solved by detecting the topologically trivial phase (for example, by constructing its local order parameter) and labeling everything that is not topologically trivial as topological. This may explain the low number of samples (in comparison to the relevant state space) needed to map out the phase diagram of the cluster-Ising model despite choosing measurement bases composed of product states. In the future, it will be interesting to study systems that feature a transition between two topological phases of different natures. In such a setup, the aforementioned strategy would not apply.

Example: Bose-Hubbard Model

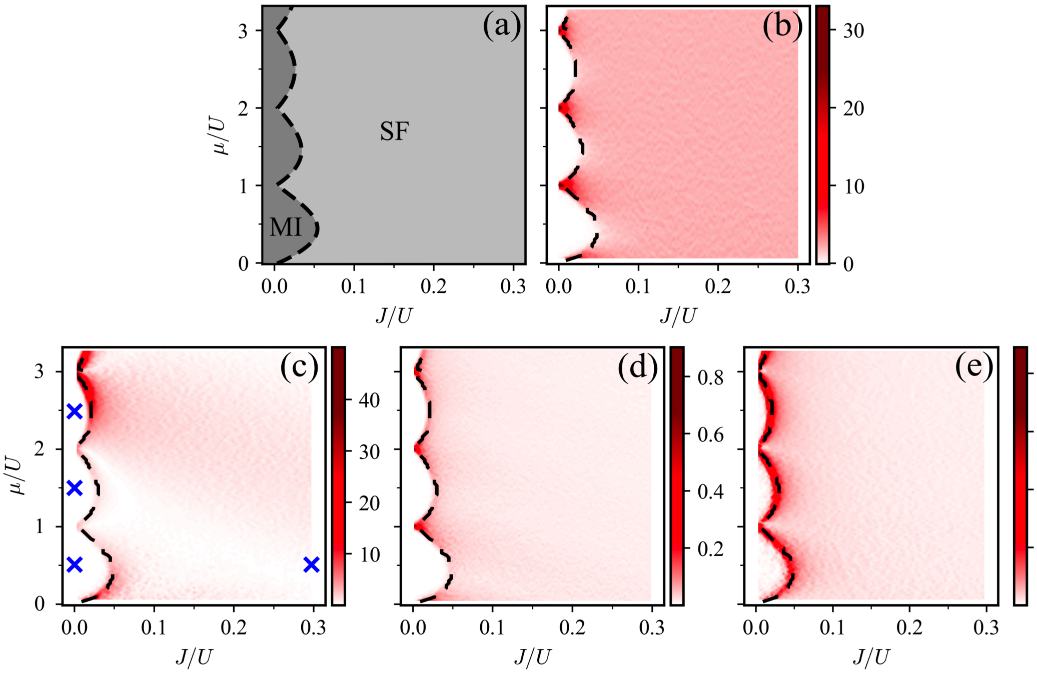

FIGURE 4.5: Results for the mean-field two-dimensional Bose-Hubbard model [Equation (4.25)]. (a) Schematic illustration of the phase diagram featuring two distinct phases: a Mott-insulating phase (left; MI) and a superfluid phase (right; SF). (b) Compressibility $\kappa = \partial\langle n_i \rangle/\partial\gamma_2$. (c) $I_{\mathrm{SL}}(\boldsymbol{\gamma})$ [Equation (4.5)] where the sets of points $\{\Gamma_y\}_{y\in\mathcal{Y}}$ representative of each phase are marked by blue crosses ($\mathcal{Y} = \{0,1\}$). Here, we choose three points within the Mott-insulating phase and a single point within the superfluid phase. (d) $I_{\mathrm{LBC}}(\boldsymbol{\gamma})$ [Equation (4.7)] with $l = 1$. (e) $I_{\mathrm{PBM}}(\boldsymbol{\gamma})$ [Equation (4.45)] where $\hat{\boldsymbol{\gamma}}(\boldsymbol{x})$ is estimated elementwise from line scans. The expected value involved in the computation of I_{SL}, I_{LBC}, and I_{PBM} are computed exactly. The set Γ is composed of a uniform grid with 101 points for J/U (i.e., γ_1) and 110 points for the μ/U (i.e., γ_2). Estimated phase boundaries (black dashed lines) are determined from maxima in $|\partial^2\langle H\rangle_{\boldsymbol{\gamma}}/\partial\gamma_1^2|$.

Finally, to illustrate the versatility of our framework, we map out the ground-state phase diagram of the two-dimensional Bose-Hubbard model using generative classifiers based on mean-field descriptions. The Hamiltonian is given by

$$H = -J \sum_{\langle ij \rangle} (b_i^\dagger b_j + \text{h.c.}) + \sum_i \frac{U}{2} n_i(n_i - 1) - \mu n_i, \tag{4.25}$$

with J denoting the nearest-neighbor hopping strength, U the on-site interaction strength, μ the chemical potential, and $\langle \cdot \rangle$ a sum over nearest neighbors. The two tuning parameters are $\boldsymbol{\gamma} = (J/U, \mu/U)$. As $\gamma_1 = J/U$ is increased at a fixed chemical potential $\gamma_2 = \mu/U$, the model undergoes a quantum phase transition from a Mott-insulating phase to a superfluid phase [Fisher *et al.*, 1989; Jaksch *et al.*, 1998], see Figure 4.5(a). We treat the model in mean-field using a Gutzwiller ansatz [Krauth *et al.*, 1992], i.e, we write the ground-state wave function as a product state

$$|\Psi_{\text{MF}}(\boldsymbol{\gamma})\rangle = \prod_i |\phi_i(\boldsymbol{\gamma})\rangle, \tag{4.26}$$

where

$$|\phi_i(\boldsymbol{\gamma})\rangle = \sum_{n=0}^{n_{\max}} f_n(\boldsymbol{\gamma}) |n_i\rangle, \tag{4.27}$$

with $|n_i\rangle$ denoting the Fock state with n bosons at site i. We use simulated annealing [Comparin, 2017; Huembeli *et al.*, 2018] to optimize the Gutzwiller coefficients $\{|f_n(\boldsymbol{\gamma})|^2\}_{n=0}^{n_{\max}}$ with $n_{\max} = 20$.

We perform single-site projective measurements in the Fock basis where the probability to measure n bosons is given by $\tilde{P}(n|\boldsymbol{\gamma}) = |\langle n_i | \Psi_{\text{MF}}(\boldsymbol{\gamma})\rangle|^2 = |f_n(\boldsymbol{\gamma})|^2$. Based on the set of generative models $\{\tilde{P}(\cdot|\boldsymbol{\gamma})\}_{\boldsymbol{\gamma}\in\Gamma}$, we compute indicators of phase transitions [Equations (4.5), (4.7), and (4.45)], see Figures 4.5(c)-(e). The signals correctly reproduce the phase diagram obtained from physics-informed quantities, such as the compressibility $\kappa = \partial\langle n_i \rangle / \partial \gamma_2$, where the Mott-insulating phase has zero compressibility, i.e., the characteristic Mott lobes display a constant value of $\langle n_i \rangle$ [cf. Figure 4.5(b)]. Note that the Mott-insulator to superfluid transition fails to be detected by LBC when class imbalance is not properly taken care of, i.e., using the formulation of LBC as originally introduced in Chapters 2 and 3, see Appendix A. This highlights the importance of this modification we introduced in the present chapter.

4.5 Connections between generative and discriminative approaches

The computation of all three indicators of phase transitions [Equations (4.5), (4.7), and (4.45)] boils down to solving classification tasks. In Section 4.4, we showcased how such tasks can be solved in a generative manner. Here, we discuss how they can be solved in a discriminative manner. In this case, we look for the parameters $\boldsymbol{\theta}$ of a parametric model $\tilde{P}_{\boldsymbol{\theta}}(y|\boldsymbol{x})$ that minimize the following cross-entropy loss function

$$\mathcal{L}(\boldsymbol{\theta}) = -\frac{1}{|\mathcal{Y}|} \sum_{y\in\mathcal{Y}} \frac{1}{|\mathcal{D}_y|} \sum_{\boldsymbol{x}\in\mathcal{D}_y} \ln\left[\tilde{P}_{\boldsymbol{\theta}}(y|\boldsymbol{x})\right], \tag{4.28}$$

where $\mathcal{D}_y = \{\boldsymbol{x} \in \mathcal{D}_{\boldsymbol{\gamma}} | \boldsymbol{\gamma} \in \Gamma_y\}$ is the relevant data set drawn from $P(\boldsymbol{x}|y) = 1/|\Gamma_y| \sum_{\boldsymbol{\gamma}\in\Gamma_y} P(\boldsymbol{x}|\boldsymbol{\gamma})$ and $\mathcal{D}_{\boldsymbol{\gamma}}$ corresponding to a set of samples drawn from $P(\cdot|\boldsymbol{\gamma})$.

Here, $|\mathcal{D}_{\boldsymbol{\gamma}}|$ is the same for all $\boldsymbol{\gamma} \in \Gamma_y$ (corresponding to the choice $P(\boldsymbol{\gamma}|y) = 1/|\Gamma_y|$ for $\boldsymbol{\gamma} \in \Gamma_y$ and zero otherwise). Class imbalance is addressed by the rescaling factors $1/|\mathcal{D}_y|$ as is common in NN training [Paszke *et al.*, 2019].

Based on the trained parametric model, in SL we can estimate the central quantity of interest $P(y|\boldsymbol{\gamma})$ as

$$P(y|\boldsymbol{\gamma}) \approx \frac{1}{|\mathcal{D}_{\boldsymbol{\gamma}}|} \sum_{\boldsymbol{x} \in \mathcal{D}_{\boldsymbol{\gamma}}} \tilde{P}_{\boldsymbol{\theta}}(y|\boldsymbol{x}). \tag{4.29}$$

In LBC, the relevant quantity p_{err} can, for example, be estimated as

$$p_{\text{err}} \approx \frac{1}{2} \sum_{y \in \{0,1\}} \frac{1}{|\mathcal{D}_y|} \sum_{\boldsymbol{x} \in \mathcal{D}_y} \left| y - \text{argmax}_{y'} \, \tilde{P}_{\boldsymbol{\theta}}(y'|\boldsymbol{x}) \right|. \tag{4.30}$$

In PBM, the key quantity is $\hat{\gamma}(\boldsymbol{x})$ which can be estimated as

$$\hat{\gamma}(\boldsymbol{x}) \approx \sum_{y \in \mathcal{Y}} \tilde{P}_{\boldsymbol{\theta}}(y|\boldsymbol{x}) \gamma_y. \tag{4.31}$$

Alternatively, it may also be approximated directly using a parametric predictive model $\hat{\boldsymbol{\gamma}}_{\boldsymbol{\theta}}(\boldsymbol{x})$ that is trained to solve a regression task instead of a classification task as we have done in Chapters 2 and 3. In this case, the relevant loss function is

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{|\mathcal{Y}|} \sum_{y \in \mathcal{Y}} \frac{1}{|\mathcal{D}_y|} \sum_{\boldsymbol{x} \in \mathcal{D}_y} ||\hat{\gamma}_{\boldsymbol{\theta}}(\boldsymbol{x}) - \gamma_y||_2^2 . \tag{4.32}$$

Above we have introduced the loss functions that are used to optimize parametric predictive models in the discriminative approach. In the following, we prove that in the infinite-data limit, an optimal predictive model, i.e., a model that minimizes the corresponding loss functions, yields Bayes-optimal predictions. Analyzing its predictions yields the indicator signals discussed in Section 4.2. Moreover, we show that an optimal discriminative classifier yields the same predictions as a generative classifier with $\tilde{P}(\boldsymbol{x}|\boldsymbol{\gamma})$ given by the empirical distribution obtained from the dataset $\mathcal{D}_{\boldsymbol{\gamma}}$.

Proof

For a given sample $\boldsymbol{x} \in \{\boldsymbol{x} \in \mathcal{D}_y | y \in \mathcal{Y}\}$, we can determine the corresponding empirically optimal model prediction $P_{\text{emp}}(y|\boldsymbol{x})$ by minimizing the loss function in Equation (4.28) with respect to $\tilde{P}_{\boldsymbol{\theta}}(y|\boldsymbol{x})$ subjected to the equality constraint

$$\sum_{y \in \mathcal{Y}} \tilde{P}_{\boldsymbol{\theta}}(y|\boldsymbol{x}) - 1 = 0. \tag{4.33}$$

Using the method of Lagrange multipliers, for any $y \in \mathcal{Y}$ the stationary points satisfy the following conditions

$$\lambda = \frac{1}{|\mathcal{Y}|} \frac{\mathcal{D}_y(\boldsymbol{x})}{|\mathcal{D}_y|} \frac{1}{P_{\text{emp}}(y|\boldsymbol{x})}, \tag{4.34}$$

where $\mathcal{D}_y(\boldsymbol{x})$ denotes the number of times $\boldsymbol{x}$ appears in $\mathcal{D}_y$ and λ is the corresponding Lagrange multiplier. Together with the equality constraint in Equation (4.33), this yields

$$P_{\text{emp}}(y|\boldsymbol{x}) = \frac{\frac{\mathcal{D}_y(\boldsymbol{x})}{|\mathcal{D}_y|}\frac{1}{|\mathcal{Y}|}}{\sum_{y'\in\mathcal{Y}}\frac{\mathcal{D}_{y'}(\boldsymbol{x})}{|\mathcal{D}_{y'}|}\frac{1}{|\mathcal{Y}|}}. \tag{4.35}$$

Identifying the empirical distribution $P_{\text{emp}}(\boldsymbol{x}|y) = \mathcal{D}_y(\boldsymbol{x})/|\mathcal{D}_y|$ as well as the uniform prior over the classes, $P(y) = 1/|\mathcal{Y}|$ for any $y \in \mathcal{Y}$, we have

$$P_{\text{emp}}(y|\boldsymbol{x}) = \frac{P_{\text{emp}}(\boldsymbol{x}|y)P(y)}{\sum_{y'\in\mathcal{Y}}P_{\text{emp}}(\boldsymbol{x}|y')P(y')}. \tag{4.36}$$

In the infinite-data limit $|\mathcal{D}_y| \to \infty$, we have $P_{\text{emp}}(\cdot|y) \to P(\cdot|y)$. Thus, the predictions of the empirically optimal model converge to the predictions of a Bayes-optimal model

$$P_{\text{emp}}(y|\boldsymbol{x}) \to P(y|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|y)P(y)}{\sum_{y'\in\mathcal{Y}}P(\boldsymbol{x}|y')P(y')}, \tag{4.37}$$

and we have recovered Equation (4.1). Moreover, based on these optimal predictions, we straightforwardly recover the optimal indicator of SL [Equation (4.5)]. To recover the optimal indicator of LBC, it remains to be shown that the estimated error rate in Equation (4.30) converges to the error rate as defined in Equation (4.6) in the case of an optimal discriminative model and infinite data. In this limit, Equation (4.30) transforms to

$$\begin{aligned} p_{\text{err}} &= \frac{1}{2}\sum_{y\in\{0,1\}}\sum_{\boldsymbol{x}\in\mathcal{X}} P(\boldsymbol{x}|y)\left|y - \operatorname{argmax}_{y'} P(y'|\boldsymbol{x})\right| \\ &= \frac{1}{2}\sum_{\substack{\boldsymbol{x}\in\mathcal{X}\\ P(\boldsymbol{x}|0)\geq P(\boldsymbol{x}|1)}} P(\boldsymbol{x}|1) + \frac{1}{2}\sum_{\substack{\boldsymbol{x}\in\mathcal{X}\\ P(\boldsymbol{x}|1)> P(\boldsymbol{x}|0)}} P(\boldsymbol{x}|0) \\ &= \frac{1}{2}\left(1 - \frac{1}{2}\sum_{\boldsymbol{x}\in\mathcal{X}}|P(\boldsymbol{x}|0) - P(\boldsymbol{x}|1)|\right). \end{aligned} \tag{4.38}$$

For the last step, we used the fact that for any two probability distributions p and q over a discrete variable $x \in \mathcal{X}$, we have

$$\frac{1}{2}\sum_{x\in\mathcal{X}}|p(x) - q(x)| = 1 - \sum_{\substack{x\in\mathcal{X}\\ q(x)>p(x)}} p(x) - \sum_{\substack{x\in\mathcal{X}\\ p(x)\geq q(x)}} q(x). \tag{4.39}$$

Continuing with Equation (4.38), we have

$$\begin{aligned} p_{\mathrm{err}} &= \frac{1}{2}\left(\sum_{\boldsymbol{x}\in\mathcal{X}} P(\boldsymbol{x}) - \sum_{\boldsymbol{x}\in\mathcal{X}} P(\boldsymbol{x})\,|P(0|\boldsymbol{x}) - P(1|\boldsymbol{x})|\right) \\ &= \sum_{\boldsymbol{x}\in\mathcal{X}} P(\boldsymbol{x})\left(\frac{1}{2}\Big[1 - |P(0|\boldsymbol{x}) - P(1|\boldsymbol{x})|\Big]\right) \\ &= \sum_{\boldsymbol{x}\in\mathcal{X}} P(\boldsymbol{x})\min\{P(0|\boldsymbol{x}), P(1|\boldsymbol{x})\}, \end{aligned} \tag{4.40}$$

where in step 2 we used Bayes' theorem $P(\boldsymbol{x}|y)/P(\boldsymbol{x}) = P(y|\boldsymbol{x})/P(y)$ with $P(y) = 1/2$. Plugging the definition of $P(\boldsymbol{x})$ given by

$$\begin{aligned} P(\boldsymbol{x}) &= \sum_{y\in\{0,1\}} P(\boldsymbol{x}|y)P(y) = \frac{1}{2}\sum_{y\in\{0,1\}}\sum_{\gamma\in\Gamma} P(\boldsymbol{x}|\gamma)P(\gamma|y) \\ &= \frac{1}{2}\sum_{y\in\{0,1\}}\frac{1}{|\Gamma_y|}\sum_{\gamma\in\Gamma_y} P(\boldsymbol{x}|\gamma) \end{aligned} \tag{4.41}$$

into Equation (4.40) and defining $p_{\mathrm{err}}(\boldsymbol{x}) = \min\{P(0|\boldsymbol{x}), P(1|\boldsymbol{x})\}$ we recover the error rate in Equation (4.6).

For PBM, following a similar minimization procedure for the loss function in Equation (4.32), we obtain

$$\begin{aligned} \hat{\gamma}_{\mathrm{emp}}(\boldsymbol{x}) &= \sum_{y\in\mathcal{Y}} \frac{\frac{\mathcal{D}_y(\boldsymbol{x})}{|\mathcal{D}_y|}\frac{1}{|\mathcal{Y}|}}{\sum_{y'\in\mathcal{Y}}\frac{\mathcal{D}_{y'}(\boldsymbol{x})}{|\mathcal{D}_{y'}|}\frac{1}{|\mathcal{Y}|}}\gamma_y \\ &= \sum_{y\in\mathcal{Y}} \frac{P_{\mathrm{emp}}(\boldsymbol{x}|y)P(y)}{\sum_{y'\in\mathcal{Y}} P_{\mathrm{emp}}(\boldsymbol{x}|y')P(y')}\gamma_y. \end{aligned} \tag{4.42}$$

In the infinite-data limit, this converges to

$$\hat{\gamma}(\boldsymbol{x}) = \sum_{y\in\mathcal{Y}} \frac{P(\boldsymbol{x}'|y)P(y)}{\sum_{y'\in\mathcal{Y}} P(\boldsymbol{x}'|y')P(y')}\gamma_y = \sum_{y\in\mathcal{Y}} P(y|\boldsymbol{x})\gamma_y, \tag{4.43}$$

corresponding to Equation (4.8). Based on these predictions, we directly recover the indicator of PBM in Equation (4.45).
Finally, note that using

$$\begin{aligned} P_{\mathrm{emp}}(\boldsymbol{x}|y) &= \frac{1}{|\Gamma_y|}\sum_{\gamma\in\Gamma_y}\frac{\mathcal{D}_\gamma(\boldsymbol{x})}{|\mathcal{D}_\gamma|} \\ &= \frac{1}{|\Gamma_y|}\sum_{\gamma\in\Gamma_y} P_{\mathrm{emp}}(\boldsymbol{x}|\gamma), \end{aligned} \tag{4.44}$$

in Equation (4.36), we obtain the same expression as in Equation (4.3) with $P(\boldsymbol{x}|\gamma)$ replaced by $P_{\mathrm{emp}}(\boldsymbol{x}|\gamma)$. That is, the predictions of an optimal discriminative model are equivalent to the ones of a generative model with $\tilde{P}(\cdot|\gamma) = P_{\mathrm{emp}}(\cdot|\gamma)$.

4.5.1 Numerical comparison

In this section, we will compare the computational cost associated with the discriminative and generative approaches to mapping out phase diagrams in detail. Let $t_{\text{eval}}^{\text{gen}}$ and $t_{\text{eval}}^{\text{discr}}$ be the times corresponding to evaluating the generative model $\tilde{P}(\boldsymbol{x}|\boldsymbol{\gamma})$ or discriminative model $\tilde{P}(y|\boldsymbol{x})$ for a given $\boldsymbol{x}$, respectively. These times depend heavily on the choice of model, implementation, and hardware. Here, the reported computation times were assessed in `Julia` (version 1.8.2) on a single 3.70 GHz Intel Xeon W-2135 CPU. For the nonparametric generative model of the anisotropic Ising model [$L = 20$, Equation (4.21)], the evaluation time is negligible $t_{\text{eval}}^{\text{gen}} \approx 40$ ns. In the case of the matrix product state-based generative models for the ground states of the cluster-Ising Hamiltonian [Equation (4.23)], the evaluation time $t_{\text{eval}}^{\text{gen}}$ ranges from ≈ 96 μs for $L = 7$ (on average across the sampled parameter space) to ≈ 3 ms for $L = 71$. For comparison, the evaluation time associated with a feedforward NN is $t_{\text{eval}}^{\text{discr}} \approx 7$ μs in the case of a single hidden layer containing a single node and, for example, $t_{\text{eval}}^{\text{discr}} \approx 2$ ms in the case of five hidden layers containing 1024 nodes each (assuming only a single input and output node for simplicity).

The computation time of a single prediction via the generative approach, i.e., computation of $\tilde{P}_{\text{gen}}(y|\boldsymbol{x})$ for all $y \in \mathcal{Y}$ given the sample $\boldsymbol{x}$ via Equation (4.3), scales as $t_{\text{pred}}^{\text{gen}} = t_{\text{eval}}^{\text{gen}}|\Gamma_{\mathcal{Y}}|$, where $\Gamma_{\mathcal{Y}} = \{\boldsymbol{\gamma} \in \Gamma_y | y \in \mathcal{Y}\}$. In Section 4.4, in SL we have $|\Gamma_{\mathcal{Y}}| = G = 5$ and 3, in LBC, $|\Gamma_{\mathcal{Y}}| = 2l = 2$, and in PBM, $|\Gamma_{\mathcal{Y}}| = 60$ and 101, in the case of the anisotropic Ising and cluster-Ising model, respectively. Here, in the case of PBM we consider elementwise estimators from corresponding line scans.

The time associated with computing $\tilde{P}_{\text{discr}}(y|\boldsymbol{x})$ via the discriminative approach is $t_{\text{pred}}^{\text{discr}} = t_{\text{eval}}^{\text{discr}}$. Given a model $\tilde{P}(y|\boldsymbol{x})$ (either discriminative or generative), the time associated with computing an indicator $I(\boldsymbol{\gamma})$ for all $\boldsymbol{\gamma} \in \Gamma$ scales as $t_{\text{pred}}|\Gamma||\mathcal{D}_{\boldsymbol{\gamma}}|$ in case of SL and PBM, and $t_{\text{pred}}|\Gamma||\mathcal{D}_{\boldsymbol{\gamma}}||\Gamma_y|$ in case of LBC. Based on this, for the anisotropic Ising model, the generative approach yields a speedup in the indicator computation of *at least* (assuming a feedforward NN of minimal size as the discriminative model) a factor of $t_{\text{pred}}^{\text{discr}}/t_{\text{pred}}^{\text{gen}} \approx 3$, 35, and 87 in the case of PBM, SL, and LBC, respectively.

In scenarios where the generative model acts as a data source, such as in the systems discussed in Section 4.4, the discriminative approach is guaranteed to have an overhead with respect to the generative approach due to training. We can estimate this overhead, i.e., the computation time required until the first evaluation of $\tilde{P}_{\text{discr}}(y|\boldsymbol{x})$, to scale as $t_{\text{eval}}^{\text{discr}}|\Gamma_{\mathcal{Y}}||\mathcal{D}_{\boldsymbol{\gamma}}|N_{\text{epochs}}$. Note that to predict the indicator in LBC, $d|\Gamma|$ such discriminative models need to be trained (one for each bipartition at each point in parameter space), making the overhead scale accordingly. The number of training epochs is denoted by N_{epochs}. In practice, N_{epochs} may range from $\mathcal{O}(10^2) - \mathcal{O}(10^4)$ and above. In cases where $\tilde{P}(\cdot|\boldsymbol{\gamma}) \approx P(\cdot|\boldsymbol{\gamma})$, for the discriminative approach to yield a predictive model of comparable quality, one expects it to be highly expressive (resulting in a high $t_{\text{eval}}^{\text{discr}}$), trained for a long time, and with a large dataset, resulting in a large overhead overall. The requirement of a larger dataset comes with an additional cost (with respect to the generative approach) associated with sample generation.

Because of this overhead, the generative approach can be more computationally efficient than the discriminative approach *even if* $t_{\text{pred}}^{\text{gen}} > t_{\text{pred}}^{\text{discr}}$. This is illustrated in Figure 4.6 for the case of the cluster-Ising model. For LBC [Figure 4.6(b)], the overhead of the discriminative approach is so large that the generative approach is more efficient irrespective of the error tolerance. In particular, in LBC, for small l it is expected to be difficult to achieve accurate results using the discriminative approach with a small dataset, because of the small differences in the underlying

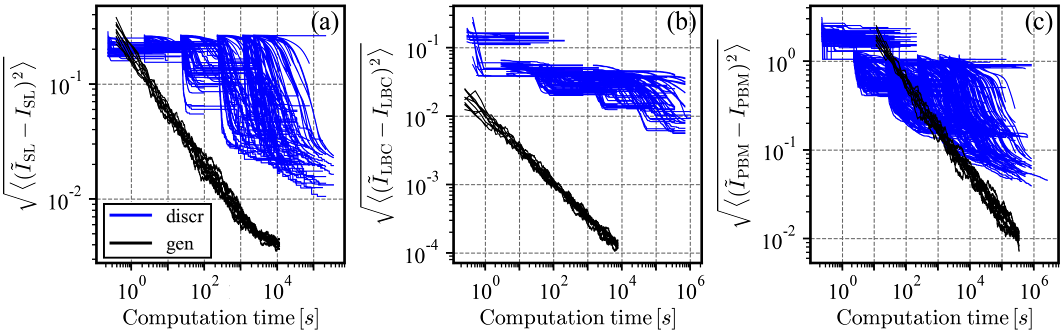

FIGURE 4.6: Root-mean-square error of estimated indicator of phase transitions $\tilde{I}$ as a function of the computation time associated with the discriminative (blue) and generative approach (black) to (a) SL, (b) LBC, and (c) PBM for the cluster-Ising model [Equation (4.23), $L = 7$] at $h_1/J = 0.2$. The total computation time is comprised of the time associated with data generation, training, and computation of the indicator based on the predictions, i.e., $\tilde{P}(y|\boldsymbol{x})$. For the data generation and construction of generative classifiers, we consider generative models based on matrix product states optimized via the density matrix renormalization group algorithm. The indicators $\tilde{I}$ are estimated from datasets $\{\mathcal{D}_\gamma\}_{\gamma\in\Gamma}$ of various size. The reference indicators I are computed exactly based on ground states obtained via exact diagonalization. The set Γ is composed of a uniform grid with 101 points. In SL, $\Gamma_1 = \{-1.5\ h_2/J\}$, $\Gamma_2 = \{-0.03\ h_2/J\}$, and $\Gamma_3 = \{1.47\ h_2/J\}$. In LBC, we consider $l = 2$. The discriminative approach to PBM is performed according to Equation (4.32). For the generative approach, 10 independent runs with distinct datasets are shown. For the discriminative approach, we considered feedforward NNs of various sizes (ranging from a single hidden layer with 64 nodes to five hidden layers with 64 nodes each; containing ReLUs as activation functions), different dataset sizes $|\mathcal{D}_\gamma|$ (ranging from 10 to $2 \cdot 10^5$), learning rates (ranging from $5 \cdot 10^{-5}$ to $5 \cdot 10^{-1}$), and number of training epochs (ranging from 1 to $5 \cdot 10^4$). Each blue curve corresponds to a training run with a fixed choice of hyperparameters and we keep track of the best root-mean-square error throughout training. The NNs are implemented using Flux [Innes, 2018] in `Julia`, where the weights and biases are optimized via gradient descent with Adam [Kingma and Ba, 2014]. Gradients are calculated using backpropagation [Baydin *et al.*, 2018]. The NN inputs are composed of six integers per qubit encoding the POVM element as well as the corresponding measurement outcome and are standardized before training.

distributions. For SL and PBM [Figures 4.6(a) and (c)], there exists a crossover point in terms of the error of the estimated indicator below which the generative approach is computationally more efficient. Note that hyperparameter tuning is required for the discriminative approach to yield accurate estimates of the indicator, which is not accounted for in the overall computation time. In contrast, using the generative approach, the estimated indicator can be systematically improved by sampling more data.

4.6 Review of method modifications

In Section 4.4, we demonstrated that the modified versions of SL, LBC, and PBM introduced in Section 4.2 can successfully map out phase diagrams of various systems using different generative models. Before we conclude this chapter, let us review these modifications in detail.

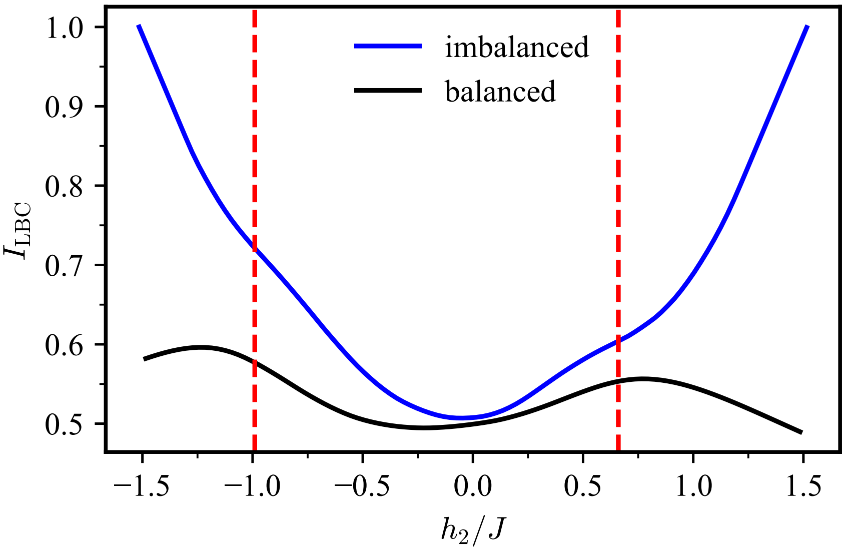

FIGURE 4.7: Results for the cluster-Ising model [Equation (4.23), $L = 7$] at $h_1/J = 0.2$ measured using a Pauli-6 POVM. The set Γ is composed of a uniform grid with 101 points. Indicator $I_{\mathrm{LBC}}(\gamma)$ with $l = 101$ (equivalent to $l = \infty$), i.e., considering global bipartitions of the one-dimensional parameter space. The (Bayes-optimal) indicator is computed in the presence (blue) or absence (black) of class imbalance corresponding to different choices of $P(y)$ in Equation (4.1). Here, the expected value involved in $I_{\mathrm{LBC}}(\gamma)$ is computed exactly and the ground state is obtained via exact diagonalization. Estimated critical points (red dashed lines) are determined from maxima in $|\partial^2\langle H\rangle_\gamma/\partial\gamma_2^2|$ for $L = 71$, see Figure 4.4.

4.6.1 Uniform distribution across class-specific parameter regions

When setting up the classification tasks underlying the phase-transition-detection problem, we choose a uniform distribution over the set of parameters associated with each class, i.e., $P(\boldsymbol{\gamma}|y) = 1/|\Gamma_y|$ for $\boldsymbol{\gamma} \in \Gamma_y$ and zero otherwise. This choice has been implicitly made via the definition of the loss function in previous works tackling the problem of mapping out phase diagrams with discriminative models, including Chapters 2 and 3. Due to the application of Bayes' theorem within the generative approach, such probabilistic assumptions are brought to light. In principle, our framework allows for a free choice of $P(\boldsymbol{\gamma}|y)$. It represents our freedom to choose to what degree different points in parameter space are considered representative of the label y. Given that the labeling procedure is different for the three methods for mapping out phase diagrams discussed in Section 4.2, its choice affects them differently.

In PBM [Equation (4.45)], each sampled value of the tuning parameter is considered its own class $\Gamma_y = \{\boldsymbol{\gamma}_y\}$. As such, setting $P(\boldsymbol{\gamma}|y) = 1$ if $\boldsymbol{\gamma} = \boldsymbol{\gamma}_y$ and zero otherwise is the only sensible choice. This corresponds to a special case of the uniform distribution.

In LBC [Equation (4.7)], the sets $\Gamma_0^{(i)}(\boldsymbol{\gamma})$ and $\Gamma_1^{(i)}(\boldsymbol{\gamma})$ are each comprised of the l points closest to $\boldsymbol{\gamma}$ in part I and II of the split parameter space. In Section 4.4, we consider $l = 1$ and each set is composed of a single point leaving only a single sensible choice as mentioned above. For $l > 1$, choosing $P(\boldsymbol{\gamma}|y)$ to be distinct from a uniform distribution would put different weights on different points within a given set. However, without additional system information, there is no compelling reason to do so. Moreover, note that for small l and dense sampling of the parameter space, reweighting is expected to have a marginal effect since, in this case, $P(\boldsymbol{x}|\boldsymbol{\gamma})$ is similar for points $\boldsymbol{\gamma}$ within each of the two sets.

In SL [Equation (4.5)], the sets $\{\Gamma_y\}_{y\in\mathcal{Y}}$ are chosen to be representative points within the G distinct phases of the system. As such, SL already allows for flexibility in choosing points representative of a given label. While additional flexibility in the weighting of these points can be introduced, without additional system information there is no compelling reason to do so.

4.6.2 Removing class imbalance

In traditional discriminative ML tasks, class imbalance occurs when the number of samples representing each class within the data set differs. In the classification tasks underlying the phase-transition-detection problem, this can, for example, occur when one does not have access to an equal amount of data at each point in parameter space. Moreover, in SL and LBC, an imbalance can also occur when sampling an *equal* amount of data at each point in parameter space. In both cases, an imbalance may arise if the sets $\{\Gamma_y\}_{y\in\mathcal{Y}}$ are not of equal size. In SL, this can be circumvented by choosing an equal number of points in parameter space to represent each phase. In LBC, data sets of unequal size occur as the bipartitions artificially divide the uniformly sampled one-dimensional parameter space. In previous works utilizing LBC, this class imbalance was not accounted for. As a result, the corresponding indicator I_{LBC} exhibits trivial local maxima ($I_{\mathrm{LBC}} = 1$) at the edges of the sampled region of the parameter space where the entire data is given the same label. If no phase transition is present, the indicator will thus have a characteristic V-shape. In the presence of a phase transition, a W-shape is expected, where the location of the intermediate maximum corresponds to the predicted critical point. Such behavior is undesirable for an indicator of phase transitions, and can even lead to a failure to detect certain transitions (or misdetection). In [Bohrdt *et al.*, 2021], an attempt has been made to correct this pathological behavior. However, we find that this procedure biases the transition point toward the center of the parameter range under consideration and hence does not seem viable, see Appendix F.

In this chapter, we addressed class imbalances arising from sets $\{\Gamma_y\}_{y\in\mathcal{Y}}$ of unequal size by choosing a uniform prior distribution $P(y)$. In particular, this removes the trivial local maxima at the edges of the sampled region of the parameter space previously encountered for the indicator I_{LBC} of LBC and we instead expect a single peak in the presence of a phase transition. Figure 4.7 shows an example of two transitions in the cluster-Ising model that are hardly visible using LBC in the presence of class imbalance, highlighting the importance of our proposed modification.

4.6.3 Division by standard deviation in prediction-based method

For PBM, we have proposed to divide the corresponding signal by its standard deviation. We find this modified indicator to be superior compared to its unmodified version

$$\begin{aligned} I'_{\mathrm{PBM}}(\boldsymbol{\gamma}) &= \sqrt{\sum_{i=1}^{d}\left(\frac{\partial \hat{\gamma}_i(\boldsymbol{\gamma})}{\partial \gamma_i}\right)^2}, \\ &= \left|\left|\mathbb{E}_{\boldsymbol{x}\sim P(\cdot|\boldsymbol{\gamma})}\left[\hat{\boldsymbol{\gamma}}(\boldsymbol{x})\nabla_{\boldsymbol{\gamma}} \ln\Big(P(\boldsymbol{x}|\boldsymbol{\gamma})\Big)\right]\right|\right|_2, \end{aligned} \tag{4.45}$$

which (up to a constant offset and absolute value) has been considered in previous studies [Schäfer and Lörch, 2019; Greplova *et al.*, 2020; Arnold *et al.*, 2021], including Chapters 2 and 3. As an example, Figure 4.8 compares the two indicators for the anisotropic Ising model. The indicator I'_{PBM} shows four distinct peaks with the dominant two occurring within the two ordered phases (i.e., at larger $|J_y/k_{\mathrm{B}}T|$). Such spurious signals have also been observed by Schäfer and Lörch [2019] which used the indicator where the standard deviation is not taken into account. Also recall the two peaks that were observed when studying the Ising model using PBM in Chapter 3 (see Figure 3.5). In contrast, the modified indicator I_{PBM} we propose in this chapter shows two distinct peaks that qualitatively agree with the two critical points. A further justification will be provided in Chapter 6 where we link the modified indicator to the system's Fisher information.

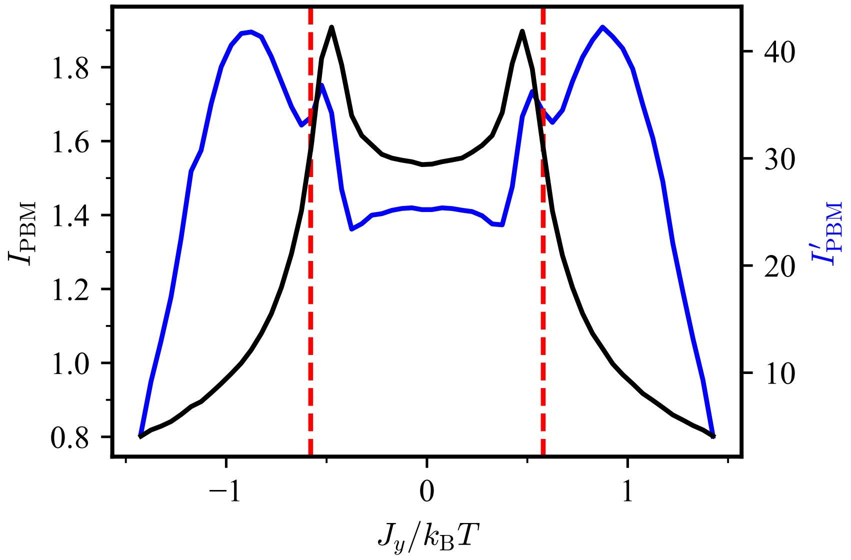

FIGURE 4.8: Results for the anisotropic Ising model on a square lattice ($L = 20$) at $J_x/k_{\mathrm{B}}T = -0.325$. The indicator $I_{\mathrm{PBM}}(\gamma)$ (black, left y-axis) and $I'_{\mathrm{PBM}}(\gamma)$ (blue, right y-axis) are computed based on predictions $\hat{\boldsymbol{\gamma}}$ that are estimated elementwise from line scans across the entire two-dimensional phase diagram. The set Γ is composed of a uniform grid with 60 points for each axis and $|\mathcal{D}_{\boldsymbol{\gamma}}| = 10^5$ for all $\boldsymbol{\gamma} \in \Gamma$. Onsager's analytical solution for the phase boundaries is shown as a red dashed line.

4.7 Summary

In this chapter, we have introduced generative classifiers as alternatives to discriminative classifiers to solve the classification tasks required for mapping out phase diagrams in an autonomous fashion, i.e., to detect the phase transitions underlying a physical system. The numerical procedure in Chapter 3 has been restricted to generative models with nonparametric descriptions, e.g., obtained via exact diagonalization or histogram binning, as well as analytically exact descriptions. This limited the applicability of the generative approach to systems with a tractable state space. Here, we showcased how this numerical procedure can be extended to all generative models with an explicit, tractable density. This encompasses a large class of methods for approximately describing many-body systems, ranging from autoregressive networks, such as fully visible belief networks or recurrent NNs, to normalizing flows, and tensor networks, as well as mean-field descriptions, allowing us to tackle systems with large state spaces.

The generative approach naturally allows for the incorporation of system knowledge, such as the Hamiltonian or the functional form of the relevant family of probability distributions. This makes the approach favorable for numerical investigations where such information is readily available, as we have explicitly demonstrated for classical systems in equilibrium as well as quantum ground states. In the case of the anisotropic Ising model, for example, we have identified a low-dimensional sufficient statistic. This allowed us to compress the state space without loss of information and construct accurate empirical distributions with few samples.

More generally, in numerical investigations, we often have direct access to a description of the system in terms of generative models. The generative approach can use this information to create a classifier, bypassing an explicit data-driven construction as in the discriminative approach.[7]

We have successfully generalized the phase-transition-detection methods of SL, LBC, and PBM making them applicable in arbitrary dimensional parameter spaces featuring multiple phases and phase boundaries. Moreover, we have suggested several modifications that helped to make the methods more robust at detecting phase transition and alleviate some of the failure modes previously encountered in Chapter 3. Overall, the present chapter establishes a powerful framework for the autonomous determination of phase diagrams with little to no human supervision.

4.8 Outlook

In this chapter, we showed applications to classical equilibrium systems and quantum ground states. Extensions to classical nonequilibrium systems, as well as other quantum states, are possible by adapting the generative model. Note that it is generally unclear how to identify (approximate) sufficient statistics in such systems. Hence, parametric approaches may likely need to be employed. For the dynamics of open quantum systems, for example, compatible ML-inspired ansätze [Luo *et al.*, 2022] and time-dependent variational principles [Reh *et al.*, 2021] have recently been proposed. If system knowledge is scarce, such as when characterizing quantum states prepared

[7] In previous works utilizing SL, LBC, or PBM to detect phase transitions in numerical investigations, the fact that such system information is readily available and can be utilized to make the procedure more efficient has simply been ignored. In many cases, this can likely be attributed to the authors focusing on eventually applying these methods in experimental settings, where such information may indeed be lacking.

in an experiment [Carrasquilla *et al.*, 2019; Gomez *et al.*, 2022; Fitzek *et al.*, 2024], generative models can be constructed in a data-driven manner, e.g., via maximum likelihood estimation.

When comparing the discriminative and generative approach in terms of computational resources, we have so far focused on the special case where the generative model itself acts as a data source. In this case, the computational cost associated with constructing the generative model is an overhead that is present in both the generative and discriminative approaches for detecting phase transitions and can effectively be ignored (see Section 4.5.1). Consequently, a generative approach is likely to beat a discriminative approach. This assumption may not be met, for example, when constructing empirical distributions and the resulting generative model is a poor approximation of the true underlying distribution. Typically, this occurs when the number of available samples is much smaller compared to the size of the relevant state space, $|\mathcal{D}_\gamma| \ll |\mathcal{X}|$. It is particularly hard to collect a sufficiently large number of samples if insufficient system knowledge is present to compress the state space, such as when analyzing experimental data. In [Zhang *et al.*, 2024a], we analyzed a small Ising model as an example and found that – in certain instances – a parametric generative approach (here using autoregressive PixelCNNs [Van Den Oord *et al.*, 2016]) can indeed perform on-par or even outperform a parametric discriminative approach in terms of the level of accuracy with which the Bayes-optimal indicator is reproduced at a given budget for computation time or number of samples. Determining whether a discriminative or generative approach is more suitable for detecting phase transitions in a given problem is nontrivial: the answer depends on many factors, such as how much system knowledge is available[8,9], the choice of model, the size of the available dataset, the required accuracy that needs to be reached, the computational budget, as well as the type of indicator that is being computed. Going beyond small Ising lattices, in future work, it remains to be analyzed to what extent the numerical advantage of generative approaches over discriminative approaches persists when training of the generative models needs to be explicitly accounted for.

Generative classifiers are highly versatile and applicable in various contexts. Because generative models play a fundamental role in many-body physics, it is expected that other tasks in this domain that can be cast as classification problems, such as testing physical theories [Bohrdt *et al.*, 2019; Zhang *et al.*, 2019b; Muñoz-Gil *et al.*, 2021], detecting entanglement [Lu *et al.*, 2018], or investigating thermodynamic principles [Seif *et al.*, 2021], will benefit from a generative approach. We will explore the possibility of analyzing generative models beyond physics in the second part of this thesis, particularly Chapter 8.

The results and figures presented in this chapter have been in parts published in [Arnold *et al.*, 2024c]. The corresponding code is open source [Arnold *et al.*, 2023b].

[8] For example, if the underlying system Hamiltonian is known we may be able to identify a sufficient statistic or efficiently train a parametric generative model via a variational principle. Otherwise, we generally need to work in the large state (configuration) space and train models in a data-driven manner.

[9] This does not only influence how we construct and train our models but also how we assess the associated computational cost. In case our data comes from an experiment, we are more likely to be concerned with producing accurate estimates of the critical point given a fixed sample budget and worry less about how much computation time goes into computing the corresponding indicators. In contrast, when working fully in simulation, computation time itself is a faithful metric.

Chapter 5

Background on Statistical Distances and Their Role in Inference Tasks

The results presented in this chapter are based on the following preprint:

Machine learning phase transitions: Connections to the Fisher information,
J. Arnold, F. Holtorf, N. Lörch, and F. Schäfer,
arXiv:2311.10710 (2023).

5.1 Motivation

In Chapter 4, we formulated the phase-transition-detection methods of SL, LBC, and PBM in a fully probabilistic fashion. At this stage, it will be useful to brush up on some fundamental concepts from statistics and information theory. In particular, in this chapter, we will review the concept of statistical distances and their role in information geometry as well as the statistical tasks of hypothesis testing and parameter estimation – tasks that ultimately underly SL, LBC, and PBM. Readers familiar with these topics may skip this chapter. For an extended overview, see [Casella and Berger, 2002; Bickel and Doksum, 2015; Jarzyna and Kołodyński, 2020]. This background forms the basis for our analysis in Chapter 6. In particular, it will enable us to put our observation that these three methods are based on measuring changes in the probability distributions underlying the system at hand on a more firm footing.

5.2 Statistical distances

The space composed of all valid probability distributions on a given probability space is referred to as a *statistical manifold* $\mathcal{M}$. The field of information geometry is concerned with the geometry of this manifold. It can give useful insights for dealing with statistical inference tasks such as parameter estimation or hypothesis testing: many ML methods for detecting phase transitions make use of such tasks, including the three methods of SL, LBC, and PBM we focus on in this thesis.

Let us start by reviewing the notion of *statistical distances*. They measure the distance between statistical objects, such as probability distributions. A statistical distance between two elements of the statistical manifold is some non-negative function $D : \mathcal{M} \times \mathcal{M} \to \mathbb{R}_{\geq 0}$. A statistical distance is a proper distance or *metric* if, in addition, it satisfies *i)* *symmetry*: $D[p,q] = D[q,p]$,

ii) *identity of indiscernibles*: $D[p,q] = 0 \iff p = q$,

iii) *triangle inequality*: $D[p,r] + D[r,q] \geq D[p,q]$,

for any choice of valid probability distributions $p, q, r \in \mathcal{M}$ defined over the state space $\mathcal{X}$.[1] It turns out that many statistical distances of interest do not satisfy all three of these criteria. An important class of statistical distances is the class of so-called f-divergences [Liese and Vajda, 2006].

Definition (f-divergence)
Given a convex function $f : \mathbb{R}_{\geq 0} \to \mathbb{R}$ with $f(1) = 0$, the corresponding f-divergence is a statistical distance defined as

$$D_f[p,q] = \sum_{\boldsymbol{x} \in \mathcal{X}} q(\boldsymbol{x}) f\left(\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\right). \tag{5.1}$$

In general, an f-divergence does not constitute a proper metric. The non-negativity of f-divergences follows from their convexity via Jensen's inequality. Moreover, if $f(x)$ is strictly convex at $x = 1$, the corresponding f-divergence can be shown to satisfy the identity of indiscernible, which justifies referring to D_f as a divergence.

The *total variation* (TV) distance is an f-divergence that is going to turn out to be particularly useful for us later on in this thesis. It is defined as

$$D_{\mathrm{TV}}[p,q] = \frac{1}{2} \sum_{\boldsymbol{x} \in \mathcal{X}} |p(\boldsymbol{x}) - q(\boldsymbol{x})|. \tag{5.2}$$

The TV distance is the f-divergence with $f(x) = \frac{1}{2}|1 - x|$. In contrast to other f-divergences, the function f of the TV distance is not differentiable at 1. Other important examples of f-divergences include the *Kullback-Leibler* (KL) divergence

$$D_{\mathrm{KL}}[p,q] = \sum_{\boldsymbol{x} \in \mathcal{X}} p(\boldsymbol{x}) \ln\left(\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\right), \tag{5.3}$$

with $f(x) = x\ln(x)$, and the *Jensen-Shannon* (JS) divergence

$$D_{\mathrm{JS}}[p,q] = \frac{1}{2} D_{\mathrm{KL}}\left[p, \frac{p+q}{2}\right] + \frac{1}{2} D_{\mathrm{KL}}\left[q, \frac{p+q}{2}\right], \tag{5.4}$$

with $f(x) = \frac{1}{2}\left[x\ln(\frac{2x}{1+x}) + \ln(\frac{2}{1+x})\right]$. Note that the generating function f of a given f-divergence is not uniquely defined, but only up to an affine term. That is, $D_f = D_g$ if $f(x) = g(x) + c(x-1)$ for some constant $c \in \mathbb{R}$.

While f-divergences are not proper metrics, they satisfy other crucial properties. For example, a good statistical distance should capture the information loss associated with data processing. As such, it should fulfill the so-called data-processing inequality.

Proposition (Data-processing inequality)
Consider a mapping from $\mathcal{X}$ to an alternative space $\mathcal{Z}$, $S : \mathcal{X} \to \mathcal{Z}$, such that $p(\boldsymbol{z}) = \sum_{\boldsymbol{x} \in \mathcal{X}} W(\boldsymbol{z}|\boldsymbol{x}) p(\boldsymbol{x})$ with W being a left-stochastic transition matrix, i.e., a matrix with non-negative entries and columns summing up to one. Then, $D_f[p(\boldsymbol{x}), q(\boldsymbol{x})] \geq D_f[p(\boldsymbol{z}), q(\boldsymbol{z})]$ for any f-divergence D_f.

[1] For simplicity, in what follows, we assume the state space $\mathcal{X}$ to be discrete and countable.

Proof

$$\begin{aligned}
D_f[p(\boldsymbol{x}), q(\boldsymbol{x})] &= \sum_{\boldsymbol{x}\in\mathcal{X}} q(\boldsymbol{x}) f\left(\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\right) = \sum_{\boldsymbol{x}\in\mathcal{X},\boldsymbol{z}\in\mathcal{Z}} q(\boldsymbol{x}) W(\boldsymbol{z}|\boldsymbol{x}) f\left(\frac{p(\boldsymbol{x})W(\boldsymbol{z}|\boldsymbol{x})}{q(\boldsymbol{x})W(\boldsymbol{z}|\boldsymbol{x})}\right) \\
&= \sum_{\boldsymbol{x}\in\mathcal{X},\boldsymbol{z}\in\mathcal{Z}} q(\boldsymbol{x},\boldsymbol{z}) f\left(\frac{p(\boldsymbol{x},\boldsymbol{z})}{q(\boldsymbol{x},\boldsymbol{z})}\right) = D_f[p(\boldsymbol{x},\boldsymbol{z}), q(\boldsymbol{x},\boldsymbol{z})] \\
&= \sum_{\boldsymbol{z}\in\mathcal{Z}} q(\boldsymbol{z}) \sum_{\boldsymbol{x}\in\mathcal{X}} q(\boldsymbol{x}|\boldsymbol{z}) f\left(\frac{p(\boldsymbol{z})p(\boldsymbol{x}|\boldsymbol{z})}{q(\boldsymbol{z})q(\boldsymbol{x}|\boldsymbol{z})}\right) \\
&\geq \sum_{\boldsymbol{z}\in\mathcal{Z}} q(\boldsymbol{z}) f\left(\sum_{\boldsymbol{x}\in\mathcal{X}} q(\boldsymbol{x}|\boldsymbol{z}) \frac{p(\boldsymbol{z})p(\boldsymbol{x}|\boldsymbol{z})}{q(\boldsymbol{z})q(\boldsymbol{x}|\boldsymbol{z})}\right) \qquad (5.5)
\end{aligned}$$

where we have used Jensen's inequality in the last step. Finally noting that

$$\begin{aligned}
\sum_{\boldsymbol{z}\in\mathcal{Z}} q(\boldsymbol{z}) f\left(\sum_{\boldsymbol{x}\in\mathcal{X}} q(\boldsymbol{x}|\boldsymbol{z}) \frac{p(\boldsymbol{z})p(\boldsymbol{x}|\boldsymbol{z})}{q(\boldsymbol{z})q(\boldsymbol{x}|\boldsymbol{z})}\right) &= \sum_{\boldsymbol{z}\in\mathcal{Z}} q(\boldsymbol{z}) f\left(\sum_{\boldsymbol{x}\in\mathcal{X}} p(\boldsymbol{x}|\boldsymbol{z}) \frac{p(\boldsymbol{z})}{q(\boldsymbol{z})}\right) \\
&= \sum_{\boldsymbol{z}\in\mathcal{Z}} q(\boldsymbol{z}) f\left(\frac{p(\boldsymbol{z})}{q(\boldsymbol{z})}\right) \\
&= D_f[p(\boldsymbol{z}), q(\boldsymbol{z})] \qquad (5.6)
\end{aligned}$$

completes the proof.

The intuition is that processing $\boldsymbol{x}$ (via a physical operation described by a Markov process) can only make it more difficult to distinguish two distributions. Note that statistical distances that satisfy the data processing inequality are also called *monotonic* (under stochastic maps).

A good statistical distance should also be invariant under mappings between sample spaces that preserve all "relevant information" about $\boldsymbol{x}$. To this end, let us endow the statistical manifold $\mathcal{M}$ with a coordinate system by parametrizing all probability distributions in the manifold $p \mapsto p_{\boldsymbol{\gamma}}, \boldsymbol{\gamma} \in \mathbb{R}^d$, where $d = \dim \mathcal{M}$. In this case, a *sufficient* statistic is a quantity that encodes all relevant information about the value of the parameter $\boldsymbol{\gamma}$. The statistic is sufficient in the sense that there does not exist any other quantity that could be calculated from the samples $\boldsymbol{x}$ that would provide additional information regarding the value of the parameter.

Definition (Sufficient statistic)
Given a mapping $S : \mathcal{X} \to \mathcal{Z}$, the statistic $S(\boldsymbol{x})$ is sufficient for $\boldsymbol{\gamma}$ if and only if $\forall \boldsymbol{\gamma}, \boldsymbol{z}$ we have that $p_{\boldsymbol{\gamma}}(\boldsymbol{x}|\boldsymbol{z} = S(\boldsymbol{x}))$ is independent of $\boldsymbol{\gamma}$.

Having specified what is meant by relevant information, we can verify that f-divergences are indeed invariant under mappings between sample spaces that leave this information intact. In particular, f-divergences can be shown to be invariant under mappings $S : \mathcal{X} \to \mathcal{Z}$ where the statistic $S(\boldsymbol{x})$ is sufficient for $\boldsymbol{\gamma}$.

Proposition (Invariance under sufficient statistic)
Consider a mapping $S : \mathcal{X} \to \mathcal{Z}$ where the statistic $S(\boldsymbol{x})$ is sufficient for $\boldsymbol{\gamma}$. Then, $D_f[p_{\boldsymbol{\gamma}_1}(\boldsymbol{x}), p_{\boldsymbol{\gamma}_2}(\boldsymbol{x})] = D_f[p_{\boldsymbol{\gamma}_1}(\boldsymbol{z}), p_{\boldsymbol{\gamma}_2}(\boldsymbol{z})]$ for any choice of $\boldsymbol{\gamma}_1$, $\boldsymbol{\gamma}_2$, and f-divergence.

Proof

Any f-divergence satisfies the data-processing inequality. During data processing, equality is achieved if

$$\frac{p(\boldsymbol{x}|\boldsymbol{z})}{q(\boldsymbol{x}|\boldsymbol{z})} = 1. \tag{5.7}$$

Choosing $p = p_{\gamma_1}$ and $q = p_{\gamma_2}$, this is satisfied given that $S(\boldsymbol{x})$ is a sufficient statistic.

The Fisher-Neyman factorization theorem provides another convenient characterization of a sufficient statistic.

Theorem (Fisher-Neyman factorization theorem)
Given a mapping $S : \mathcal{X} \to \mathcal{Z}$, the statistic $S(\boldsymbol{x})$ is sufficient for $\boldsymbol{\gamma}$ if and only if non-negative functions h and g can be found such that $p_{\boldsymbol{\gamma}}(\boldsymbol{x}) = h(\boldsymbol{x})g_{\boldsymbol{\gamma}}(S(\boldsymbol{x}))$ for all choices of $\boldsymbol{\gamma}, \boldsymbol{x}$.

Proof

($\Longrightarrow$): If $\boldsymbol{z} = S(\boldsymbol{x})$, we have

$$p_{\boldsymbol{\gamma}}(\boldsymbol{x}, \boldsymbol{z}) = p_{\boldsymbol{\gamma}}(\boldsymbol{x})p_{\boldsymbol{\gamma}}(\boldsymbol{z}|\boldsymbol{x}) = p_{\boldsymbol{\gamma}}(\boldsymbol{x}) \tag{5.8}$$

given that $p_{\boldsymbol{\gamma}}(\boldsymbol{z}|\boldsymbol{x}) = 1$. Thus,

$$p_{\boldsymbol{\gamma}}(\boldsymbol{x}) = p_{\boldsymbol{\gamma}}(\boldsymbol{x}, \boldsymbol{z}) = p_{\boldsymbol{\gamma}}(\boldsymbol{x}|\boldsymbol{z})p_{\boldsymbol{\gamma}}(\boldsymbol{z}) = p(\boldsymbol{x}|\boldsymbol{z})p_{\boldsymbol{\gamma}}(\boldsymbol{z}), \tag{5.9}$$

where the last equality follows by invoking that $\boldsymbol{z} = S(\boldsymbol{x})$ is a sufficient statistic. Therefore, with $h(\boldsymbol{x}) \equiv p(\boldsymbol{x}|\boldsymbol{z})$ and $g_{\boldsymbol{\gamma}}(\boldsymbol{z}) \equiv p_{\boldsymbol{\gamma}}(\boldsymbol{z})$, we obtain $p_{\boldsymbol{\gamma}}(\boldsymbol{x}) = h(\boldsymbol{x})g_{\boldsymbol{\gamma}}(S(\boldsymbol{x}))$.

($\Longleftarrow$): We have

$$p_{\boldsymbol{\gamma}}(\boldsymbol{z}) = \sum_{\boldsymbol{x}\in\mathcal{X};\; S(\boldsymbol{x})=\boldsymbol{z}} p_{\boldsymbol{\gamma}}(\boldsymbol{x}, \boldsymbol{z}) = \sum_{\boldsymbol{x}\in\mathcal{X};\; S(\boldsymbol{x})=\boldsymbol{z}} p_{\boldsymbol{\gamma}}(\boldsymbol{x}). \tag{5.10}$$

Inserting $p_{\boldsymbol{\gamma}}(\boldsymbol{x}) = h(\boldsymbol{x})g_{\boldsymbol{\gamma}}(\boldsymbol{z})$, we obtain

$$p_{\boldsymbol{\gamma}}(\boldsymbol{z}) = \sum_{\boldsymbol{x}\in\mathcal{X};\; S(\boldsymbol{x})=\boldsymbol{z}} h(\boldsymbol{x})g_{\boldsymbol{\gamma}}(\boldsymbol{z}) = \left(\sum_{\boldsymbol{x}\in\mathcal{X};\; S(\boldsymbol{x})=\boldsymbol{z}} h(\boldsymbol{x}) \right) g_{\boldsymbol{\gamma}}(\boldsymbol{z}). \tag{5.11}$$

The quantity $p_{\boldsymbol{\gamma}}(\boldsymbol{x}|\boldsymbol{z}) = p_{\boldsymbol{\gamma}}(\boldsymbol{x}, \boldsymbol{z})/p_{\boldsymbol{\gamma}}(\boldsymbol{z}) = p_{\boldsymbol{\gamma}}(\boldsymbol{x})/p_{\boldsymbol{\gamma}}(\boldsymbol{z})$ is thus independent of $\boldsymbol{\gamma}$, and

$$p_{\boldsymbol{\gamma}}(\boldsymbol{x}|\boldsymbol{z}) = \frac{h(\boldsymbol{x})g_{\boldsymbol{\gamma}}(\boldsymbol{z})}{\left(\sum_{\boldsymbol{x}\in\mathcal{X};\; S(\boldsymbol{x})=\boldsymbol{z}} h(\boldsymbol{x})\right) g_{\boldsymbol{\gamma}}(\boldsymbol{z})} = \frac{h(\boldsymbol{x})}{\left(\sum_{\boldsymbol{x}\in\mathcal{X};\; S(\boldsymbol{x})=\boldsymbol{z}} h(\boldsymbol{x})\right)}. \tag{5.12}$$

This proves that $S(\boldsymbol{x})$ is a sufficient statistic.

Interestingly, the factorized form guaranteed by the Fisher-Neyman factorization theorem, i.e., the fact that the dependence of $\boldsymbol{x}$ on $\boldsymbol{\gamma}$ only enters through the sufficient statistic $S(\boldsymbol{x})$, implies that Bayes-optimal estimates of parameters as well as strategies in hypothesis testing only depend on the sufficient statistic. That is, the optimal indicators of phase transitions of SL, LBC, and PBM can be computed solely from the sufficient statistic as opposed to measurements capturing the full state space. In Chapter 4 (Section 4.4.1), we have explicitly proven this for the special case of the sufficient statistic associated with distributions belonging to the exponential family.

5.3 Information geometry

Consider any statistical distance $D[p,q]$ smooth in $p,q \in \mathcal{M}$. We have

$$D[p_{\boldsymbol{\gamma}}, p_{\boldsymbol{\gamma}+\boldsymbol{\Delta\gamma}}] = \frac{1}{2}\boldsymbol{\Delta\gamma}^T H_D(\boldsymbol{\gamma})\boldsymbol{\Delta\gamma} + \mathcal{O}(\boldsymbol{\Delta\gamma}^3), \tag{5.13}$$

where we use the fact that $D[p,p] = 0$ and the first-order term vanishes because $D[p,p] = 0$ is a minimum (any statistical distance is non-negative). Here, $H_D(\boldsymbol{\gamma}) = H_D(p_{\boldsymbol{\gamma}})$ is the Hessian matrix with entries

$$[H_D(\boldsymbol{\gamma})]_{i,j} = \frac{\partial^2}{\partial\phi_i\partial\phi_j} D[p_{\boldsymbol{\gamma}}, p_{\boldsymbol{\phi}}]\bigg|_{\boldsymbol{\phi}=\boldsymbol{\gamma}}. \tag{5.14}$$

The components of the Hessian matrix measure how susceptible the probability distribution $p_{\boldsymbol{\gamma}}$ is to small changes in the underlying coordinates $\boldsymbol{\gamma}$, where the resulting deviations are measured by the statistical distance D. One can show that any Hessian induced by a monotonic statistical distance (such as f-divergences) must also be monotonic.

Proposition (Monotonicity of Hessian)
Consider a stochastic map $\mathcal{S} : \mathcal{M} \to \mathcal{M}$ such that $p' = \mathcal{S}p$, where

$$p'(\boldsymbol{z}) = \sum_{\boldsymbol{x}\in\mathcal{X}} W(\boldsymbol{z}|\boldsymbol{x})p(\boldsymbol{x}) \tag{5.15}$$

with W being a left-stochastic transition matrix. The Hessian of any sufficiently smooth statistical distance D that is monotonic under such maps must also be monotonic.

Proof

Because D is monotonic, it satisfies the data-processing inequality. Thus, we have $D[p'_{\boldsymbol{\gamma}}, p'_{\boldsymbol{\gamma}+\boldsymbol{\Delta\gamma}}] \leq D[p_{\boldsymbol{\gamma}}, p_{\boldsymbol{\gamma}+\boldsymbol{\Delta\gamma}}]$. Expanding the statistical distance to second order according to Equation (5.13), we have that $H_D(p_{\boldsymbol{\gamma}}) \geq H_D(\mathcal{S}p_{\boldsymbol{\gamma}})$ up to $\mathcal{O}(\boldsymbol{\Delta\gamma}^3)$. Letting $\|\boldsymbol{\Delta\gamma}\| \to 0$ concludes the proof.

Moreover, if D is an f-divergence with $f'(1) = 0$, the Hessian can be shown to be proportional to the (classical) Fisher information matrix $\mathcal{F}(\boldsymbol{\gamma}) = \mathcal{F}(p_{\boldsymbol{\gamma}})$, where

$$\mathcal{F}_{i,j}(\boldsymbol{\gamma}) = \mathcal{F}_{i,j}(p_{\boldsymbol{\gamma}}) = \mathbb{E}_{\boldsymbol{x}\sim p_{\boldsymbol{\gamma}}}\left[\left(\frac{\partial \ln(p_{\boldsymbol{\gamma}})}{\partial\gamma_i}\right)\left(\frac{\partial \ln(p_{\boldsymbol{\gamma}})}{\partial\gamma_j}\right)\right]. \tag{5.16}$$

Proposition (Relation between Hessian and Fisher information matrix)
The Hessian matrix of any f-divergence D_f with f being twice-differentiable is given by $H_{D_f}(\boldsymbol{\gamma}) = f''(1)\mathcal{F}(\boldsymbol{\gamma})$.

Proof

$$
\begin{aligned}
[H_{D_f}(\boldsymbol{\gamma})]_{i,j} &= \frac{\partial^2}{\partial\phi_i\partial\phi_j} D_f[p_{\boldsymbol{\gamma}}, p_{\boldsymbol{\phi}}]\bigg|_{\boldsymbol{\phi}=\boldsymbol{\gamma}} \\
&= \sum_{\boldsymbol{x}\in\mathcal{X}} \frac{1}{p_{\boldsymbol{\gamma}}(\boldsymbol{x})} f''(1) \frac{\partial p_{\boldsymbol{\gamma}}(\boldsymbol{x})}{\partial\gamma_i} \frac{\partial p_{\boldsymbol{\gamma}}(\boldsymbol{x})}{\partial\gamma_j} \\
&= f''(1)\mathcal{F}_{i,j}(\boldsymbol{\gamma}), \qquad (5.17)
\end{aligned}
$$

where we used the fact that $f(1) = f'(1) = 0$. If $f'(1) \neq 0$, we may use our freedom in the choice of generating function to ensure otherwise. That is, we replace $f \mapsto g$, where $g(x) = f(x) - f'(1)(x-1)$ retaining $D_f = D_g$.

Note that the scalar version of the Fisher information matrix is referred to as Fisher information.

Combining the above findings, namely that f-divergences are monotonic, that the Hessian corresponding to any monotonic statistical distance must also be monotonic, and that the Hessian of any f-divergence is proportional to the Fisher information matrix, we have that the Fisher information matrix is also monotonic, i.e.,

$$\mathcal{F}(\mathcal{S}p_{\boldsymbol{\gamma}}) \leq \mathcal{F}(p_{\boldsymbol{\gamma}}). \qquad (5.18)$$

This also implies that the Fisher information remains invariant under the mapping $S : \mathcal{X} \to \mathcal{Z}$ if $\boldsymbol{z} = S(\boldsymbol{x})$ is a sufficient statistic.

Chentsov's theorem [Chentsov, 1978; Fujiwara, 2022] extends the above argument to all monotonic metrics. It states that all Riemannian metrics defined on a given statistical manifold that are monotonic correspond to the Fisher metric (i.e., the Fisher information matrix) up to a multiplicative constant. Note that H_{D_f} can be interpreted as a Riemannian metric on the statistical manifold $\mathcal{M}$. Thus, Chentov's theorem covers the above discussion as a special case. Monotonicity is an important property for any sensible statistical distance. As such, Chentsov's theorem effectively singles out the Fisher information matrix as the only natural metric on the statistical manifold, justifying why many properties of statistical models should be describable in terms of the Fisher information matrix [Dowty, 2018]. In the context of detecting phase transitions from data, the above discussion provides some intuition on why many approaches that rely on measuring changes in the underlying probability distributions may ultimately be related to the Fisher information. We will prove such a relation for SL, LBC, and PBM in Chapter 6. In particular, while one may *a priori* choose distinct f-divergences for gauging such changes, as long as the distributions are sufficiently close in parameter space, any such choice reduces to the Fisher information.

5.4 Hypothesis testing

For the second part of this chapter, let us discuss how f-divergences and the Fisher information relate to statistical inference tasks. We start with the task of hypothesis testing. Given the outcomes $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ of n independent rounds of an experiment, one needs to decide which distribution from a set of m possible choices $\{p_1, p_2, \ldots, p_m\}$ is most likely to describe the experiment. In the following, we are concerned with binary hypothesis testing, i.e., distinguishing between two distinct distributions ($m = 2$) given a single measurement outcome ($n = 1$). Moreover, we treat false positives and false negatives equally. This special case is referred to as *single-shot symmetric binary hypothesis testing*. In this task, one is interested in minimizing the average error probability

$$p_{\mathrm{err}} = \frac{1}{2}P(q|p) + \frac{1}{2}P(p|q), \tag{5.19}$$

where $P(q|p)$ corresponds to the probability of selecting the probability distribution q while the data actually came from p [and vice versa for $P(p|q)$]. The factors of $1/2$ in Equation (5.19) reflect the fact that, *a priori*, a sample is equally likely to be drawn from p or q.

The goal is to find a decision function $\hat{y} : \mathcal{X} \to \{0, 1\}$, where $\hat{y}(\boldsymbol{x}) = 0$ corresponds to the conclusion that $\boldsymbol{x}$ has been drawn from p and $\hat{y}(\boldsymbol{x}) = 1$ corresponds to the conclusion that $\boldsymbol{x}$ has been drawn from q instead. The optimal inference strategy can be found by minimizing the average error probability p_{err} in Equation (5.19). In the scenario we consider, the optimal strategy, also called *Neyman-Pearson* strategy, corresponds to guessing p if $p(\boldsymbol{x}) \geq q(\boldsymbol{x})$ and guessing q otherwise. Under this strategy, the error probabilities can be expressed as

$$P(p|q) = \sum_{\substack{\boldsymbol{x}\in\mathcal{X} \\ p(\boldsymbol{x})\geq q(\boldsymbol{x})}} q(\boldsymbol{x}) \quad \text{and} \quad P(q|p) = \sum_{\substack{\boldsymbol{x}\in\mathcal{X} \\ q(\boldsymbol{x})> p(\boldsymbol{x})}} p(\boldsymbol{x}). \tag{5.20}$$

Proposition (Error corresponding to Neyman-Pearson strategy)
When making predictions according to the Neyman-Pearson strategy, the (optimal) average error probability is equal to

$$p_{\mathrm{err}}^{\mathrm{opt}} = \frac{1}{2}\left(1 - \frac{1}{2}\sum_{\boldsymbol{x}\in\mathcal{X}} |p(\boldsymbol{x}) - q(\boldsymbol{x})|\right). \tag{5.21}$$

Proof

$$\begin{aligned}
p_{\mathrm{err}}^{\mathrm{opt}} &= \frac{1}{2}\left(1 - \frac{1}{2}\sum_{\boldsymbol{x}\in\mathcal{X}} |p(\boldsymbol{x}) - q(\boldsymbol{x})|\right) \\
&= \frac{1}{2} - \frac{1}{4}\sum_{\substack{\boldsymbol{x}\in\mathcal{X} \\ p(\boldsymbol{x})\geq q(\boldsymbol{x})}} [p(\boldsymbol{x}) - q(\boldsymbol{x})] - \frac{1}{4}\sum_{\substack{\boldsymbol{x}\in\mathcal{X} \\ q(\boldsymbol{x})> p(\boldsymbol{x})}} [q(\boldsymbol{x}) - p(\boldsymbol{x})] \\
&= \frac{1}{2} - \frac{1}{4}\left[\sum_{\substack{\boldsymbol{x}\in\mathcal{X} \\ p(\boldsymbol{x})\geq q(\boldsymbol{x})}} p(\boldsymbol{x}) + \sum_{\substack{\boldsymbol{x}\in\mathcal{X} \\ q(\boldsymbol{x})> p(\boldsymbol{x})}} q(\boldsymbol{x}) - \sum_{\substack{\boldsymbol{x}\in\mathcal{X} \\ p(\boldsymbol{x})\geq q(\boldsymbol{x})}} q(\boldsymbol{x}) - \sum_{\substack{\boldsymbol{x}\in\mathcal{X} \\ q(\boldsymbol{x})> p(\boldsymbol{x})}} p(\boldsymbol{x})\right].
\end{aligned} \tag{5.22}$$

Using $1 = \sum_{\boldsymbol{x}\in\mathcal{X}} p(\boldsymbol{x}) = \sum_{\substack{\boldsymbol{x}\in\mathcal{X} \\ p(\boldsymbol{x})\geq q(\boldsymbol{x})}} p(\boldsymbol{x}) + \sum_{\substack{\boldsymbol{x}\in\mathcal{X} \\ q(\boldsymbol{x})> p(\boldsymbol{x})}} p(\boldsymbol{x})$ (and similarly for q) to rewrite the first two terms, we have

$$p_{\mathrm{err}}^{\mathrm{opt}} = \frac{1}{2}\sum_{\substack{\boldsymbol{x}\in\mathcal{X} \\ p(\boldsymbol{x})\geq q(\boldsymbol{x})}} q(\boldsymbol{x}) + \frac{1}{2}\sum_{\substack{\boldsymbol{x}\in\mathcal{X} \\ q(\boldsymbol{x})> p(\boldsymbol{x})}} p(\boldsymbol{x}), \tag{5.23}$$

corresponding to the error rate under the optimal strategy obtained by plugging in Equation (5.20) into Equation (5.19).

Using the definition of the TV distance in Equation (5.2), the minimal average error probability [Equation (5.21)] can be expressed as

$$p_{\mathrm{err}}^{\mathrm{opt}} = \frac{1}{2}\Big(1 - D_{\mathrm{TV}}[p,q]\Big), \tag{5.24}$$

giving the TV distance operational meaning. That is, we can express the TV distance as

$$D_{\mathrm{TV}}[p,q] = 1 - 2p_{\mathrm{err}}^{\mathrm{opt}}. \tag{5.25}$$

Figure 5.1 gives a simple graphical proof of this relation and provides some intuition on this result. Recall that in Chapter 4 (Section 4.5), we have explicitly shown that the optimal predictive model in LBC makes predictions according to the Neyman-Pearson strategy and achieves the optimal error rate in Equation (5.24).

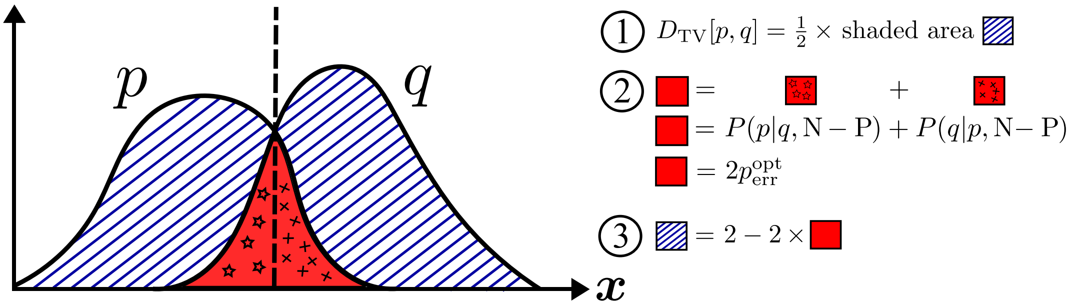
p
q
$\boldsymbol{x}$
1 $D_{\mathrm{TV}}[p,q] = \frac{1}{2} \times$ shaded area ▨
2 🟥 = 🟥 + 🟥
🟥 $= P(p|q, \mathrm{N-P}) + P(q|p, \mathrm{N-P})$
🟥 $= 2p_{\mathrm{err}}^{\mathrm{opt}}$
3 ▨ $= 2 - 2 \times$ 🟥

FIGURE 5.1: Schematic illustration of the relation between the TV distance and the optimal error probability in single-shot symmetric binary hypothesis testing. The key proof steps 1-3 are outlined on the right (boxes are placeholders for the size of certain areas). Statement 1 follows from the definition of the TV distance in Equation (5.2). Statement 2 follows from Equations (5.19) and (5.20) where N – P denotes the Neyman-Pearson strategy. Statement 3 follows from the fact that both p and q are normalized probability distributions that integrate to one. Combining the three statements, we can express the TV distance in terms of the optimal error rate $p_{\mathrm{err}}^{\mathrm{opt}}$, arriving at Equation (5.25).

5.5 Parameter estimation

Finally, let us consider the statistical inference task of parameter estimation which underlies PBM. Here, one tries to estimate an unknown set of parameters $\boldsymbol{\gamma}$ based on independent measurements $\boldsymbol{x}$ distributed according to the probability distribution $p_{\boldsymbol{\gamma}}(\boldsymbol{x})$. This is done by calculating an estimator of $\boldsymbol{\gamma}$ denoted $\hat{\boldsymbol{\gamma}}(\boldsymbol{x})$ given a measurement $\boldsymbol{x}$. The bias of the estimator can be calculated as

$$\boldsymbol{b}(\boldsymbol{\gamma}) = \mathbb{E}_{\boldsymbol{x}\sim p_{\boldsymbol{\gamma}}}[\hat{\boldsymbol{\gamma}}(\boldsymbol{x}) - \boldsymbol{\gamma}] = \boldsymbol{\psi}(\boldsymbol{\gamma}) - \boldsymbol{\gamma}. \tag{5.26}$$

An estimator is called *unbiased* if $\boldsymbol{b}(\boldsymbol{\gamma}) = 0$ for all $\boldsymbol{\gamma}$. The *Cramér-Rao bound* (also known as information inequality) [Cramér, 1946; Rao, 1945] states that

$$\mathrm{Cov}(\boldsymbol{\gamma}) \geq \left(\frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{\gamma}}\right)^T \mathcal{F}(\boldsymbol{\gamma})^{-1} \left(\frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{\gamma}}\right), \tag{5.27}$$

where $\mathrm{Cov}(\boldsymbol{\gamma})$ is the covariance matrix of the estimator evaluated at $\boldsymbol{\gamma}$ and $\left(\frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{\gamma}}\right)$ denotes the Jacobian matrix of $\boldsymbol{\psi}$ with matrix elements $\left(\frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{\gamma}}\right)_{i,j} = \partial \psi_i(\boldsymbol{\gamma})/\partial \gamma_j$. Here, we have assumed that $\mathcal{F}(\boldsymbol{\gamma})$ is nonsingular. A proof of the Cramér-Rao bound [Equation (5.27)] can be found in [Bickel and Doksum, 2015] (pp. 179–188).

For an unbiased estimator, i.e., $\boldsymbol{\psi}(\boldsymbol{\gamma}) = \boldsymbol{\gamma}$, the Cramér-Rao bound reduces to

$$\mathrm{Cov}(\boldsymbol{\gamma}) \geq \mathcal{F}(\boldsymbol{\gamma})^{-1}. \tag{5.28}$$

Moreover, in the special scalar case, the Cramér-Rao bound reads

$$\sigma^2(\gamma) \geq \frac{[1 + b'(\gamma)]^2}{\mathcal{F}(\gamma)} = \frac{\psi'(\gamma)^2}{\mathcal{F}(\gamma)}, \tag{5.29}$$

where $b'(\gamma) = \partial b/\partial \gamma$ and $\mathcal{F}$ the Fisher information associated with γ.

The results and figures presented in this chapter have been in parts published in [Arnold *et al.*, 2023a].

Chapter 6

On Connections to the Fisher Information

The results presented in this chapter are based on the following preprint:

Machine learning phase transitions: Connections to the Fisher information,
J. Arnold, F. Holtorf, N. Lörch, and F. Schäfer,
arXiv:2311.10710 (2023).

6.1 Motivation

Traditionally, critical phenomena have been studied by relying on the Ginzburg-Landau-Wilson paradigm which is based on concepts such as symmetry breaking and local order parameters [Goldenfeld, 2018]. However, identifying the proper order parameters of systems whose symmetry-breaking patterns are unknown is difficult. Information-theoretic quantities are particularly promising for studying phase transitions without relying on this traditional paradigm. Such quantities are universal and their computation does not require a detailed analysis of the system's physics, such as its order parameters.

In this context, the classical Fisher information [Fisher, 1922] and its quantum counterpart [Helstrom, 1967] have been extensively studied as universal indicators of phase transitions, i.e., as quantities whose maxima are signatures of critical points. The Fisher information is a generalized susceptibility that measures the sensitivity of the system with respect to a tuning parameter. In the case of classical equilibrium systems, it measures fluctuations in the system's collective variables and is proportional to well-known response functions, such as the magnetic susceptibility or the heat capacity [Prokopenko *et al.*, 2011]. Similarly, the quantum Fisher information reduces to the fidelity susceptibility [You *et al.*, 2007; Gu, 2010; Liu *et al.*, 2014], i.e., the leading-order response of the fidelity between quantum states to parameter fluctuations. The fidelity susceptibility has been shown to detect symmetry-breaking [Campos Venuti and Zanardi, 2007; Zanardi *et al.*, 2007], topological [Abasto *et al.*, 2008; Yang *et al.*, 2008; Garnerone *et al.*, 2009], and BKT-type [Yang, 2007; Wang *et al.*, 2010] quantum phase transitions. Moreover, the quantum Fisher information has been used to investigate finite-temperature transitions as well as non-equilibrium phenomena, such as dissipative [Banchi *et al.*, 2014; Rota *et al.*, 2017; Heugel *et al.*, 2019], dynamical [Macieszczak *et al.*, 2016; Guan and Lewis-Swan, 2021], or excited-state [Zhou *et al.*, 2023] phase transitions.

We have devoted this thesis to the study of another alternative paradigm for studying phase transitions: machine learning [Carleo *et al.*, 2019; Carrasquilla, 2020; Dawid *et al.*, 2022]. Interestingly, the appeal of ML methods is akin to the one of information-theoretic approaches. ML methods are generic and can be used to characterize a system using minimal explicit knowledge of its underlying physics. In this chapter, we root these methods in information theory. In particular, we prove that the indicators of phase transitions obtained in SL, LBC, and PBM (as they are defined in Chapter 4) are lower bounds to the square root of the system's Fisher information with respect to the tuning parameter. These bounds reveal a strong link between the ML and information-theoretic paradigm for detecting phase transitions. We numerically demonstrate the quality of these underapproximations for phase transitions in classical and quantum systems. Building upon previous results on the Fisher information in the context of statistical and quantum physics, this yields insights into the operation of ML methods for detecting phase transitions. These insights help to understand the limitations and strengths of such methods when applied to different classes of phase transitions, suggest the methods' usage as algorithms for approximating the Fisher information in more general settings, and improve their performance via modifications motivated by this information-theoretic correspondence.

6.2 Recap: How to make a machine detect phase transitions

Let us start with recapping how the three ML methods for detecting phase transitions from data – SL, LBC, and PBM – are currently defined based on the modifications we suggested in Chapter 4. For improved clarity of the presentation, let us first restrict ourselves to the simple case where the physical system is characterized by a single tunable parameter γ along which a phase transition occurs. We will discuss how to analyze the methods in their full generality in Section 6.3.1.

Supervised learning

In SL, we assume we know a set of points Γ_0 and Γ_1 lying within each of the two phases, phase 0 and phase 1, where $\forall\gamma \in \Gamma_0, \gamma' \in \Gamma_1 : \gamma < \gamma'$. Then, for $y \in \mathcal{Y} = \{0,1\}$ we can assign to all the samples $\boldsymbol{x}$ drawn at points in Γ_y the label $y(\boldsymbol{x}) = y$. Based on this, we train a classifier $\hat{y} : \mathcal{X} \to [0,1]$ to assign the correct phase to a given sample $\boldsymbol{x}$. Intuitively, the mean prediction $\hat{y}(\gamma) = \mathbb{E}_{\boldsymbol{x}\sim P(\cdot|\gamma)}\left[\hat{y}(\boldsymbol{x})\right]$ will change most at the critical point. This is captured by the following scalar indicator of phase transitions

$$I_{\mathrm{SL}}(\gamma) = \left|\frac{\partial \hat{y}(\gamma)}{\partial \gamma}\right|, \tag{6.1}$$

whose maximum is expected to occur at the critical point.

Learning by confusion

In LBC, at each sampled point $\gamma \in \Gamma$, we divide the parameter space into two sets of points, $\Gamma_0(\gamma)$ and $\Gamma_1(\gamma)$, each comprised of the l sampled points γ' closest to γ with $\gamma' \leq \gamma$ and $\gamma' > \gamma$, respectively. Each parameter point γ defines a bipartition and, in turn, a classification task. Given a predictive model $\hat{y} : \mathcal{X} \to [0,1]$ trained to perform this task, its error rate is defined as

$$p_{\mathrm{err}}(\gamma) = \frac{1}{2} \sum_{y\in\{0,1\}} \frac{1}{|\Gamma_y|} \sum_{\gamma'\in\Gamma_y} \mathbb{E}_{\boldsymbol{x}\sim P(\cdot|\gamma')}\left[\mathrm{err}(\boldsymbol{x}, y)\right], \tag{6.2}$$

where $\mathrm{err}(\boldsymbol{x}, y)$ is zero if the sample is classified correctly and one otherwise. Intuitively, $p_{\mathrm{err}}(\gamma)$ is lowest at a phase boundary where the data is partitioned according to its phase. Thus, in LBC, critical points can be detected as local maxima in the indicator

$$I_{\mathrm{LBC}}(\gamma) = 1 - 2p_{\mathrm{err}}(\gamma). \tag{6.3}$$

Prediction-based method

At the core of PBM lies a predictive model $\hat{\gamma}$ that estimates the parameter γ at which a given sample $\boldsymbol{x}$ was drawn. Intuitively, the mean predicted value of the tuning parameter $\hat{\gamma}(\gamma) = \mathbb{E}_{\boldsymbol{x}\sim P(\cdot|\gamma)}\left[\hat{\gamma}(\boldsymbol{x})\right]$ is expected to be most sensitive at phase boundaries. The following indicator captures this susceptibility

$$I_{\mathrm{PBM}}(\gamma) = \frac{\partial \hat{\gamma}(\gamma)/\partial\gamma}{\mathrm{std}(\gamma)}, \tag{6.4}$$

where $\mathrm{std}(\gamma) = \sqrt{\mathbb{E}_{\boldsymbol{x}\sim P(\cdot|\gamma)}\left[\hat{\gamma}(\boldsymbol{x})^2\right] - \hat{\gamma}(\gamma)^2}$.

How to compute indicators of phase transitions in practice

We have formulated the problem of detecting a phase transition as the computation of an indicator function [Equations (6.1), (6.3), and (6.4)]. This computation involves solving a classification or regression task, i.e., finding a suitable predictive model. This model can be constructed in a data-driven way given a set of samples $\mathcal{D}_\gamma$ drawn from $P(\cdot|\gamma)$ for each $\gamma \in \Gamma$. Typically, a discriminative approach is chosen in which the predictive model ($\hat{y}$ or $\hat{\gamma}$) is an NN whose parameters $\boldsymbol{\theta}$ are optimized in a supervised fashion via the minimization of a loss function $\mathcal{L}(\boldsymbol{\theta})$. For SL and LBC dealing with classification tasks, a typical choice is an unbiased binary cross-entropy loss

$$\mathcal{L}(\boldsymbol{\theta}) = -\frac{1}{2}\sum_{y\in\{0,1\}}\frac{1}{|\mathcal{D}_y|}\sum_{\boldsymbol{x}\in\mathcal{D}_y}\Big(y\ln\left[\hat{y}_{\boldsymbol{\theta}}(\boldsymbol{x})\right] + (1-y)\ln\left[1-\hat{y}_{\boldsymbol{\theta}}(\boldsymbol{x})\right]\Big), \tag{6.5}$$

where $\mathcal{D}_y$ is composed of all sets of samples $\mathcal{D}_\gamma$ with $\gamma \in \Gamma_y$. For the regression task in PBM, a mean squared error loss is used

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{|\Gamma|}\sum_{\gamma\in\Gamma}\frac{1}{|\mathcal{D}_\gamma|}\sum_{\boldsymbol{x}\in\mathcal{D}_\gamma}\left[\gamma - \hat{\gamma}_{\boldsymbol{\theta}}(\boldsymbol{x})\right]^2. \tag{6.6}$$

Given a predictive model, an estimate of the corresponding indicator can be obtained by replacing expected values with sample means.

6.3 Relating indicators to the Fisher information

In the following, we are going to establish a connection between the three aforementioned indicators of phase transitions [Equations (6.1), (6.3), and (6.4)] and the Fisher information

$$\mathcal{F}(\gamma) = \mathbb{E}_{\boldsymbol{x}\sim P(\cdot|\gamma)}\left[\left(\frac{\partial \ln\left[P(\boldsymbol{x}|\gamma)\right]}{\partial\gamma}\right)^2\right], \tag{6.7}$$

which quantifies the amount of information that the random variable $\boldsymbol{x}$ carries about the parameter γ characterizing its distribution $P(\cdot|\gamma)$. The intuitive explanation for the existence of such a relationship lies in the fact that all three indicators gauge changes in the underlying probability distributions as a function of the tuning parameter. We have first observed this in Chapter 3 after deriving the optimal predictions and indicators of these methods.

Supervised learning

The indicator of SL [Equation (6.1)] can be written as

$$I_{\mathrm{SL}}(\gamma) = \left|\mathbb{E}_{\boldsymbol{x}\sim P(\cdot|\gamma)}\left[\hat{y}(\boldsymbol{x})\frac{\partial \ln\left[P(\boldsymbol{x}|\gamma)\right]}{\partial\gamma}\right]\right| \tag{6.8}$$

using the log-derivative trick. By the Cauchy-Schwarz inequality, the indicator $I_{\mathrm{SL}}(\gamma)$ is maximal if and only if $\hat{y}$ is perfectly correlated with the score, i.e.,

$$\frac{\partial \ln\left[P(\boldsymbol{x}|\gamma)\right]}{\partial\gamma} = \pm\sqrt{\mathcal{F}(\gamma)}\left(\frac{\hat{y}(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{x}\sim P(\cdot|\gamma)}\left[\hat{y}(\boldsymbol{x})\right]}{\mathrm{std}(\gamma)}\right), \tag{6.9}$$

where $\mathrm{std}(\gamma)$ is the standard deviation of $\hat{y}$ at γ and we have used the fact that the score $\partial\ln\left[P(\cdot|\gamma)\right]/\partial\gamma$ has zero mean and the mean of its square corresponds to the Fisher information [cf. Equation (6.7)]. Because samples with, e.g., a negative score, are typically predominantly found in phase 0 compared to phase 1, the correlation of $\hat{y}$ with the score (and thus I_{SL} itself) is expected to increase with increasing quality of the predictive model. Based on Equation (6.9), we have

$$I_{\mathrm{SL}}(\gamma) \leq \mathrm{std}(\gamma)\sqrt{\mathcal{F}(\gamma)} \leq \sqrt{\mathcal{F}(\gamma)}, \tag{6.10}$$

where the second inequality follows from the fact that $\forall \boldsymbol{x}\in\mathcal{X}\ |\hat{y}(\boldsymbol{x})|\leq 1$ by construction for any valid predictive model. Equation (6.10) further suggests an improvement of the SL method by modifying its indicator as $I_{\mathrm{SL}}(\gamma)\mapsto I_{\mathrm{SL}}(\gamma)/\mathrm{std}(\gamma)$ to obtain a tighter bound on the square root of the Fisher information. We will analyze this modified indicator of SL in more detail in Section 6.5.1.

Learning by confusion

LBC involves the statistical task of single-shot symmetric binary hypothesis testing (recall Section 5.4 from Chapter 5): based on a single measurement outcome $\boldsymbol{x}\in\mathcal{X}$, determine which of two probability distributions P and Q is more likely to describe the experiment and avoid both false positives and negatives equally. As we have derived in Section 5.4, the optimal error probability is given by $p_{\mathrm{err}}^{\mathrm{opt}} = \frac{1}{2}\left(1 - D_{\mathrm{TV}}\left[P,Q\right]\right)$, where D_{TV} denotes the total variation distance $D_{\mathrm{TV}}\left[P,Q\right] = \frac{1}{2}\sum_{\boldsymbol{x}\in\mathcal{X}}|P(\boldsymbol{x}) - Q(\boldsymbol{x})|$. The indicator value corresponding to a given bipartition can thus be upper-bounded as

$$I_{\mathrm{LBC}}(\gamma) \leq 1 - 2p_{\mathrm{err}}^{\mathrm{opt}}(\gamma) = I_{\mathrm{LBC}}^{\mathrm{opt}}(\gamma) = D_{\mathrm{TV}}\left[P_0,P_1\right], \tag{6.11}$$

where the total variation distance is measured between the probability distributions underlying the two partitions $P_y = \frac{1}{|\Gamma_y|}\sum_{\gamma'\in\Gamma_y}P(\cdot|\gamma')$, $y\in\{0,1\}$. This bound holds for any valid predictive model $\hat{y}:\mathcal{X}\to[0,1]$. In Section 4.5, we have shown that $p_{\mathrm{err}}^{\mathrm{opt}}$ is achieved by a predictive model that minimizes the loss in Equation (6.5) in the infinite data limit. Thus, we find that LBC relies on a variational lower bound of the TV distance, which improves during training, i.e., as p_{err} decreases.

Consider now the behavior of the TV distance for probability distributions separated by a small distance $\Delta\gamma$ in parameter space, i.e., $P(\cdot|\gamma)$ and $P(\cdot|\gamma+\Delta\gamma)$

$$D_{\mathrm{TV}}\left[P(\cdot|\gamma), P(\cdot|\gamma+\Delta\gamma)\right] = \frac{1}{2}\sum_{\boldsymbol{x}\in\mathcal{X}}\left|\frac{\partial P(\boldsymbol{x}|\gamma)}{\partial\gamma}\right|\Delta\gamma + \mathcal{O}(\Delta\gamma^2). \tag{6.12}$$

Using the Cauchy-Schwarz inequality, we have

$$D_{\mathrm{TV}}\left[P(\cdot|\gamma), P(\cdot|\gamma+\Delta\gamma)\right] \leq \frac{1}{2}\sqrt{\mathcal{F}(\gamma)}\Delta\gamma + \mathcal{O}(\Delta\gamma^2). \tag{6.13}$$

Plugging into Equation (6.11), we obtain

$$I_{\mathrm{LBC}}(\gamma) \leq I_{\mathrm{LBC}}^{\mathrm{opt}}(\gamma) \leq \frac{1}{2}\sqrt{\mathcal{F}(\gamma)}\Delta\gamma + \mathcal{O}(\Delta\gamma^2). \tag{6.14}$$

This is realized in a scenario where $l = 1$ and the probability distributions P_0 and P_1 corresponding to the two singletons $\Gamma_0 = \{\gamma\}$ and $\Gamma_1 = \{\gamma + \Delta\gamma\}$ are separated by a distance $\Delta\gamma$. As $\Delta\gamma \to 0$, the indicator of LBC (rescaled by $2/\Delta\gamma$) serves as a lower bound to the square root of the Fisher information. This further motivates the modifications we performed to LBC in Chapter 4, including our choice to set $l = 1$.

Prediction-based method

PBM is based on parameter estimation. The Cramér–Rao bound is a well-known lower bound on the variance of an estimator of a deterministic (fixed, though unknown) parameter γ (recall Section 5.5 in Chapter 5):

$$\mathrm{std}^2(\gamma) \geq \frac{\left(1+\partial b(\gamma)/\partial\gamma\right)^2}{\mathcal{F}(\gamma)} = \frac{\left(\partial\hat{\gamma}(\gamma)/\partial\gamma\right)^2}{\mathcal{F}(\gamma)}. \tag{6.15}$$

Here, $b(\gamma) = \mathbb{E}_{\boldsymbol{x}\sim P(\cdot|\gamma)}[\hat{\gamma}(\boldsymbol{x}) - \gamma]$ is the bias of the estimator. Rearranging the Cramér–Rao bound and plugging in the definition of the indicator of PBM from Equation (6.4), we have

$$I_{\mathrm{PBM}}(\gamma) = \frac{\partial\hat{\gamma}(\gamma)/\partial\gamma}{\mathrm{std}(\gamma)} \leq \sqrt{\mathcal{F}(\gamma)}. \tag{6.16}$$

Interestingly, both $\partial\hat{\gamma}(\gamma)/\partial\gamma$ [Schäfer and Lörch, 2019; Greplova *et al.*, 2020; Arnold *et al.*, 2021; Arnold and Schäfer, 2022b] and $\mathrm{std}(\gamma)$ [Guo and He, 2023; Guo *et al.*, 2023] have been used separately as indicators of phase transitions, i.e., quantities whose local maxima and minima, respectively, indicate critical points. The connection to the Fisher information established in Equation (6.16) further justifies using their ratio as an indicator of phase transitions – a modification that we have found to be fruitful in Chapter 4. Note that minimum mean-squared error estimation [Alves and Landi, 2022] is a widely used approach for solving parameter estimation tasks. As such, the bound in Equation (6.16) is generally expected to improve during training using the loss function in Equation (6.6). However, saturating the Cramér–Rao bound [and thus saturating the bound in Equation (6.16)] is generally not possible at the single-copy level, i.e., when the estimator is a function of a single sample.

6.3.1 Generalization to higher-dimensional parameter spaces

The relations between the three ML indicators and the Fisher information derived above constitute the central result of this chapter. We find that all three indicators (up to rescaling) form lower bounds to the square root of the system's Fisher information with respect to the tuning parameter γ.

In this section, we generalize the results derived above to parameter spaces of arbitrary dimension which possibly feature multiple phase transitions. We prove that the corresponding indicators are lower bounds to the trace of the Fisher information matrix.

Proof

Supervised learning.—Following the discussion in Section 4.2.1, in SL, we assume knowledge of the number of distinct phases, G, and their rough location in parameter space. The corresponding points in parameter space are assigned distinct labels $y \in \mathcal{Y} = \{0, 1, \dots, G-1\}$. The indicator is then given by

$$I_{\mathrm{SL}}(\boldsymbol{\gamma}) = \frac{1}{|\mathcal{Y}|} \sum_{y \in \mathcal{Y}} \sqrt{\sum_{i=1}^{d} \left(\frac{\partial \tilde{P}(y|\boldsymbol{\gamma})}{\partial \gamma_i} \right)^2}, \tag{6.17}$$

where $\tilde{P}(y|\boldsymbol{\gamma}) = \mathbb{E}_{\boldsymbol{x} \sim P(\cdot|\boldsymbol{\gamma})} \left[\tilde{P}(y|\boldsymbol{x}) \right]$ with $\tilde{P}(y|\boldsymbol{x})$ denoting the (estimated) probability that sample $\boldsymbol{x}$ carries label y. Following the derivation above for the conditional probability of each label $\tilde{P}(y|\boldsymbol{\gamma})$ separately, we have

$$I_{\mathrm{SL}}(\boldsymbol{\gamma}) \leq \frac{1}{|\mathcal{Y}|} \sum_{y \in \mathcal{Y}} \mathrm{std}_y(\boldsymbol{\gamma}) \sqrt{\sum_{i=1}^{d} \mathcal{F}_{i,i}(\boldsymbol{\gamma})} = \sqrt{\mathrm{tr}\left[\mathcal{F}(\boldsymbol{\gamma})\right]}, \tag{6.18}$$

where $\mathcal{F}_{i,i}(\boldsymbol{\gamma})$ is the ith diagonal element of the Fisher information matrix and $\mathrm{std}_y(\boldsymbol{\gamma})$ is the standard deviation of $\tilde{P}(y|\boldsymbol{x})$ at $\boldsymbol{\gamma}$.

Learning by confusion.—In Chapter 4, we generalized LBC by dividing the parameter space along each direction at each sampled point $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \dots, \gamma_d) \in \Gamma$. For a given direction $1 \leq i \leq d$, this yields two sets, $\Gamma_0^{(i)}(\boldsymbol{\gamma})$ and $\Gamma_1^{(i)}(\boldsymbol{\gamma})$, each comprised of the l points closest to $\boldsymbol{\gamma}$ in a first and second part of the split parameter space, respectively. Based on these sets, an indicator component is computed according to Equations (6.2) and (6.3), $I_{\mathrm{LBC}}^{(i)}(\boldsymbol{\gamma}) = 1 - 2p_{\mathrm{err}}^{(i)}(\boldsymbol{\gamma})$. The overall indicator is

$$I_{\mathrm{LBC}}(\boldsymbol{\gamma}) = \sqrt{\sum_{i=1}^{d} \left[I_{\mathrm{LBC}}^{(i)}(\boldsymbol{\gamma}) \right]^2}. \tag{6.19}$$

Following the proof of the one-dimensional case, for $l = 1$ each indicator component is bounded as

$$I_{\mathrm{LBC}}^{(i)}(\boldsymbol{\gamma}) \leq I_{\mathrm{LBC}}^{(i),\mathrm{opt}}(\boldsymbol{\gamma}) \leq \frac{1}{2} \sqrt{\mathcal{F}_{i,i}(\boldsymbol{\gamma})} \Delta\gamma_i + \mathcal{O}(\Delta\gamma_i^2). \tag{6.20}$$

Thus,

$$I_{\mathrm{LBC}}(\boldsymbol{\gamma}) \leq I_{\mathrm{LBC}}^{\mathrm{opt}}(\boldsymbol{\gamma}) \leq \sqrt{\sum_{i=1}^{d}\left(I_{\mathrm{LBC}}^{(i),\mathrm{opt}}(\boldsymbol{\gamma})\right)^2} \leq \frac{1}{2}\sqrt{\sum_{i=1}^{d}\mathcal{F}_{i,i}(\boldsymbol{\gamma})\Delta\gamma_i^2}+\sum_{i=1}^{d}\mathcal{O}(\Delta\gamma_i^2). \tag{6.21}$$

Assuming $\forall i\ \Delta\gamma_i = \Delta\gamma$, we have

$$I_{\mathrm{LBC}}(\boldsymbol{\gamma}) \leq I_{\mathrm{LBC}}^{\mathrm{opt}}(\boldsymbol{\gamma}) \leq \frac{1}{2}\Delta\gamma\sqrt{\mathrm{tr}\left[\mathcal{F}(\boldsymbol{\gamma})\right]}+\mathcal{O}(\Delta\gamma^2). \tag{6.22}$$

Thus, in the limit $\Delta\gamma \to 0$, the quantity $2I_{\mathrm{LBC}}(\boldsymbol{\gamma})/\Delta\gamma$ serves as a lower bound to the trace of the Fisher information matrix.

Prediction-based method.—Previously, we generalized the indicator of PBM as

$$I_{\mathrm{PBM}}(\boldsymbol{\gamma}) = \sqrt{\sum_{i=1}^{d}\left(\frac{\partial\hat{\gamma}_i(\boldsymbol{\gamma})/\partial\gamma_i}{\mathrm{std}_i(\boldsymbol{\gamma})}\right)^2}, \tag{6.23}$$

where $\hat{\boldsymbol{\gamma}}(\boldsymbol{\gamma}) = \mathbb{E}_{\boldsymbol{x}\sim P(\cdot|\boldsymbol{\gamma})}\left[\hat{\boldsymbol{\gamma}}(\boldsymbol{x})\right]$ and $\mathbf{std}(\boldsymbol{\gamma}) = \sqrt{\mathbb{E}_{\boldsymbol{x}\sim P(\cdot|\boldsymbol{\gamma})}\left[\hat{\boldsymbol{\gamma}}(\boldsymbol{x})^2\right]-\left(\mathbb{E}_{\boldsymbol{x}\sim P(\cdot|\boldsymbol{\gamma})}\left[\hat{\boldsymbol{\gamma}}(\boldsymbol{x})\right]\right)^2}$ with $\hat{\boldsymbol{\gamma}}(\boldsymbol{x})$ being an estimator for $\boldsymbol{\gamma}$ (operations are carried out element-wise). Applying the scalar Cramér-Rao bound [Equation (5.27)] to each component of the sum in Equation (6.23), we obtain the bound

$$I_{\mathrm{PBM}}(\boldsymbol{\gamma}) \leq \sqrt{\mathrm{tr}\left[\mathcal{F}(\boldsymbol{\gamma})\right]}. \tag{6.24}$$

6.4 Applications to physical systems

We have proven a general relation between the Fisher information of a system and the ML indicators of phase transitions in Section 6.3. This result holds independent of the system under consideration. Let us now investigate this relationship in different physical contexts and provide numerical evidence for the quality of our bounds. As concrete examples, we consider a classical and a quantum model: the two-dimensional Ising model and the one-dimensional transverse-field Ising model.

6.4.1 Classical equilibrium systems

For a system at equilibrium with a large thermal reservoir, we have

$$P(\boldsymbol{x}|\boldsymbol{\gamma}) = e^{-\mathcal{H}(\boldsymbol{x},\boldsymbol{\gamma})}/Z(\boldsymbol{\gamma}), \tag{6.25}$$

where $\boldsymbol{x} \in \mathcal{X}$ denotes a configuration of the system, $Z(\boldsymbol{\gamma})$ is the partition function, and $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \ldots, \gamma_d)$ are tunable parameters, such as the temperature or magnetic field strength. Typically, the dimensionless Hamiltonian $\mathcal{H} = H/k_{\mathrm{B}}T$ takes the form

$$\mathcal{H} = \sum_{i=1}^{d}\gamma_i X_i(\boldsymbol{x}), \tag{6.26}$$

where $X_i(\boldsymbol{x})$ is a collective variable coupled to the tuning parameter γ_i. The Fisher information $\mathcal{F}_i$ associated with the parameter γ_i can be shown to measure changes in these collective variables [Prokopenko *et al.*, 2011]

$$\mathcal{F}_i(\boldsymbol{\gamma}) = -\frac{\partial \langle X_i \rangle}{\partial \gamma_i}. \tag{6.27}$$

Moreover, given that

$$\frac{\partial \beta A}{\partial \gamma_i} = \langle X_i \rangle, \tag{6.28}$$

where A is the Helmholtz free energy, we have

$$\mathcal{F}_i(\boldsymbol{\gamma}) = -\frac{\partial^2 \beta A}{\partial \gamma_i^2} \tag{6.29}$$

with $\beta = 1/k_{\mathrm{B}}T$. Because the Fisher information is related to second derivatives of the free energy, it is sensitive to first- and second-order divergences. In the case of thermal transitions where the tuning parameter is a function of T, for example, the Fisher information is proportional to the heat capacity $\mathcal{F} \propto C$.

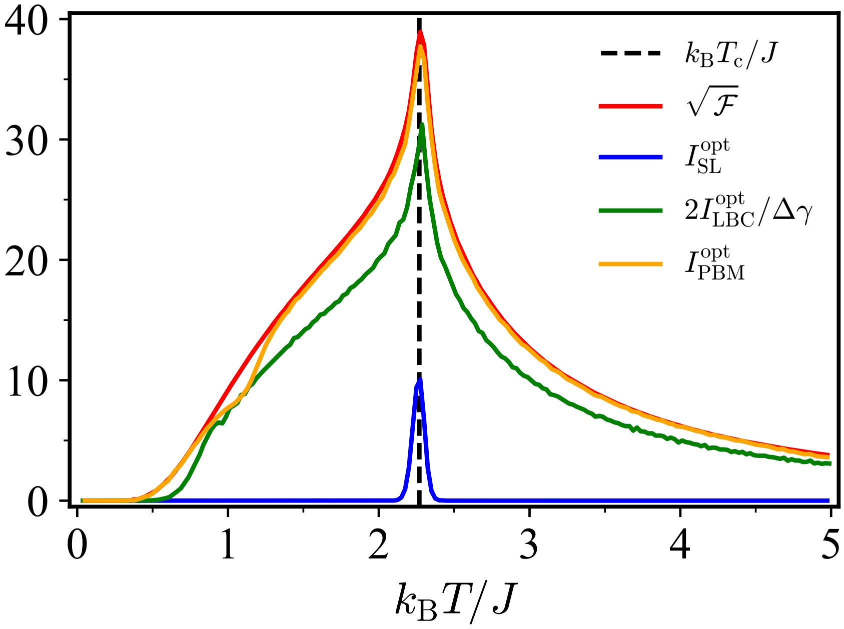

FIGURE 6.1: Results of the three data-driven approaches for detecting phase transitions [Equations (6.1), (6.3), and (6.4)] applied to the square-lattice ferromagnetic Ising model ($L = 60$). Here, we consider (approximate) Bayes-optimal predictive models. The critical point $k_{\mathrm{B}}T_{\mathrm{c}}/J = 2/\ln(1+\sqrt{2})$ is highlighted by a black dashed line. All shown quantities are lower bounds to the square root of the system's Fisher information. The set Γ is composed of a uniform grid with 200 points and grid spacing $\Delta\gamma = 0.025$. Each dataset D_γ consists of 10^6 spin configurations. For SL, we choose $\Gamma_0 = \{0.025, \dots, 2.25\}$ and $\Gamma_1 = \{2.275, \dots, 5\}$, i.e., we choose the two regions in parameter space to coincide with the two phases. For LBC, we choose $l = 1$.

Ising model

As a concrete example, we consider the thermal phase transition in the $L \times L$ square-lattice classical Ising model, see Section 2.3.1 for a discussion of the model and the data generation process.[1] Here, we choose $\gamma = k_{\mathrm{B}}T/J$ as a tuning parameter. The energy is the relevant collective variable $X(\boldsymbol{\sigma}) = H(\boldsymbol{\sigma})/J$ and the system's Fisher information corresponds to $CJ^2/k_{\mathrm{B}}^3 T^2$ where C is the heat capacity.

Based on the resulting datasets of drawn spin configurations $\{\mathcal{D}_\gamma\}_{\gamma\in\Gamma}$, we construct (approximate) *Bayes-optimal* predictive models using nonparametric generative models obtained via histogram binning of the sufficient statistic X following the procedure outline in Section 4.4.1. These models constitute global minima of the relevant loss functions [Equations (6.5) and (6.6)]. Figure 6.1 shows the indicators I^{opt} corresponding to these Bayes-optimal predictive models for the Ising model. They are good underapproximators of the square root of the system's Fisher information showing similar functional behavior. In particular, their peak positions are in agreement. Note that the SL indicator (including its peak position) depends heavily on the choice of Γ_0 and Γ_1, i.e., on prior knowledge of the location of the phase transition. Here, we chose to split the entire parameter range Γ into Γ_0 and Γ_1 at the critical point, i.e., we explicitly utilized information about the location of the underlying critical point. Different choices are investigated in Section 6.5.1.

6.4.2 Quantum systems

In quantum physics, measurements are most generally described by a positive operator-valued measure (POVM). The probability of obtaining the measurement outcome $\boldsymbol{x} \in \mathcal{X}$ associated with the POVM element $\Pi_{\boldsymbol{x}}$ is given by $P(\boldsymbol{x}|\gamma) = \mathrm{tr}\left[\Pi_{\boldsymbol{x}}\rho(\gamma)\right]$. The *quantum* Fisher information corresponds to the classical Fisher information maximized over all possible measurements [Braunstein and Caves, 1994]

$$\mathcal{F}^Q(\gamma) = \mathcal{F}^Q\left[\rho(\gamma)\right] = \max_{\{\Pi_{\boldsymbol{x}}\}_{\boldsymbol{x}\in\mathcal{X}}} \mathcal{F}\left[P(\cdot|\gamma)\right]. \tag{6.30}$$

Moreover, expanding the fidelity $F\left[\rho_1,\rho_2\right] = \mathrm{tr}\left[\sqrt{\sqrt{\rho_2}\rho_1\sqrt{\rho_2}}\right]$ between infinitesimally close states [Jozsa, 1994], we have

$$\begin{aligned} F\left[\rho(\gamma),\rho(\gamma+\Delta\gamma)\right] &= 1 - \Delta\gamma^2\mathcal{F}^Q\left[\rho(\gamma)\right]/8 + \mathcal{O}(\Delta\gamma^3) \\ &= 1 - \Delta\gamma^2\chi_{\mathcal{F}}\left[\rho(\gamma)\right]/2 + \mathcal{O}(\Delta\gamma^3), \end{aligned} \tag{6.31}$$

where $\chi_{\mathcal{F}} = \mathcal{F}^Q\left[\rho(\gamma)\right]/4$ is the fidelity susceptibility [Gu, 2010; Liu *et al.*, 2014].

Transverse-Field Ising model

As an example of a quantum system, we consider the transverse-field Ising model [Sachdev, 2011] on a (periodic) one-dimensional chain of length L whose Hamiltonian is given by

$$H = -J\sum_{\langle ij\rangle} Z_i Z_j - h\sum_i X_i, \tag{6.32}$$

where $J > 0$ is the nearest-neighbor interaction strength, h is the external field strength, and $\{X_i, Y_i, Z_i\}$ are the Pauli operators acting on the spin at site i. This model undergoes a quantum phase transition at zero temperature from a ferromagnetically ordered phase at $\gamma = h/J < 1$ to a disordered phase. We perform exact diagonalization using the QuSpin package [Weinberg and Bukov, 2017, 2019] in `Python` and consider projective measurements in the x-basis. We compute the quantum Fisher

[1] In the Metropolis-Hastings algorithm, after a thermalization period of $|\mathcal{D}_\gamma|$ lattice sweeps, we collect $|\mathcal{D}_\gamma|$ samples, where we set $|\mathcal{D}_\gamma| = 10^5$ or $|\mathcal{D}_\gamma| = 10^6$ in this chapter as specified in the respective figure captions.

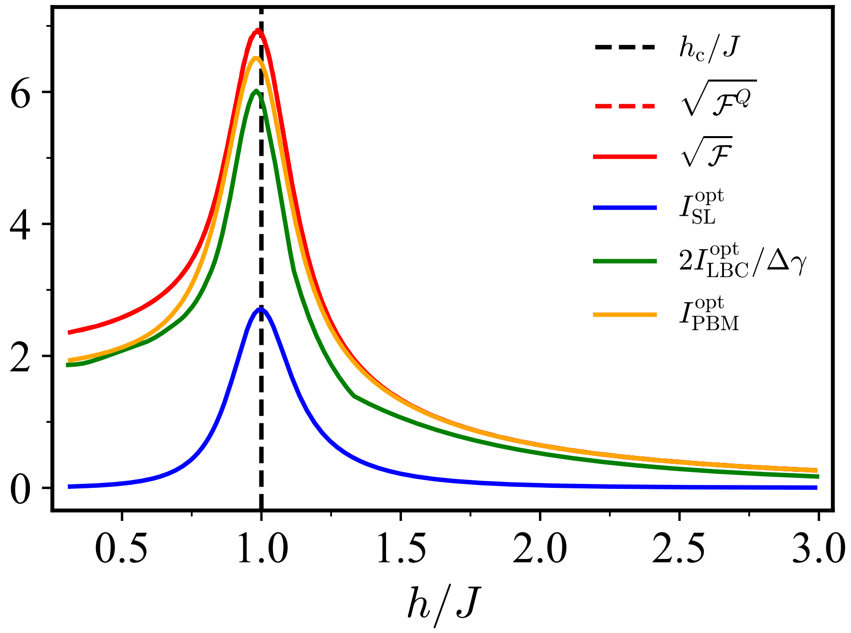

FIGURE 6.2: Results of the three data-driven approaches for detecting phase transitions [Equations (6.1), (6.3), and (6.4)] applied to the one-dimensional transverse-field Ising model ($L = 20$). Here, we consider Bayes-optimal predictive models constructed from the exact probability distributions underlying the measurement statistics (see Chapters 3 and 4), which globally minimize the corresponding loss functions. The critical point $h_c/J = 1$ is highlighted by a black dashed line [Sachdev, 2011]. All shown quantities are lower bounds to the square root of the system's classical and quantum Fisher information (in the figure these two curves overlap). The set Γ is composed of a uniform grid with 201 points and grid spacing $\Delta\gamma = 0.0135$. For SL, we choose $\Gamma_0 = \{0.3\}$ and $\Gamma_1 = \{3\}$, i.e., we choose the two sets to be composed of single points at the edges of the sampled region in parameter space. For LBC, we choose $l = 1$.

information as the second derivative of the fidelity F according to Equation (6.31). Speficially, we calculate a finite difference approximation of the second derivative using the second-order central difference formula.

The optimal indicators of all three methods yield non-trivial bounds on the square root of the system's classical and quantum Fisher information and their peak positions agree, see Figure 6.2. Applying classical data-driven methods to quantum systems requires the choice of a POVM. Many previous works have successfully detected phase transitions using simple projective measurements in a single basis [Greplova *et al.*, 2020; Miles *et al.*, 2021; Bohrdt *et al.*, 2021; Maskara *et al.*, 2022; Miles *et al.*, 2023]. We have also made that choice in Chapter 3. Our findings in this chapter highlight that a good choice of measurement is one that results in a high classical Fisher information, i.e., one for which the latter is close to the quantum Fisher information.

6.5 Choice of training regions, novel bounds, and neural network-based results

Having showcased that all three data-driven methods – SL, LBC, and PBM – provide accurate bounds to the square root of the system's Fisher information, in this section, we dive deeper into additional refinements and variants of these methods and provide further insights, particularly related to NN-based training. The section is split into three parts dedicated to each of the three methods:

In Section 6.5.1, we discuss the influence of the choice of training region on the bound to the Fisher information obtained with SL. In particular, we identify a choice of training regions that leads to a tight bound of the Fisher information. In Section 6.5.2, we showcase that the indicator of LBC obtained in a discriminative fashion using NN-based classifiers quickly approaches the optimal indicator during training. Moreover, we prove that the loss function in LBC also provides a (tight) bound to the Fisher information and may be used as an alternative indicator. Finally, in Section 6.5.3, we explain how NN-based predictive models in PBM fail or succeed at detecting phase transitions at different stages during training through the bias-variance tradeoff.

6.5.1 Supervised learning

The indicator of SL (including its peak position) depends heavily on the choice of Γ_0 and Γ_1, i.e., on prior knowledge of the location of the phase transition, or equivalently, the system's phases. Figure 6.3 shows optimal indicators $I_{\mathrm{SL}}^{\mathrm{opt}}$ for various choices of the training regions in the case of the two-dimensional classical Ising model. Note that while the peak position shifts depending on the choice of training regions, the largest indicator value (i.e., the best bound to the Fisher information) is achieved when the parameter space is split at the critical point. Recall that we have made a similar observation in Appendix C. This suggests that, in practice, SL should be performed for various choices of the sets Γ_0 and Γ_1. By tracing the maximum indicator value attained at each point in parameter space, an overall improved lower bound to the square root of the Fisher information can be obtained, see envelopes given by dashed and dotted lines in Figure 6.3. Note that such an approach is close in spirit to LBC.

In Section 6.3, we pointed out that SL could be improved by modifying the indicator $I_{\mathrm{SL}}(\gamma) \mapsto I'_{\mathrm{SL}}(\gamma) = I_{\mathrm{SL}}(\gamma)/\mathrm{std}(\gamma)$. Figure 6.3 also displays the rescaled optimal indicators $I_{\mathrm{SL}}^{\mathrm{opt}\prime}$ for various choices of the training regions, highlighting that this small modification indeed yields improved bounds and, in fact, accurate approximations to the Fisher information near the bipartition boundary.

Let us now discuss a natural scenario in which the optimal predictive model $\hat{y}_{\mathrm{SL}}^{\mathrm{opt}}$ with respect to the cross-entropy loss [Equation (6.5)] is strongly correlated with the score $\partial \ln\left[P(\cdot|\gamma)\right]/\partial\gamma$. For this, let $\Gamma_0 = \{\gamma_0\}$ and $\Gamma_1 = \{\gamma_1\}$ be singletons such that $\gamma_0 < \gamma_1$. In this case, $\hat{y}_{\mathrm{SL}}^{\mathrm{opt}}$ admits the following closed-form expression (see Chapter 3)

$$\hat{y}_{\mathrm{SL}}^{\mathrm{opt}}(\boldsymbol{x}) = 1 - \frac{P(\boldsymbol{x}|\gamma_0)}{P(\boldsymbol{x}|\gamma_0) + P(\boldsymbol{x}|\gamma_1)}. \tag{6.33}$$

When γ_0 and γ_1 are far away from each other, there is, at first glance, no reason to expect that $\hat{y}_{\mathrm{SL}}^{\mathrm{opt}}$ will be well correlated with the score for any $\gamma \in [\gamma_0, \gamma_1]$. However, if we consider γ_0 and γ_1 to be close, i.e., $\gamma_1 = \gamma_0 + \Delta\gamma$, and assume that the distributions evolve sufficiently smoothly with γ so that

$$\delta P = P(\cdot|\gamma_1) - P(\cdot|\gamma_0) = \left.\frac{\partial P(\cdot|\gamma)}{\partial\gamma}\right|_{\gamma=\gamma_0} \Delta\gamma + \mathcal{O}(\Delta\gamma^2), \tag{6.34}$$

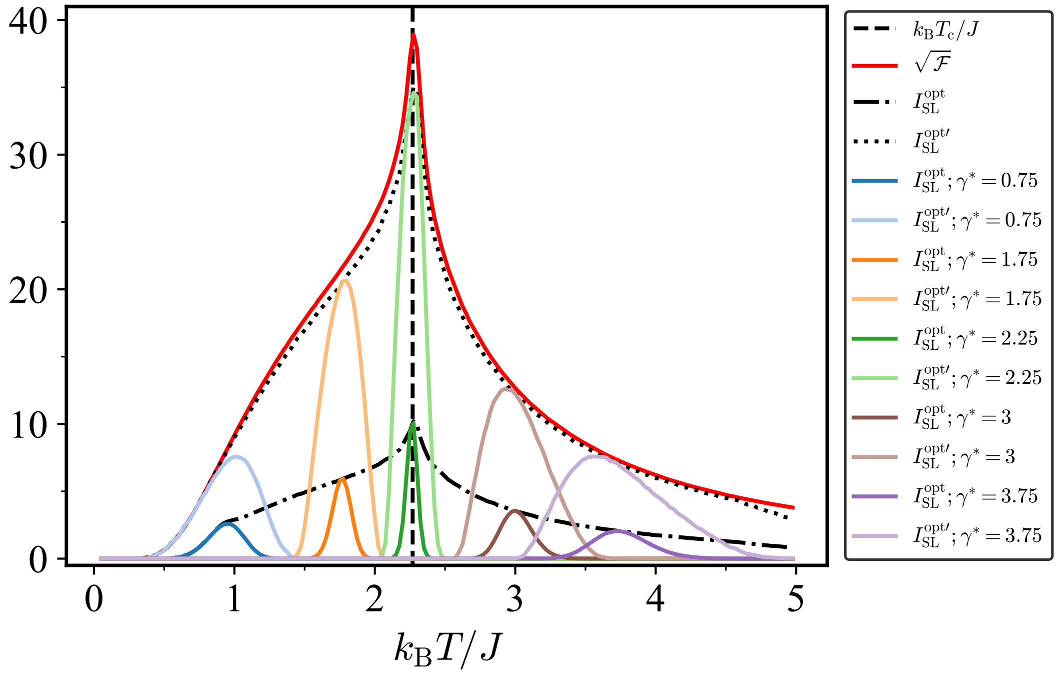

FIGURE 6.3: Results of SL applied to the square-lattice ferromagnetic Ising model ($L=60$) with tuning parameter $\gamma = k_\mathrm{B}T/J$. The critical point $k_\mathrm{B}T_\mathrm{c}/J = 2/\ln(1+\sqrt{2})$ is highlighted by a vertical black dashed line. All shown quantities are lower bounds to the square root of the system's Fisher information, here corresponding to $CJ^2/k_\mathrm{B}^3T^2$. The indicators $I_\mathrm{SL}^\mathrm{opt}$ and $I_\mathrm{SL}^{\mathrm{opt}\prime}$ are computed for bipartitions of the parameter range $\Gamma = \{0.025, 0.1, \ldots, 5.0\}$ into two regions $\Gamma_0 = \{0.025, \ldots, \gamma^*\}$ and $\Gamma_1 = \{\gamma^*+0.025, \ldots, 5.0\}$. For both $I_\mathrm{SL}^\mathrm{opt}$ and $I_\mathrm{SL}^{\mathrm{opt}\prime}$, the five bold lines correspond to five distinct $\gamma^* \in \{0.75, 1.75, 2.25, 3, 3.75\}$ that determine the location of the bipartition. Note that the bipartition at 2.25 almost coincides with the critical point located at ≈ 2.27. The envelopes of the two indicator curves across all possible bipartitions, i.e., all possible choices of γ^*, are shown by dashed and dotted lines. The set Γ is composed of a uniform grid with 200 points ranging from $\gamma = 0.025$ to $\gamma = 5$ (grid spacing $\Delta\gamma = 0.025$). Each dataset D_γ consists of 10^6 spin configurations.

we have

$$\begin{aligned}
\hat{y}_\mathrm{SL}^\mathrm{opt}(\boldsymbol{x}) &= 1 - \frac{1}{2}\frac{1}{1+\frac{\delta P(\boldsymbol{x})}{2P(\boldsymbol{x}|\gamma_0)}} \\
&= \frac{1}{2} + \frac{1}{4}\frac{\delta P(\boldsymbol{x})}{P(\boldsymbol{x}|\gamma_0)} + \mathcal{O}\left(\delta P(\boldsymbol{x})^2\right) \\
&= \frac{1}{2} + \frac{1}{4}\left.\frac{\partial \ln\left[P(\boldsymbol{x}|\gamma)\right]}{\partial\gamma}\right|_{\gamma=\gamma_0}\Delta\gamma + \mathcal{O}\left(\Delta\gamma^2\right). \qquad (6.35)
\end{aligned}$$

Thus, in the limit of small $\Delta\gamma$, $\hat{y}$ is indeed strongly correlated with $\partial \ln\left[P(\cdot|\gamma)\right]/\partial\gamma|_{\gamma_0}$ and so $I_\mathrm{SL}^\mathrm{opt}(\gamma_0)/\Delta\gamma$ is a first-order accurate, constant factor approximation for $\mathcal{F}(\gamma_0)$, $I_\mathrm{SL}^\mathrm{opt}(\gamma_0)/\Delta\gamma \to \mathcal{F}(\gamma_0)/4$ as $\Delta\gamma \to 0$ (note that $I_\mathrm{SL}^\mathrm{opt}$ depends implicitly on $\Delta\gamma$). This scenario is similar to the one of LBC with $l=1$ and suggests a new scheme for detecting phase transitions by approximating the Fisher information in a data-driven manner: for each $\gamma \in \Gamma$ where Γ is composed of a uniform grid with spacing $\Delta\gamma$, let

$\gamma_0 = \gamma$ and $\gamma_1 = \gamma + \Delta\gamma$ and obtain a point-wise estimate of the Fisher information as $4I_{\rm SL}(\gamma)/\Delta\gamma$.

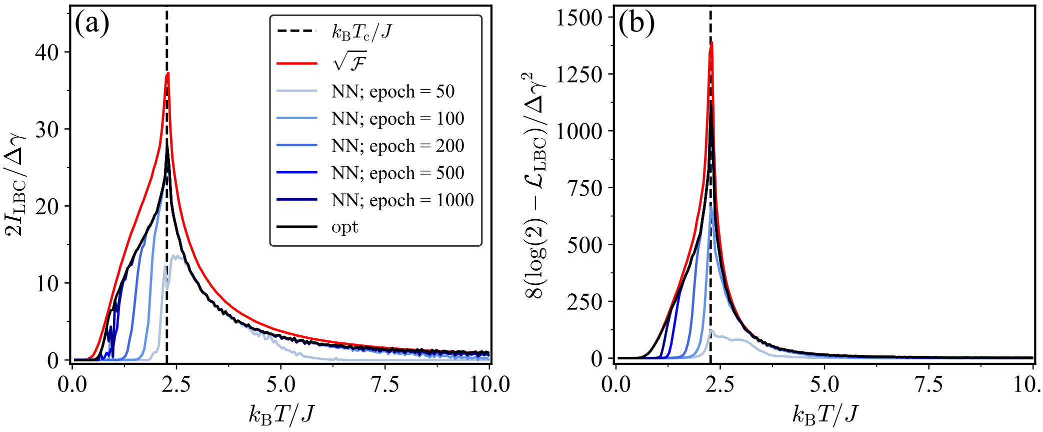

FIGURE 6.4: Results of LBC applied to the square-lattice ferromagnetic Ising model ($L = 60$) with tuning parameter $\gamma = k_{\rm B}T/J$. The critical point $k_{\rm B}T_{\rm c}/J = 2/\ln(1+\sqrt{2})$ is highlighted by a vertical black dashed line. The results of an NN-based classifier for various training epochs are shown in blue. Indicators corresponding to the (approximate) Bayes-optimal classifier are shown in black (see Chapter 4). (a) Rescaled indicator $I_{\rm LBC}$ [Equation (6.3), $l = 1$]. (b) Bound based on loss function [Equation (6.42)]. The set Γ is composed of a uniform grid with 200 points ranging from $\gamma = 0.05$ to $\gamma = 10$ (grid spacing $\Delta\gamma = 0.05$). Each dataset D_γ consists of 10^5 spin configurations. We consider feedforward NNs (implemented using Flux [Innes, 2018] in `Julia` [Bezanson *et al.*, 2012]) with three hidden layers composed of 64 nodes each, ReLUs as activation functions, and a learning rate of $5 \cdot 10^{-4}$. Weights and biases are optimized via gradient descent with Adam [Kingma and Ba, 2014], where the gradients are calculated using backpropagation. As an NN input, we use the energy of a sample, which corresponds to the sufficient statistic. The inputs are standardized before training.

6.5.2 Learning by confusion

In Section 6.3, we proved that the indicator of LBC arising from any predictive model serves as a lower bound to the indicator of the corresponding Bayes-optimal predictive model. Figure 6.4(a) shows the indicator of LBC for a (discriminative) NN-based predictive model at various stages of training. The NN-based indicator quickly approaches the optimal indicator, highlighting that good approximations to the Fisher information may also be achieved in practice via explicit data-driven training.

Figures D.1-D.6 in Appendix D showed that the loss function of LBC also carries information about the underlying phase transition and may serve as an alternative indicator function that shows a local minimum at the critical point. In the following, we will put this observation on firm footing:

In the infinite data limit, the loss function in LBC [Equation (6.5)] becomes

$$\mathcal{L}_{\rm LBC} = -\frac{1}{2}\Big(\mathbb{E}_{\boldsymbol{x}\sim P_0}\Big[\ln\Big(1 - \hat{y}_{\boldsymbol{\theta}}(\boldsymbol{x})\Big)\Big] + \mathbb{E}_{\boldsymbol{x}\sim P_1}\Big[\ln\Big(\hat{y}_{\boldsymbol{\theta}}(\boldsymbol{x})\Big)\Big]\Big) \tag{6.36}$$

where $P_y = \frac{1}{|\Gamma_y|}\sum_{\gamma'\in\Gamma_y} P(\cdot|\gamma')$, $y \in \{0,1\}$. By definition $\mathcal{L}_{\rm LBC} \geq \mathcal{L}_{\rm LBC}^{\rm opt}$ where $\mathcal{L}_{\rm LBC}^{\rm opt}$ is the global minimum of the loss function attained by the Bayes-optimal strategy,

$\hat{y}_{\boldsymbol{\theta}}(\boldsymbol{x}) \mapsto \hat{y}^{\mathrm{opt}}_{\mathrm{LBC}}(\boldsymbol{x})$. In LBC, the optimal model makes the following predictions (see Section 5.4 and Chapter 3)

$$\hat{y}^{\mathrm{opt}}_{\mathrm{LBC}}(\boldsymbol{x}) = \frac{P_1(\boldsymbol{x})}{P_0(\boldsymbol{x}) + P_1(\boldsymbol{x})}. \tag{6.37}$$

Thus,

$$\begin{aligned}\mathcal{L}^{\mathrm{opt}}_{\mathrm{LBC}} &= -\frac{1}{2}\left(\mathbb{E}_{\boldsymbol{x}\sim P_0}\left[\ln\left(\frac{P_0(\boldsymbol{x})}{P_0(\boldsymbol{x}) + P_1(\boldsymbol{x})}\right)\right] + \mathbb{E}_{\boldsymbol{x}\sim P_1}\left[\ln\left(\frac{P_1(\boldsymbol{x})}{P_0(\boldsymbol{x}) + P_1(\boldsymbol{x})}\right)\right]\right)\\ &= -D_{\mathrm{JS}}\left[P_0, P_1\right] + \ln(2). \end{aligned}\tag{6.38}$$

Hence,

$$D_{\mathrm{JS}}\left[P_0, P_1\right] = \ln(2) - \mathcal{L}^{\mathrm{opt}}_{\mathrm{LBC}} \geq \ln(2) - \mathcal{L}_{\mathrm{LBC}}. \tag{6.39}$$

Next, we choose $l = 1$ and the two sets of points Γ_0 and Γ_1 such that P_0 and P_1 correspond to probability distributions separated by a small distance $\Delta\gamma$ in parameter space. Expanding the JS divergence in lowest order according to

$$D_f[p_{\boldsymbol{\gamma}}, p_{\boldsymbol{\gamma}+\boldsymbol{\Delta\gamma}}] \approx \frac{f''(1)}{2}\boldsymbol{\Delta\gamma}^T\mathcal{F}(\boldsymbol{\gamma})\boldsymbol{\Delta\gamma} \tag{6.40}$$

with $f''(1) = 1/4$ (note that $f'(1) = 0$), we have

$$D_{\mathrm{JS}}\left[P_0, P_1\right] = \frac{\Delta\gamma^2}{8}\mathcal{F}(\gamma) + \mathcal{O}(\Delta\gamma^3). \tag{6.41}$$

Together with the above bound, this yields

$$8(\ln(2) - \mathcal{L}_{\mathrm{LBC}})/\Delta\gamma^2 \leq 8(\ln(2) - \mathcal{L}^{\mathrm{opt}}_{\mathrm{LBC}})/\Delta\gamma^2 = \mathcal{F}(\gamma) + \mathcal{O}(\Delta\gamma). \tag{6.42}$$

That is, an affine transformation of the loss value serves as a lower bound to the Fisher information in the limit $\Delta\gamma \to 0$. We derived this bound by relating the optimal loss value in LBC to the JS divergence [Equation (5.4)] which is an f-divergence. An expansion in the lowest order then allows us to relate the indicator to the Fisher information (recall the discussion in Section 5.3). As such, looking at the loss in LBC corresponds to another data-driven scheme for estimating the Fisher information based on approximating an f-divergence that can be used to detect phase transitions. The results for the Ising model are shown in Figure 6.4(b).

6.5.3 Prediction-based method

The loss function in PBM [mean squared error in Equation (6.6)]

$$\mathcal{L}_{\mathrm{PBM}} = \frac{1}{|\Gamma|}\sum_{\gamma\in\Gamma}\mathbb{E}_{\boldsymbol{x}\sim P(\cdot|\gamma)}\left[\left(\hat{\gamma}(\boldsymbol{x}) - \gamma\right)^2\right], \tag{6.43}$$

can be decomposed into two terms

$$\mathcal{L}_{\mathrm{PBM}} = \frac{1}{|\Gamma|}\sum_{\gamma\in\Gamma}\mathrm{std}^2(\gamma) + b^2(\gamma) = \langle\mathrm{std}^2\rangle + \langle b^2\rangle, \tag{6.44}$$

where $\mathrm{std}^2(\gamma)$ measures the variance of the prediction at γ

$$\mathrm{std}^2(\gamma) = \mathbb{E}_{\boldsymbol{x}\sim P(\cdot|\gamma)}\left[\left(\hat{\gamma}(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{x}\sim P(\cdot|\gamma)}\left[\hat{\gamma}(\boldsymbol{x})\right]\right)^2\right] \tag{6.45}$$

and $b(\gamma)$ measures the bias of the prediction

$$b(\gamma) = \mathbb{E}_{\boldsymbol{x}\sim P(\cdot|\gamma)}\left[\hat{\gamma}(\boldsymbol{x}) - \gamma\right]. \tag{6.46}$$

When working with a finite dataset, the variance and bias should be replaced by their finite sample approximations, i.e., expected values should be replaced with sample means

$$\mathbb{E}_{\boldsymbol{x}\sim P(\cdot|\gamma)}\left[\cdot\right] \mapsto \frac{1}{|\mathcal{D}_\gamma|}\sum_{\boldsymbol{x}\in\mathcal{D}_\gamma}\left[\cdot\right]. \tag{6.47}$$

The PBM indicator

$$I_{\mathrm{PBM}}(\gamma) = \frac{\partial\hat{\gamma}(\gamma)/\partial\gamma}{\mathrm{std}(\gamma)} = \frac{\partial b(\gamma)/\partial\gamma + 1}{\mathrm{std}(\gamma)}, \tag{6.48}$$

where $\hat{\gamma}(\gamma) = \mathbb{E}_{\boldsymbol{x}\sim P(\cdot|\gamma)}\left[\hat{\gamma}(\boldsymbol{x})\right]$, can be viewed as a "signal-to-noise" ratio where the "signal" term $\partial\hat{\gamma}(\gamma)/\partial\gamma$ corresponds to the change in the bias of the estimator and the "noise" term corresponds to the standard deviation of the estimator. Both have independently been used as indicators of phase transitions. In particular, in Chapter 3 we have found that the change in the bias of the estimator alone is a reliable indicator early on during training of NN-based predictive models. However, as the capacity of the predictive models increases, the change in their bias can show additional, erroneous peaks that do not correspond to critical points. This is illustrated in Figure 6.5(b) in the case of the Ising model. In Chapter 4, we have found that these erroneous peaks can be removed by dividing by the standard deviation. Due to the bias-variance tradeoff [Friedman *et al.*, 2001], during NN training, the bias contribution to the loss typically decreases while the variance contribution increases. This is shown for the Ising model in Figure 6.5(d). If the decrease of the bias precedes the increase in variance, it is expected that the change in the bias alone can constitute a reliable indicator early on during training. In contrast, the standard deviation is expected to be reliable only at later stages, see Figure 6.5(d). Their ratio, the indicator proposed in Equation (6.48), yields a reliable signal throughout both stages of training, see Figure 6.5(a). In particular, their ratio yields a fairly good lower bound to the square root of the Fisher information even at the early stages of training.

6.6 Discussion

We find that all three methods – SL, LBC, and PBM – can be viewed as data-driven approaches for constructing approximations of the Fisher information that remain operationally useful for identifying phase transitions. This provides a strong unification of these methods, justifies their current formulation, and explains their previous successes in detecting phase transitions, given that the Fisher information is known to highlight phase transitions. In the case of classical equilibrium systems, for example, the Fisher information is related to second derivatives of the free energy and is thus sensitive to first- and second-order divergences. Because SL, LBC, and PBM yield approximations to the Fisher information, these methods can also detect such transitions (provided that the approximation to the Fisher information is accurate enough).

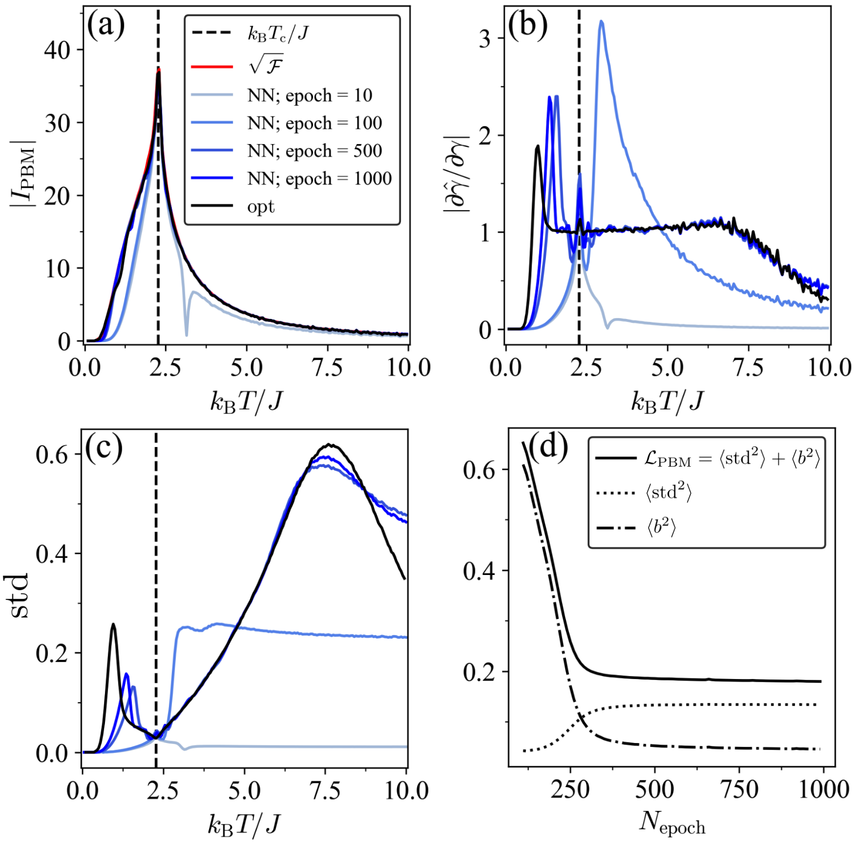

FIGURE 6.5: Results of PBM applied to the square-lattice ferromagnetic Ising model ($L = 60$) with tuning parameter $\gamma = k_{\mathrm{B}}T/J$. The critical point $k_{\mathrm{B}}T_{\mathrm{c}}/J = 2/\ln(1+\sqrt{2})$ is highlighted by a vertical black dashed line. (a) The optimal indicator (black) as well as NN-based indicators (blue) are lower bounds to the square root of the system's Fisher information (red). (b) Numerator of the indicator I_{PBM} [Equation (6.4)] in optimal case (black) and NN-based case after various training epochs (blue). (c) Denominator of indicator I_{PBM} [Equation (6.4)] in optimal case (black) and NN-based case after various training epochs (blue). (d) Loss function of NN-based predictive model and its bias-variance decomposition as a function of the number of training epochs. The set Γ is composed of a uniform grid with 200 points ranging from $\gamma = 0.05$ to $\gamma = 10$ (grid spacing $\Delta\gamma = 0.05$). Each dataset D_γ consists of 10^5 spin configurations. We consider feedforward NNs (implemented using Flux [Innes, 2018] in `Julia` [Bezanson *et al.*, 2012]) with three hidden layers with 64 nodes each, ReLUs as activation functions, and a learning rate of 10^{-3}. Weights and biases are optimized via gradient descent with Adam [Kingma and Ba, 2014], where the gradients are calculated using backpropagation. As an NN input, we use the energy of a sample, which corresponds to the sufficient statistic. The inputs are standardized before training.

In the case of thermal transitions with temperature as a tuning parameter, for example, the Fisher information is proportional to the heat capacity. Similarly, when the magnetic field strength is a tuning parameter, the Fisher information is proportional to the magnetic susceptibility. While a connection between such physical susceptibilities and the ML indicators has previously been suggested [Beach *et al.*, 2018; Suchsland and Wessel, 2018; Guo and He, 2023] (see also Chapter 3), our results make this connection explicit. In fact, *a posteriori* it seems natural to devise generic methods for detecting phase transitions by generalizing known quantities that highlight them, such as physical susceptibilities. The Fisher information is a suitably generalized susceptibility that reduces to well-known susceptibilities in well-studied physical systems.

Having identified the Fisher information as an ideal quantity leaves us with the problem of computing it. The Fisher information is difficult to access if only samples of the system are available, as computing it generally requires knowledge of the probabilistic model underlying the measurement statistics of the system [see Equation (6.7)]. When paired with discriminative models, SL, LBC, and PBM provide frameworks for approximating the Fisher information from samples – without having access to a probabilistic model.

The relation to the Fisher information not only explains the successes of SL, LBC, and PBM, but also their limitations. A variety of previous numerical studies [Beach *et al.*, 2018; Suchsland and Wessel, 2018; Guo and He, 2023] (including Chapter 3) have found that NN-based methods struggle to detect thermal transitions of the BKT-type in classical equilibrium systems. In particular, the NN-based indicators have been observed to show a peak at the same position as the heat capacity, away from the critical point. The relation between NN-based indicators and the Fisher information we have uncovered offers a natural explanation for this observation, given that the Fisher information reduces to the heat capacity in such a case while the BKT transition is of infinite order. More generally, given that the Fisher information is related to second derivatives of the free energy and is thus only sensitive to first- and second-order divergences, our work suggests that the same holds for the ML methods we considered.[2]

Note that this statement only holds when the methods are applied to raw measurement data (or representations that can be shown to contain the same information, such as the sufficient statistic). In [Beach *et al.*, 2018], for example, it was found that SL works significantly better at detecting the BKT transition in the XY model when fed with the vorticities (i.e., winding numbers) instead of raw spin configurations. While feature engineering may provide a workaround to detect topological phase transitions, it is not a particularly satisfactory one given our motivation to map out novel phase diagrams with little to no prior system knowledge. Finding a suitable quantity beyond the Fisher information that can be approximated to reliably detect higher-order phase transitions remains open.

Focusing on learning by confusion

For the remainder of this thesis, we will focus on LBC, leaving SL and PBM aside. Let us comment on the reasons behind this choice:

- First, we have seen time and time again that SL requires prior knowledge of the number and location of the underlying phases to work well. In particular, its lower bound on the Fisher information is fairly loose and strongly dependent on the choice of the underlying training regions.

- Compared to SL, PBM yields more accurate approximations of the Fisher information and does not require such prior knowledge. However, its indicator

[2] Our work only *suggests* this fact given that the methods only yield approximations to the Fisher information and do not reproduce the Fisher information exactly. As such, in any application, the ML indicator may, strictly speaking, differ from the Fisher information.

requires an estimate of the variance of the predictions and involves the explicit computation of a derivative; both seem to make the procedure more sample-intensive compared to LBC.[3] Recall that the discriminative variant of PBM has originally been introduced to alleviate the computational burden that comes with the original formulation of discriminative LBC ($l = \infty$): the latter requires training a distinct predictive model for each bipartition whereas PBM only requires a single predictive model. When constructing predictive models in a generative manner, the computational discrepancy between the two methods largely disappears (see Section 4.5.1). In fact, when the parameter space is large and l in LBC is chosen fairly small, generative PBM is typically more expensive than LBC – even considering the same number of samples. As we will show in Chapter 7, we can also reduce the computational cost of discriminative LBC further, making it comparable to discriminative PBM (even considering the same number of samples). In summary, from a computation point of view there does not seem to be a reason to choose PBM over LBC.

- In SL and PBM, the connection to well-established statistical quantities, such as the Fisher information, only emerges via bounds. In contrast, the Bayes-optimal indicator of LBC can be shown to correspond to the TV distance between the probability distributions underlying neighboring regions in parameter space. Similarly, in the infinite-data limit, its optimal loss value can be related to the JS divergence. For any predictive model, the corresponding indicator and loss yield bounds to the TV distance and JS divergence, respectively. As the predictive model gets more accurate, such as during training, these bounds are guaranteed to systematically improve. Both the TV distance and the JS divergence are f-divergences (see Section 5.2). Hence, when considering neighboring regions that are separated by a distance $\Delta\gamma$ in parameter space, in lowest-order in $\Delta\gamma$, these quantities relate to the Fisher information. In fact, the bound based on the loss value of LBC (i.e., the JS divergence) is tight as $\Delta\gamma \to 0$ [Equation (6.42)].

At the core of LBC sits the idea of detecting phase transitions as large statistical distances between the probability distributions underlying neighboring regions in parameter space. These statistical distances can be estimated based on a classifier trained from data, i.e., without explicit access to the underlying distributions. It turns out that this technique can be leveraged to estimate other f-divergences beyond the TV distance and the JS divergence. We will introduce this general framework of detecting phase transitions based on f-divergences in more detail in Chapter 8.

6.7 Summary

We have unveiled a fundamental connection between the (so far disparate) information-theoretic and ML paradigms of studying critical phenomena. The indicators of phase transitions of SL, LBC, and PBM (and its various variants) can be shown to approximate the square root of the system's Fisher information from below – a quantity that is well-known to signal phase transitions. We numerically demonstrated the high quality of these underapproximations for phase transitions in classical equilibrium systems and quantum ground states. Our result sheds light on the fundamental working principle and limitations of these ML methods for detecting phase transitions.

[3] Recall that we have found the division of the signal by the standard deviation to be crucial for PBM to accurately detect phase transitions, cf. Chapter 4.

6.8 Outlook

Interestingly, data-driven schemes for estimating the Fisher information based on approximating different statistical divergences – akin to the variational lower bound of the TV distance that underlies LBC – have recently been proposed [Berisha and Hero, 2014; Duy *et al.*, 2022]. Similar ML methods based on variational representations have also been utilized to estimate other classical [Nguyen *et al.*, 2007; Belghazi *et al.*, 2018] as well as quantum information-theoretic quantities [Cerezo *et al.*, 2020; Tan and Volkoff, 2021; Beckey *et al.*, 2022; Shin *et al.*, 2024; Goldfeld *et al.*, 2024]. In light of our results, such approaches may give rise to a whole new set of numerical methods to detect phase transitions from data (besides SL, LBC, and PBM) that have gone unnoticed. Similarly, it remains to be investigated to what extent methods for detecting phase transitions may be useful beyond this task, i.e., whether these methods yield novel ways to approximate statistical quantities such as the Fisher information in more general contexts. We will start to explore the use of these methods in problems beyond physics in the second part of this thesis.

In the quantum case, we have so far worked with a fixed measurement setting. One may, however, imagine combining the classical ML schemes of SL, LBC, and PBM with quantum NNs [Cong *et al.*, 2019; Herrmann *et al.*, 2022], i.e., variational quantum circuits, to estimate a system's quantum Fisher information in a data-driven manner. In such an approach, the quantum NN parametrizes the measurement basis and enables a search procedure over this space. For each choice of basis, i.e., the setting of the parameters of the quantum circuit, the classical ML schemes yield an indicator signal – an estimate of the corresponding classical Fisher information – that can be used for updating the parameters of the quantum circuit. Estimations of the quantum Fisher information are not only useful for detecting quantum phase transitions but also, for example, for characterizing quantum sensing protocols [MacLellan *et al.*, 2024].

In this chapter, we have investigated the quality of the approximations to the Fisher information obtained by the ML methods for two concrete physical systems numerically. Can one characterize the gap between the indicators and the Fisher information, i.e., the tightness of the bounds presented in this chapter, more thoroughly? In particular, given that all three methods and their variants yield approximations to the Fisher information, can one determine which one yields the best approximations?

The question of how many samples are needed to obtain accurate estimates of the indicators – and in turn the Fisher information – also remains largely open. Having formulated the ML methods in a fully probabilistic fashion in Chapter 4, one may consider concentration inequalities, such as Hoeffding's and Chebyshev's inequalities, as a starting point to obtain bounds within the probably approximately correct (PAC) learning framework [Kliesch, 2021]. The connection to well-known statistical distances may enable other approaches for tackling questions regarding sample complexity.

Throughout this thesis, including this chapter, we have focused on SL, LBC, and PBM for detecting phase transitions. It will be interesting to see what other ML methods for detecting phase transitions are related to the Fisher information. For example, the generative adversarial network fidelity defined in [Singh *et al.*, 2021] can also be shown to approximate the square root of the Fisher information from below, see Appendix G. We anticipate that information theory will be instrumental in understanding, categorizing, and improving the growing number of ML methods for detecting phase transitions, with our results forming the basis for such efforts.

The results and figures presented in this chapter have been in parts published in [Arnold *et al.*, 2023a].

Chapter 7

Getting Confused Faster Through Multitasking and Detecting Changes in Diffusion Models

The results presented in this chapter are based on the following publication:

Fast detection of phase transitions with multi-task learning-by-confusion,
J. Arnold, F. Schäfer, and N. Lörch,
NeurIPS 2023 Machine Learning and the Physical Sciences Workshop,
arXiv:2311.09128 (2023).

7.1 Motivation

So far, the implementation of LBC in a discriminative manner (i.e., without access to the probability distribution underlying the system) has been computationally costly. In the case of a one-dimensional parameter space, for example, it involves training $|\Gamma|-1$ distinct binary classifiers[1] and analyzing their accuracy, where $|\Gamma|$ is the number of sampled values of the tuning parameter.

In this chapter, we propose an alternative implementation of the discriminative LBC scheme for one-dimensional parameter spaces that works by training a *single* $(|\Gamma|-1)$-class classifier. In the ideal case, this implementation yields a speedup by a factor of $|\Gamma|-1$. Numerically, we demonstrate a significant speedup in detecting the thermal phase transition of the Ising model. In addition, we investigate an image dataset generated with the text-to-image generative model *Stable Diffusion* (version 2.1) [Rombach *et al.*, 2021], which serves as a more challenging example where no prior knowledge of transitions is available. In this case, the full speedup is realized.

7.2 Learning by confusion with multitasking

For clarity, let us recall how the discriminative LBC scheme has been applied so far. Here, we consider the simplest application scenario of LBC in which a physical system undergoes a phase transition as a function of a single real-valued parameter γ. To detect the critical point γ_c at which the phase transition occurs, the γ-axis is discretized into $|\Gamma| = K$ different points where $\Gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_K\}$ and at each point, $|\mathcal{D}_\gamma|$ samples are drawn from the system. With $|\Gamma|$ points, there are $|\Gamma|-1$ possibilities to separate the γ-axis in two non-empty, contiguous regions $\Gamma_0(\gamma_k^{\mathrm{bp}}) =$

[1] Here, we disregard the trivial classification tasks arising at the edges of the sampled parameter space in which all data is assigned the same label.

$\{\gamma \in \Gamma | \gamma < \gamma_k^{\mathrm{bp}}\}$ and $\Gamma_1(\gamma_k^{\mathrm{bp}}) = \{\gamma \in \Gamma | \gamma > \gamma_k^{\mathrm{bp}}\}$.[2] Each bipartition corresponds to a tentative location $\gamma_k^{\mathrm{bp}} = (\gamma_k + \gamma_{k+1})/2$ of the phase transition that lies in-between the grid points γ_k and γ_{k+1}, where $k \in \{1, 2, \ldots, K-1\}$. All of the samples drawn at parameter values $\gamma \in \Gamma_{0/1}$ are assigned the label 0 or 1, respectively.

Traditionally, for each of these splittings, a separate classifier is trained to distinguish the two corresponding classes of samples, see Figure 7.1(a). Intuitively, whichever classifier $k \in \{1, 2, \ldots, K-1\}$ is least confused, i.e., achieves the lowest error rate on evaluation, must have been trained on the most natural splitting of the data. Therefore, the value γ^{bp} associated with the lowest error rate is our best guess for the location of the phase transition.

The training loss function of the kth classifier is an unbiased binary cross-entropy loss

$$\mathcal{L}(\boldsymbol{\theta}^{(k)}|\gamma_k^{\mathrm{bp}}) = -\frac{1}{2} \sum_{y \in \{0,1\}} \frac{1}{|\mathcal{T}_y^{(k)}|} \sum_{\boldsymbol{x} \in \mathcal{T}_y^{(k)}} \left(y \ln \left[\hat{y}^{(k)}_{\boldsymbol{\theta}^{(k)}}(\boldsymbol{x}) \right] + (1-y) \ln \left[1 - \hat{y}^{(k)}_{\boldsymbol{\theta}^{(k)}}(\boldsymbol{x}) \right] \right), \tag{7.1}$$

where $\mathcal{T}_y^{(k)} \subset \mathcal{D}_y^{(k)}$ is the corresponding training dataset ($\mathcal{D}_y^{(k)}$ is the dataset of samples $\boldsymbol{x}$ drawn within region $\Gamma_y(\gamma_k^{\mathrm{bp}})$) and $\hat{y}^{(k)}_{\boldsymbol{\theta}^{(k)}}$ is the estimated class probability of the kth classifier with parameter set $\boldsymbol{\theta}^{(k)}$.[3] The error rate of the kth classifier can be estimated as

$$p_{\mathrm{err}}^{(k)} \approx \frac{1}{2} \sum_{y \in \{1,0\}} \frac{1}{|\mathcal{E}_y^{(k)}|} \sum_{\boldsymbol{x} \in \mathcal{E}_y^{(k)}} \mathrm{err}^{(k)} \left[\hat{y}^{(k)}_{\boldsymbol{\theta}^{(k)}}(\boldsymbol{x}) \right], \tag{7.2}$$

where for each sample $\boldsymbol{x}$ the error $\mathrm{err}^{(k)}$ is 0 if it is classified correctly and 1 if it is classified erroneously. As before, the continuous predictions $\hat{y}$ are converted to binary predictions via the Heaviside step function centered around 0.5. The evaluation set is denote as $\mathcal{E}_y^{(k)}$.

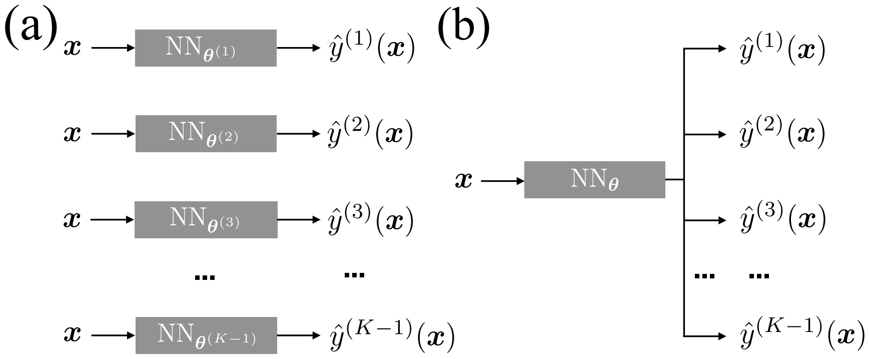
(a)
$\boldsymbol{x}$
$\mathrm{NN}_{\boldsymbol{\theta}^{(1)}}$
$\hat{y}^{(1)}(\boldsymbol{x})$
$\boldsymbol{x}$
$\mathrm{NN}_{\boldsymbol{\theta}^{(2)}}$
$\hat{y}^{(2)}(\boldsymbol{x})$
$\boldsymbol{x}$
$\mathrm{NN}_{\boldsymbol{\theta}^{(3)}}$
$\hat{y}^{(3)}(\boldsymbol{x})$
...
...
$\boldsymbol{x}$
$\mathrm{NN}_{\boldsymbol{\theta}^{(K-1)}}$
$\hat{y}^{(K-1)}(\boldsymbol{x})$
(b)
$\boldsymbol{x}$
$\mathrm{NN}_{\boldsymbol{\theta}}$
$\hat{y}^{(1)}(\boldsymbol{x})$
$\hat{y}^{(2)}(\boldsymbol{x})$
$\hat{y}^{(3)}(\boldsymbol{x})$
...
...
$\hat{y}^{(K-1)}(\boldsymbol{x})$

FIGURE 7.1: Schematic illustration of discriminative LBC for detecting phase transitions (a) with original single-task architecture and (b) with our proposed multi-task architecture.

[2] Having introduced the l parameter in Section 4.2.2, in this chapter, we consider the setting of $l \to \infty$, i.e., we always partition the entire sampled parameter range into two regions.

[3] In previous formulations of LBC, we typically suppressed the dependence on the bipartition point for convenience.

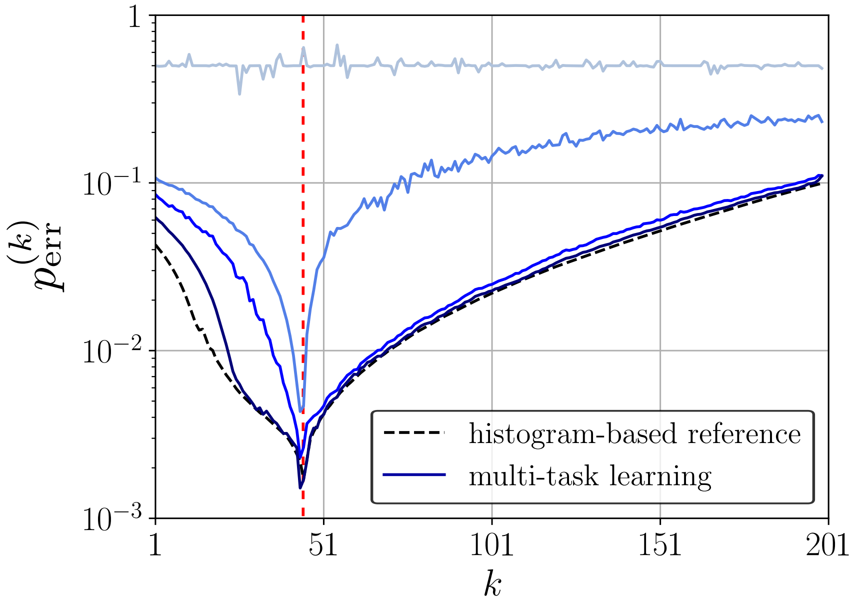

FIGURE 7.2: Estimated classification error for the Ising model (see Section 2.3.1) at each node using a single multi-task network trained on spin configurations. The solid blue lines represent the average results from 5 independent training runs after 0, 1, 5, and 50 training epochs in descending order from top to bottom. The black dashed line corresponds to an estimate of the (approximate) Bayes-optimal error rate obtained using a histogram-based generative classifier in energy space (see Chapter 4). The critical temperature from theory is highlighted by a red dashed line.

Multitasking

Instead of training a new classifier for each tentative splitting $k \in \{1, 2, \dots, K-1\}$, we propose to train a single classifier with weights $\boldsymbol{\theta}$ that has $K-1$ outputs

$$\left\{ \hat{y}^{(k)}_{\boldsymbol{\theta}^{(k)}} \right\}_{k=1}^{K-1} \mapsto \left\{ \hat{y}^{(k)}_{\boldsymbol{\theta}} \right\}_{k=1}^{K-1}, \tag{7.3}$$

cf. Figures 7.1(a) and (b). The loss function for this multi-task architecture is

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{K-1} \sum_{k=1}^{K-1} \mathcal{L}(\boldsymbol{\theta} | \gamma_k^{\mathrm{bp}}) \tag{7.4}$$

The evaluation of the error rate remains the same as before.

Multi-task learning [Caruana, 1997; Ruder, 2017] is expected to be highly efficient because the $K-1$ classification tasks are very similar to each other and only differ in a slight alteration of the tentative splitting of the sampled parameter range. As a result, the learned features are highly transferable between tasks. Note that one can always replace several networks that solve different tasks but operate on the same input space with a single network. However, this weight-sharing only provides a benefit if the data representations emerging within the joint network are useful for several tasks.

Consider the task of detecting phase transitions in a classical equilibrium system with raw spin configurations as input. In this case, the joint network may learn to extract the low-dimensional sufficient statistic as an intermediate representation, which is useful for all downstream classification tasks (recall our findings from Chapter 4). In multitasking, this sufficient statistic only needs to be learned once. In contrast, in the single-task approach to LBC, each distinct NN-based classifier may need to learn to extract this information separately.

Similar to multitasking, one also expects transfer learning techniques to provide a benefit. In transfer learning, one may, for example, utilize the weights of previously trained classifiers along the parameter range to initialize new ones.

7.3 Benchmark and applications

7.3.1 Ising model

As a benchmark system, we consider the two-dimensional square-lattice ferromagnetic Ising model. We consider a system of linear size $L = 60$ with 10^5 sampled spin configurations per tuning parameter value $\gamma = k_{\mathrm{B}}T/J$. The set Γ consists of 200 equally-spaced parameter values between 0.05 $k_{\mathrm{B}}T/J$ and 10 $k_{\mathrm{B}}T/J$. For a detailed description of the model and the data generation procedure, see Section 2.3.1. Figure 7.2 shows how the LBC signal of a multi-task CNN trained on this dataset evolves with training epochs. Eventually, the node achieving the lowest error rate coincides with the critical point.

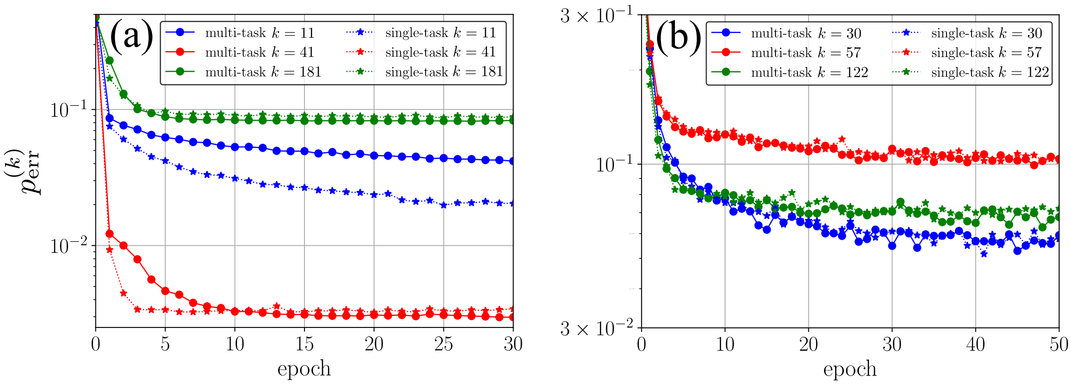

FIGURE 7.3: Error rate at representative nodes as a function of training epoch for single-task and multi-task LBC for (a) the Ising dataset averaged over 5 independent NN training runs and (b) the Stable Diffusion dataset averaged over 4 independent NN training runs. In the case of Stable Diffusion, we consider only one out of the three training and test sets, i.e., 16 training and 11 test images per year. Error bars derived from the standard deviation are negligible (i.e., on the same scale as markers).

As a classifier, we use a CNN with an input layer for single-channel 60 × 60 spin configurations, two convolutional layers with 16 and 32 kernels (kernel size: 2 × 2, stride: 1, padding: 1), average pooling (size: 2 × 2, stride: 2) after each convolution, and two fully connected layers, including a hidden layer with 128 units and ReLU activation functions. For training, we use Adam [Kingma and Ba, 2014] with a batch size of 1024 and a learning rate of 0.0001 (and default settings otherwise). We train for a total of 50 epochs. Here, the training and evaluation sets coincide with the entire dataset. No validation set is used. A Python implementation of the multi-task LBC procedure utilized in this chapter can be found at [Arnold *et al.*, 2023d].

To compare the single- and multi-task approach, let us compare the error rate at different network nodes, i.e., different locations in parameter space. Figure 7.3(a) shows how the error rate evolves as a function of the training epoch at nodes below, near, and above the critical point. In all cases, single-task LBC shows a slightly faster rate of convergence early on during training. Below the phase transition, multi-task LBC does not achieve an error rate as low as single-task LBC. In contrast, near and above the phase transition, multi-task LBC eventually catches up and even achieves lower error rates.

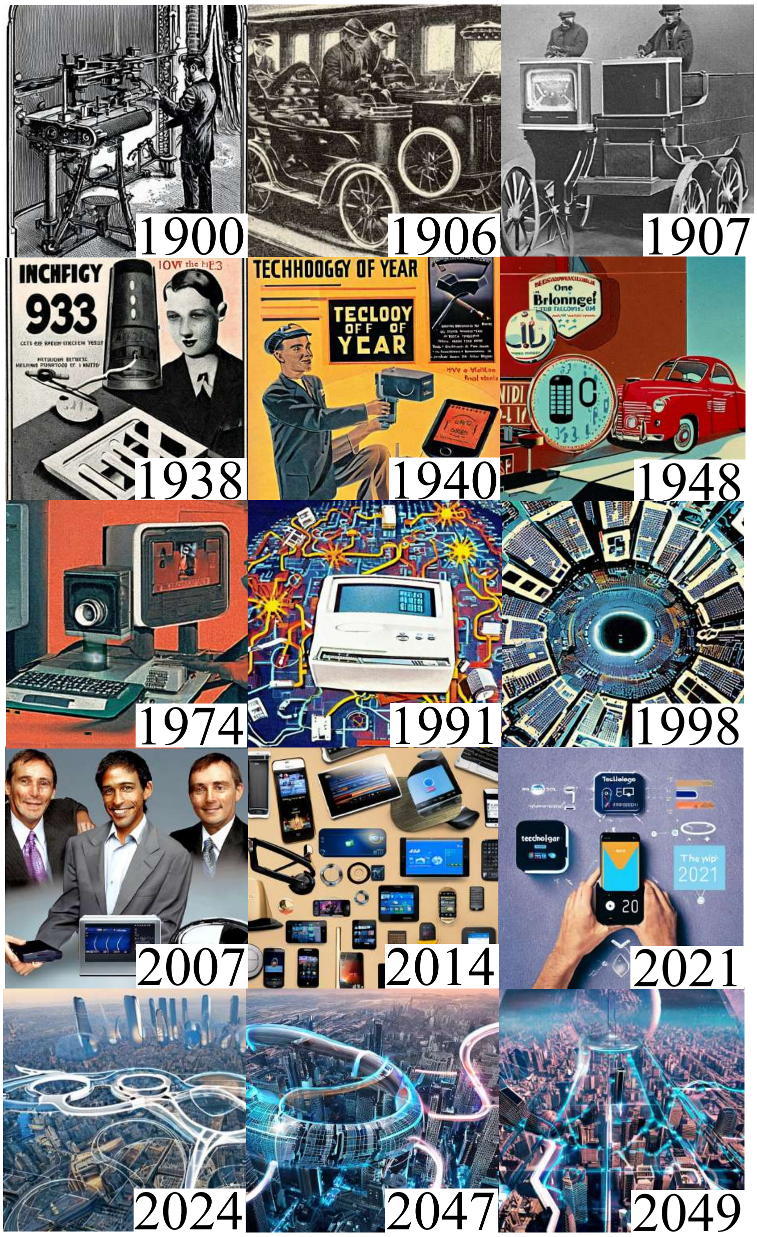

Figure 7.4: Selected images from the Stable Diffusion dataset generated with different values of γ, i.e., representative of different years.

We also studied the training behavior of multi-task networks which each have an output node corresponding to the true critical point as well as a varying number of additional output nodes. In particular, we recorded the number of epochs it takes to reach different error levels at their critical output node. At high levels, the difference in the number of epochs between different architectures was negligible. At low levels, i.e., for reaching an error rate close to Bayes' error, we observed the number of epochs to marginally increase with the number of nodes. However, we observed no clear scaling and any overhead was minor (at worst a factor of ≈ 6 for some combinations of training hyperparameters and model architectures) as compared to the speedup gained through multitasking.

7.3.2 Stable Diffusion

We now consider an image dataset generated using *Stable Diffusion* (version 2.1) [Rombach *et al.*, 2021], where for each integer γ in $[1900, 2050)$ images are sampled with the prompt "technology of the year γ". For example, node $k = 1$ corresponds to the bipartition point between the years 1900 and 1901. Figure 7.4 shows a selection of generated images across different years. While this dataset does not feature phase transitions in the physical sense, LBC can be used to identify points in parameter space at which the data distribution changes rapidly. Note that no prior analysis is available for this data and the predicted change points cannot be verified by theory.

Because these images have complex features comparable to typical image datasets, we load the pre-trained ResNet-50 CNN [He *et al.*, 2016] from PyTorch [Paszke *et al.*, 2019], and exchange its final layer to fit our task. The dataset contains $3 \cdot 27$ images per year, which we split into $3 \cdot 16$ and $3 \cdot 11$ images for training and evaluation, respectively. For training, we use Adam [Kingma and Ba, 2014] with a learning rate of 0.00005 and batch size of 256 (and default settings otherwise). We train for 150 epochs.

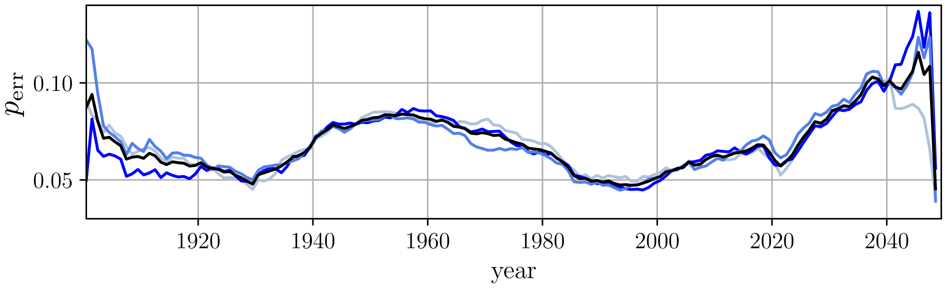

FIGURE 7.5: Error rate obtained using multi-task LBC for the Stable Diffusion dataset as a function of γ in prompt "technology of the year γ". The colored lines depict the results obtained by training on three separate data sets each with 16 training and 11 test images per year. Each colored line represents the average results of 10 independent NN training runs. The black line represents their overall mean.

Figure 7.5 shows (at least) three major local minima indicating rapid changes in the image dataset; one in-between the years 1929 and 1930, a second, broader one in the 1990s, and a third one in-between 2021 and 2022. Looking at the underlying image dataset (see Figure 7.4), the first minimum seems to coincide with a transition from grey images to colored images. The second transition is characterized by an increasing amount of images showing small electronic devices. The Stable Diffusion model has only encountered images of actual technology from before and around 2022 in its training dataset LAION-5B [Schuhmann *et al.*, 2022], which may explain the third minimum. Due to the small dataset size and the resulting challenges in generalization, the signal is expected to be less reliable close to the edges.

Figure 7.3(b) shows the error at representative points – the local extrema at nodes $k = 30$, $k = 57$, and $k = 122$ (corresponding to the bipartition points between years

1929-1930, 1956-1957, and 2021-2022) – as a function of the training epoch for a single-task and multi-task network with otherwise identical network architecture and training parameters. In this case, we find that there is no significant overhead, and the speedup of the multitasking compared to the single-task approach is approximately given by the number of bipartition points ($|\Gamma| - 1 = 149$).

7.4 Discussion

In the limit of infinite model capacity, both multi-task and single-task learning models ultimately yield the same predictions, because the multi-task loss corresponds to the aggregation of all the single-task losses [cf. Equation (7.4)]. However, in real-world scenarios where model expressivity, training time, and data are limited, the learning behavior depends on the particulars of the model and dataset at hand.

For the Ising dataset analyzed with a shallow 4-layer CNN, we observed some differences in predictions between the single-task and multi-task architectures, but not near the critical point where it would matter most. At the critical point, we found a minor overhead with respect to the ideal speedup linear in the number of sampled parameter points $|\Gamma|$.

The analysis of the Stable Diffusion dataset with the 50-layer ResNet-50 demonstrates the viability of LBC to reveal rapid changes in the distribution of complex datasets for which there does not exist any theoretical description. In this case, we found no signs of an overhead.

7.5 Summary

We find the multi-task implementation of the discriminative LBC scheme to provide much faster execution on large parameter grids as compared to its single-task version. As such, this chapter contributes a faster variant of a highly popular learning method for the data-driven detection of phase transitions.

7.6 Outlook

Here, we have restricted ourselves to one-dimensional parameter spaces. It remains an open problem to appropriately generalize the multitasking procedure to higher-dimensional parameter spaces. Similarly, in the future one may look to apply the multitasking approach with finite l.[4]

We have showcased that the output images of the Stable Diffusion generative model are highly structured and its changes can be detected using LBC. This result opens up a whole new application domain of methods for detecting phase transitions to datasets beyond physics. We will start to explore this domain in the second part of this thesis. In the context of the multitasking approach, datasets outside the realm of statistical physics are exciting, because the speedup is expected to be particularly impactful in these cases. This is because the analysis of such datasets typically requires a large amount of computational resources and probabilistic descriptions are scarce, requiring a discriminative approach.

Going beyond the first demonstration of the detection of change points in a diffusion model, the following questions remain to be answered in future investigations:

[4] Applications of the multitasking scheme for finite l will be briefly discussed in Chapter 9.

How do other tunable parameters, such as training or inference hyperparameters, influence the distribution underlying diffusion models? Are there differences between distinct diffusion models? Some diffusion models allow for the evaluation of the generation probability of a given image. Could this be utilized to map out its phase diagram in a generative fashion using the techniques from Chapter 4?

The results and figures presented in this chapter have been in parts published in [Arnold *et al.*, 2023c]. The corresponding code is open source [Arnold *et al.*, 2023d].

Part II

Venturing Beyond Physics

Chapter 8

Phase Transitions in Large Language Models

The results presented in this chapter are based on the following preprint:

Phase transitions in the output distribution of large language models,
J. Arnold, F. Holtorf, F. Schäfer, and N. Lörch,
arXiv:2405.17088 (2024).

8.1 Motivation

In the context of artificial intelligence, phase-transition-like phenomena have recently been observed in the learning behavior of NNs [Saitta *et al.*, 2011; Poole *et al.*, 2016; Shwartz-Ziv and Tishby, 2017; Gur-Ari *et al.*, 2018; McGrath *et al.*, 2022; Pan *et al.*, 2022; Ziyin and Ueda, 2022; Simon *et al.*, 2023; Cui *et al.*, 2024; Raventós *et al.*, 2024; Tamai *et al.*, 2023]. For example, during training, AlphaZero [Silver *et al.*, 2018] underwent periods of rapid knowledge acquisition in which increasingly sophisticated chess openings were favored by the engine [McGrath *et al.*, 2022]. Large language models (LLMs) have been observed to make sudden improvements in their inductive abilities during training which is related to the formation of special circuitry (so-called *induction heads*) [Olsson *et al.*, 2022]. Similar abrupt improvements in specific capabilities, often referred to as *breakthroughs*, have been observed for a variety of different models and tasks [Brown *et al.*, 2020; Hendrycks *et al.*, 2020; Ganguli *et al.*, 2022; Austin *et al.*, 2021; Radford *et al.*, 2021; Rae *et al.*, 2021; Srivastava *et al.*, 2022; Wei *et al.*, 2022; Pan *et al.*, 2022; Caballero *et al.*, 2022; Simon *et al.*, 2023]. Moreover, phenomena such as double descent [Belkin *et al.*, 2019; Nakkiran *et al.*, 2021] or grokking [Power *et al.*, 2022; Liu *et al.*, 2022a,b; Levi *et al.*, 2023; Nanda *et al.*, 2023; Rubin *et al.*, 2023] are also reminiscent of phase transitions in physics.

The detection of phase transitions[1] in deep learning systems may improve our understanding and eventually enable better model training. For example, an in-depth analysis of the grokking transition [Power *et al.*, 2022; Thilak *et al.*, 2022] led to a way for accelerating generalization [Liu *et al.*, 2022b]. Moreover, it has been shown that models are highly sensitive to perturbations, such as data corruption, at certain times during training [Achille *et al.*, 2019; Chen *et al.*, 2023]. Being able to predict the behavior of models is also crucial for ensuring safe model deployment [Ganguli *et al.*, 2022] as well as for projecting the performance of future model versions and optimally allocating resources for their training [Kaplan *et al.*, 2020].

[1] In the second part of this thesis, we adopt a more general and weaker definition of a phase transition as a sudden shift in the qualitative behavior of a system as a function of a control parameter [Saitta *et al.*, 2011; Pan *et al.*, 2022; Chen *et al.*, 2023]. For a more detailed discussion, see start of Section 8.2.

However, as in physics, the characterization of phase transitions in learning systems based on NNs is hard. NNs typically contain an enormous amount of trainable parameters and their state space, as characterized by their neural activations, is huge. This problem is exacerbated in generative models such as LLMs where the output space is also large, i.e., the high dimensionality cannot be foregone by treating the inside of the NN as a black box and focusing solely on its output characteristics. Understanding LLMs from first principles has been notoriously hard [Alishahi *et al.*, 2019]. Theories capturing their microscopic and macroscopic behavior, for instance based on *mechanistic interpretability* [Olah, 2022; Olsson *et al.*, 2022; Wang *et al.*, 2022; Nanda *et al.*, 2023; Zhong *et al.*, 2023] or *neural scaling laws* [Hestness *et al.*, 2017; Rosenfeld *et al.*, 2019; Kaplan *et al.*, 2020; Henighan *et al.*, 2020; Gordon *et al.*, 2021; Zhai *et al.*, 2022; Hoffmann *et al.*, 2022; Caballero *et al.*, 2022], are still nascent. In particular, so far, the definition of appropriate low-dimensional quantities that facilitate the detection of transitions has been done manually, for example through the extraction of appropriate circuit elements in NNs [Räuker *et al.*, 2023; Conmy *et al.*, 2023; Zhong *et al.*, 2023]. Due to this human-in-the-loop, transitions can be easily missed [Zhong *et al.*, 2023] or spuriously induced [Schaeffer *et al.*, 2023].

In the first part of this thesis, we have developed ML methods for detecting phase transitions from data with minimal prior system knowledge and human input. In this chapter, we adopt such an approach for the automated detection of phase transitions in LLMs. The method we propose is based on measuring changes in the distribution of the text output of LLMs via generic statistical distances belonging to the family of f-divergences (see Section 5.2), making it a versatile all-purpose tool for objectively and automatically mapping out phase diagrams of generative models. Such an approach has the potential to characterize unexplored phase transitions and discover new phases of behaviors. This is crucial in light of the rapid development of LLMs [Achiam *et al.*, 2023; Anthropic, 2023; Gemini Team *et al.*, 2023] and their emergent capabilities [Brown *et al.*, 2020; Hendrycks *et al.*, 2020; Rae *et al.*, 2021; Austin *et al.*, 2021; Radford *et al.*, 2021; Srivastava *et al.*, 2022; Ganguli *et al.*, 2022; Wei *et al.*, 2022; Olsson *et al.*, 2022; Pan *et al.*, 2022; Michaud *et al.*, 2023; Arora and Goyal, 2023].

As a demonstration, we characterize transitions occurring as a function of three different control parameters in Pythia [Biderman *et al.*, 2023], Mistral (7B) [Jiang *et al.*, 2023], and Llama3 (8B) [AI@Meta, 2024] language models: an integer occurring in the input prompt, the temperature hyperparameter for text generation, and the model's training epoch.

8.2 Methodology

In the second part of this thesis, we view phase transitions as rapid changes in the probability distribution $P(\cdot|\gamma)$ governing the state of the system $\boldsymbol{x} \sim P(\cdot|\gamma)$ as the control parameter γ is varied. That is, values of the parameter at which the distribution changes strongly are considered critical points where phase transitions occur. In the case of language models, $\boldsymbol{x}$ is the sampled text, and γ is any variable that influences the sampling probability. While it is possible to generalize our approach to distributions conditioned on multiple control parameters, cf. Chapter 4, for simplicity we consider the one-dimensional scenario in the following.

The colloquial definition of phase transitions as "rapid changes in $P(\cdot|\gamma)$" does encompass phase transitions as we know them in physics, where systems of interacting constituents exhibit abrupt changes in their governing distributions. In such physical systems, these changes become true non-analyticities in the thermodynamic limit of infinite system size. While the transitions remain smooth for any finite system, their sharpness increases systematically with system size, approaching discontinuous behavior as the size grows. In the second part of this thesis, we study systems where this limiting behavior is hardly accessible: they are finite in size and difficult to study analytically. Moreover, their size is typically fixed by practical constraints.[2] Whenever possible, we may try to perform a finite-size scaling analysis to confirm that the observed change is a phase transition in the proper sense. For the most part, however, we will study systems of a fixed size and what constitutes a *rapid* change will be rather vague. As in previous chapters, we are looking for changes concentrated locally in parameter space that are significant in comparison with the background level (i.e., with the rate of change across the rest of the parameter space or the local neighborhood).

How to quantify change

To quantify the rate of change we utilize f-divergences, as they have particularly nice properties, such as satisfying the data processing inequality. For a review on f-divergences and their properties, see Section 5.2. In particular, throughout this thesis the TV distance and the JS divergence have proven to successfully detect phase transitions in a variety of physical systems without prior system knowledge.[3],[4] Crucially, when comparing two distributions separated by $\Delta\gamma$, any f-divergence with f being twice-differentiable at 1 reduces to the Fisher information in lowest order of $\Delta\gamma$

$$D_f\left[P(\cdot|\gamma), P(\cdot|\gamma+\Delta\gamma)\right] = \tfrac{1}{2}f''(1)\mathcal{F}(\gamma)\Delta\gamma^2 + \mathcal{O}(\Delta\gamma^3). \tag{8.1}$$

Meaning that local changes in a distribution as measured by *any* such f-divergence reduce to the Fisher information in the limit $\Delta\gamma \to 0$.

Recovering the Fisher information as a limiting case is a desirable property. It is a well-known, generic statistical measure for quantifying how sensitive parameterized probability distributions are to changes in their parameters and its behavior is well-understood when used to detect phase transitions in physical systems. When considering parametric distributions underlying physical systems, the Fisher information can be shown to be directly related to important physical quantities, such as the heat capacity and magnetic susceptibility (see Chapter 6). Physicists routinely detect phase transitions in physical systems by identifying divergences (in the case of infinite-sized systems) or sharp local maxima (in the case of finite-sized systems) in these quantities. While they have no direct correspondence in the more abstract realm of language models, we may nevertheless port this methodology to the language domain allowing for the detection of analogous phenomena. The Fisher information provides a solid theoretical basis for applying concepts from statistical physics to LLMs, bridging the gap between physical systems and computational models.

[2] Stable Diffusion, for example, produces images of a fixed size. Similarly, we have no explicit control over the size of news articles published in The Guardian.

[3] In Chapter 6, we have shown that the Bayes-optimal indicator and loss value of LBC are related to the TV distance and JS divergence, respectively.

[4] Note that both the TV distance and the JS divergence form lower bounds to the KL divergence and other f-divergence, such as the χ^2 divergence: $D_{\mathrm{JS}}[p,q] \leq D_{\mathrm{TV}}[p,q] \leq \sqrt{D_{\mathrm{KL}}[p,q]} \leq \sqrt{D_{\chi^2}[p,q]}$ [Flammia and O'Donnell, 2024]. In this sense, detecting a large dissimilarity in terms of the TV distance or the JS divergence also signals a large dissimilarity in other measures.

Having discussed what may be appropriate notions of distance between probability distributions, let us describe their use to detect phase transitions in more detail. Consider a sampled set Γ of control parameter values γ, forming a uniform one-dimensional grid with spacing $\Delta\gamma$. For each bipartition point γ^{bp} lying halfway in between grid points, we define two sets of points $\Gamma_0(\gamma^{\text{bp}})$ and $\Gamma_1(\gamma^{\text{bp}})$ comprised of the l points closest to γ^{bp} with $\gamma < \gamma^{\text{bp}}$ and $\gamma > \gamma^{\text{bp}}$, respectively.[5] The probability distribution underlying these two regions can be constructed as

$$P(\cdot|y, \gamma^{\text{bp}}) = \frac{1}{|\Gamma_y(\gamma^{\text{bp}})|} \sum_{\gamma \in \Gamma_y(\gamma^{\text{bp}})} P(\cdot|\gamma) \tag{8.2}$$

for $y \in \{0, 1\}$. Critical points where phase transitions occur can then be identified as local maxima in[6]

$$D_f(\gamma^{\text{bp}}) = D_f\left[P(\cdot|0, \gamma^{\text{bp}}), P(\cdot|1, \gamma^{\text{bp}})\right] = \mathbb{E}_{\boldsymbol{x}\sim P(\cdot|1,\gamma^{\text{bp}})}\left[f\left(\frac{P(\boldsymbol{x}|0, \gamma^{\text{bp}})}{P(\boldsymbol{x}|1, \gamma^{\text{bp}})}\right)\right]. \tag{8.3}$$

The parameter l sets the natural length scale on which changes in the distributions are assessed. We are free to adjust it according to the problem, and examples will be discussed in Section 8.3. Note that with $l = 1$ and $\Delta\gamma \to 0$, we recover the relation to the Fisher information. In practice, however, choosing a somewhat larger value for l is found to be beneficial to average over local fluctuations.

Estimation of statistical distances

To detect phase transitions in a language model, the task that remains is to estimate the dissimilarity in Equation (8.3). Because of the autoregressive structure of language models, we can efficiently sample text $\boldsymbol{x}$ for a given prompt and evaluate its probability $P(\boldsymbol{x}|\gamma)$. As such, we have direct access to the probability distributions underlying the distinct regions in parameter space. For a given sample $\boldsymbol{x}$, we can evaluate the term $f\Big(P(\boldsymbol{x}|0, \gamma^{\text{bp}})/P(\boldsymbol{x}|1, \gamma^{\text{bp}})\Big)$ in Equation (8.3). Thus, we can obtain an unbiased estimate $\hat{D}_f$ of D_f by replacing expected values with sample means where samples correspond to text generated with language models conditioned on different parameter settings. We will discuss this numerical procedure in detail in Section 8.2.1.

To better analyze and compare the sampling efficiency and numerical stability of different choices of f-divergences, let us rewrite these dissimilarity measures in a slightly different form. To this end, when sampling around γ^{bp}, we use Bayes' theorem on Equation (8.2) to write the probability for a sample $\boldsymbol{x}$ to stem from segment $y \in \{0, 1\}$ as

$$P(y|\boldsymbol{x}, \gamma^{\text{bp}}) = \frac{P(\boldsymbol{x}|y, \gamma^{\text{bp}})}{P(\boldsymbol{x}|0, \gamma^{\text{bp}}) + P(\boldsymbol{x}|1, \gamma^{\text{bp}})}. \tag{8.4}$$

Note that this is the key quantity a classifier estimates when being trained to distinguish between samples belonging to the two distinct regions (such as in discriminative LBC).

[5] In this chapter, we choose to ignore bipartition points which lead to sets containing less than l points.

[6] We may divide this distance measure by the corresponding change in parameter space, $\Delta\gamma$, to capture a rate of change. However, since $\Delta\gamma$ is an overall constant we choose to ignore it.

Using $P(y|\boldsymbol{x}, \gamma^{\mathrm{bp}})$ as argument of a function g, we introduce a new family of dissimilarity measures

$$D_g(\gamma^{\mathrm{bp}}) = \frac{1}{2l} \sum_{y\in\{0,1\}} \sum_{\gamma\in\Gamma_y(\gamma^{\mathrm{bp}})} \mathbb{E}_{\boldsymbol{x}\sim P(\cdot|\gamma)} \left[g\left[P(y|\boldsymbol{x}, \gamma^{\mathrm{bp}}) \right] \right], \tag{8.5}$$

which we will refer to as g-dissimilarities. The g-dissimilarities [Equation (8.5)] and the f-divergences [Equation (8.3)] correspond to each other in the following sense: any g-dissimilarity $D_g(\gamma^{\mathrm{bp}})$ can be rewritten in the form of an f-divergence $D_f(\gamma^{\mathrm{bp}}) = D_f\left[P(\cdot|0, \gamma^{\mathrm{bp}}), P(\cdot|1, \gamma^{\mathrm{bp}})\right]$ with

$$f(x) = \frac{x}{2} \cdot g\left(\frac{x}{1+x}\right) + \frac{1}{2} \cdot g\left(\frac{1}{1+x}\right). \tag{8.6}$$

In particular, for the choice $g(x) = \ln(x) + \ln(2)$, D_g corresponds to the JS divergence. For $g(x) = 1 - 2\min\{x, 1-x\}$, D_g corresponds to the TV distance. We will discuss this correspondence in more detail in Section 8.2.2.

A natural choice for g is any linear function in x. In particular, setting $g(x) = 2x-1$ results in a dissimilarity measure that quantifies the ability of a Bayes-optimal classifier to tell whether a sample $\boldsymbol{x}$ has been drawn in region 0 or 1. This measure is 0 if the two distributions are completely indistinguishable and 1 if the two distributions are perfectly distinguishable. Moreover, $g(x) = 2x - 1$ has the property of being bounded between 1 and -1, where the edge values are attained for certain predictions (0 and 1), and the value 0 corresponds to uncertain predictions at 0.5. This results in a low variance and favorable convergence properties for the estimator $\hat{D}_{g(x)=2x-1}$. We will refer to $D_{g(x)=2x-1}$ as *linear dissimilarity* in what follows. This quantity is a valid f-divergence and reduces to the Fisher information in lowest non-vanishing order of the distance $\Delta\gamma$ between neighboring parameter points on the grid. In fact, any g-dissimilarity with $g(1/2) = 0$ and a twice-differentiable g-function can be shown to be proportional to the Fisher information in lowest order. We will show this in Section 8.2.2.

Summary

Our goal is to detect points in the parameter space of an LLM at which its output changes rapidly, i.e., points at which phase-transition-like phenomena occur. To this end, we quantify the dissimilarity between the distributions underlying neighboring regions in parameter space using statistical distances, particularly the class of g-dissimilarities. For each valid bipartition point γ^{bp} splitting the parameter space into two such neighboring regions, we construct an estimate for the g-dissimilarity between the two segments $\hat{D}_g(\gamma^{\mathrm{bp}})$ from samples of the LLM. Repeating this for all valid bipartition points γ^{bp} within the sampled parameter range, critical points can be identified as local maxima in $\hat{D}_g$.

8.2.1 Implementation details

Starting with a fixed set of points on a uniform grid Γ, let us denote the set of in-between points at least l points away from the border of the range as Γ', where $|\Gamma'| = |\Gamma| - 2l$. For each trial point $\gamma^{\mathrm{bp}} \in \Gamma'$, we obtain an unbiased estimate

$$\hat{D}_g(\gamma^{\mathrm{bp}}) = \frac{1}{2}\left[\hat{J}_0(\gamma^{\mathrm{bp}}) + \hat{J}_1(\gamma^{\mathrm{bp}})\right] \tag{8.7}$$

with

$$\hat{J}_y(\gamma^{\mathrm{bp}}) = \frac{1}{l} \sum_{\gamma \in \Gamma_y(\gamma^{\mathrm{bp}})} \frac{1}{|\mathcal{D}_\gamma|} \sum_{\boldsymbol{x} \in \mathcal{D}_\gamma} g\left[P(y|\boldsymbol{x}, \gamma^{\mathrm{bp}})\right]. \tag{8.8}$$

Here, $\mathcal{D}_\gamma$ denotes a set of output texts $\boldsymbol{x}$ generated via the LLM at point $\gamma \in \Gamma$. In this chapter, we choose the number of generated text samples $|\mathcal{D}_\gamma|$ to be the same for all $\gamma \in \Gamma$.

In practice, we perform the computation of $\hat{D}_g(\gamma^{\mathrm{bp}})$ for all $\gamma^{\mathrm{bp}} \in \Gamma'$ in two stages. In the first stage, we go through each grid point $\gamma \in \Gamma$ and generate text outputs that are N_{tokens} in length via the LLM. The associated computation time scales as $|\Gamma| \cdot N_{\text{tokens}} \cdot t_{\text{LLM,eval}}(|\mathcal{D}_\gamma|)$, where $t_{\text{LLM,eval}}(|\mathcal{D}_\gamma|) = O(|\mathcal{D}_\gamma|)$ corresponds to the time it takes the LLM to generate $|\mathcal{D}_\gamma|$ different outputs (single token in length).

In a second stage, for a given trial point $\gamma^{\mathrm{bp}} \in \Gamma'$, we evaluate the probability of each text output generated in its vicinity $\boldsymbol{x} \in \{\boldsymbol{x} \in \mathcal{D}_\gamma | \gamma \in \Gamma_0(\gamma^{\mathrm{bp}}) \cup \Gamma_1(\gamma^{\mathrm{bp}})\}$ to come from segment $y \in \{0, 1\}$. That is, we compute $P(y|\boldsymbol{x}, \gamma^{\mathrm{bp}})$, corresponding to a term in the sum of Equation (8.8), using Equations (8.2) and (8.4), where we have access to $P(\boldsymbol{x}|\gamma)$ for any $\gamma \in \Gamma$ due to the autoregressive property of the LLM. The computation time associated with the second stage scales as $|\Gamma'| \cdot N_{\text{tokens}} \cdot t_{\text{LLM,eval}}(|\mathcal{D}_\gamma|) \cdot 2l$. Note that in practice, one can embarrassingly parallelize over the different grid and trial (i.e., bipartition) points. Moreover, one generates and evaluates text outputs batch-wise.

8.2.2 Theoretical background on g-dissimilarities

Correspondence between f-divergences and g-dissimilarities

Let us establish a correspondence between f-divergences [Equation (8.3)] and g-dissimilarities [Equation (8.5)]. Suppressing the dependence on γ^{bp} for notational clarity, we can write any g-dissimilarity as

$$\begin{aligned}
D_g &= \frac{1}{2}\left(\mathbb{E}_{\boldsymbol{x}\sim P(\cdot|0)}\left[g\left[P(0|\boldsymbol{x})\right]\right] + \mathbb{E}_{\boldsymbol{x}\sim P(\cdot|1)}\left[g\left[P(1|\boldsymbol{x})\right]\right]\right) \\
&= \mathbb{E}_{\boldsymbol{x}\sim P(\cdot|1)}\left[\frac{1}{2}\frac{P(\boldsymbol{x}|0)}{P(\boldsymbol{x}|1)} g\left[P(0|\boldsymbol{x})\right]\right] + \mathbb{E}_{\boldsymbol{x}\sim P(\cdot|1)}\left[\frac{1}{2} g\left[P(1|\boldsymbol{x})\right]\right] \\
&= \mathbb{E}_{\boldsymbol{x}\sim P(\cdot|1)}\left[\frac{1}{2}\frac{P(\boldsymbol{x}|0)}{P(\boldsymbol{x}|1)} g\left[P(0|\boldsymbol{x})\right] + \frac{1}{2} g\left[P(1|\boldsymbol{x})\right]\right] \\
&= \mathbb{E}_{\boldsymbol{x}\sim P(\cdot|1)}\left[\frac{1}{2}\frac{P(\boldsymbol{x}|0)}{P(\boldsymbol{x}|1)} g\left(\frac{P(\boldsymbol{x}|0)}{P(\boldsymbol{x}|0)+P(\boldsymbol{x}|1)}\right) + \frac{1}{2} g\left(\frac{P(\boldsymbol{x}|1)}{P(\boldsymbol{x}|0)+P(\boldsymbol{x}|1)}\right)\right] \\
&= \mathbb{E}_{\boldsymbol{x}\sim P(\cdot|1)}\left[\frac{1}{2}\frac{P(\boldsymbol{x}|0)}{P(\boldsymbol{x}|1)} g\left(\frac{\frac{P(\boldsymbol{x}|0)}{P(\boldsymbol{x}|1)}}{1+\frac{P(\boldsymbol{x}|0)}{P(\boldsymbol{x}|1)}}\right) + \frac{1}{2} g\left(\frac{1}{1+\frac{P(\boldsymbol{x}|0)}{P(\boldsymbol{x}|1)}}\right)\right].
\end{aligned} \tag{8.9}$$

Thus, any g-dissimilarity D_g can be rewritten *in the form* of an f-divergence

$$D_f[P(\cdot|0), P(\cdot|1)] \tag{8.10}$$

with

$$f(x) = \frac{x}{2} \cdot g\left(\frac{x}{1+x}\right) + \frac{1}{2} \cdot g\left(\frac{1}{1+x}\right). \tag{8.11}$$

However, not all choices of g-functions will lead to a *proper* f-divergence: the resulting f-function may not be convex, and $f(1)$ may not be zero (recall the definition of an f-divergence in Section 5.2).

Jensen-Shannon divergence as g-dissimilarity

Using the correspondence above, we have that $D_{g(x)=\ln(x)+\ln(2)}$ is equivalent to an f-divergence $D_f[P(\cdot|0), P(\cdot|1)]$ with

$$f(x) = \frac{x}{2} \cdot \ln\left(\frac{2x}{1+x}\right) + \frac{1}{2} \cdot \ln\left(\frac{2}{1+x}\right), \tag{8.12}$$

which corresponds to the JS divergence, see Equation (5.4).

Total variation distance as g-dissimilarity

Let us further show that the $D_{g(x)=1-2\min\{x,1-x\}}$ corresponds to the TV distance $D_{\mathrm{TV}}[P(\cdot|0), P(\cdot|1)]$. We have

$$\begin{aligned} D_{g(x)=1-2\min\{x,1-x\}} = 1 &- \mathbb{E}_{\boldsymbol{x}\sim P(\cdot|0)}\left[\min\Big\{P(0|\boldsymbol{x}), P(1|\boldsymbol{x})\Big\}\right] \\ &- \mathbb{E}_{\boldsymbol{x}\sim P(\cdot|1)}\left[\min\Big\{P(0|\boldsymbol{x}), P(1|\boldsymbol{x})\Big\}\right]. \end{aligned} \tag{8.13}$$

Using the identity

$$\min\Big\{P(0|\boldsymbol{x}), P(1|\boldsymbol{x})\Big\} = \frac{1}{2}\Big(1 - \Big|P(0|\boldsymbol{x}) - P(1|\boldsymbol{x})\Big|\Big), \tag{8.14}$$

Equation (8.13) can be rewritten as

$$\begin{aligned} D_{g(x)=1-2\min\{x,1-x\}} &= \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim P(\cdot|0)}\left[\Big|P(0|\boldsymbol{x}) - P(1|\boldsymbol{x})\Big|\right] + \\ &\quad \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim P(\cdot|1)}\left[\Big|P(0|\boldsymbol{x}) - P(1|\boldsymbol{x})\Big|\right] \\ &= \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim P(\cdot|1)}\left[\left(1 + \frac{P(\boldsymbol{x}|0)}{P(\boldsymbol{x}|1)}\right)\Big|P(0|\boldsymbol{x}) - P(1|\boldsymbol{x})\Big|\right] \\ &= \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim P(\cdot|1)}\left[P(1|\boldsymbol{x})\left(1 + \frac{P(\boldsymbol{x}|0)}{P(\boldsymbol{x}|1)}\right)\left|1 - \frac{P(0|\boldsymbol{x})}{P(1|\boldsymbol{x})}\right|\right]. \end{aligned} \tag{8.15}$$

Noting that $\frac{P(0|\boldsymbol{x})}{P(1|\boldsymbol{x})} = \frac{P(\boldsymbol{x}|0)}{P(\boldsymbol{x}|1)}$ and $P(0|\boldsymbol{x}) + P(1|\boldsymbol{x}) = 1$, we finally obtain

$$D_{g(x)=1-2\min\{x,1-x\}} = \mathbb{E}_{\boldsymbol{x}\sim P(\cdot|1)}\left[\frac{1}{2}\left|1 - \frac{P(\boldsymbol{x}|0)}{P(\boldsymbol{x}|1)}\right|\right], \tag{8.16}$$

which corresponds to an f-divergence $D_f[P(\cdot|0), P(\cdot|1)]$ with $f(x) = \frac{1}{2}|1-x|$, i.e., the TV distance.

Freedom in the choice of g-function

Note that the choice of g-function leading to a particular g-dissimilarity is not unique. In particular, we have that $D_{\tilde{g}} = D_g$ for any

$$\tilde{g}(x) = g(x) + c\left(\frac{1}{x} - 2\right), \tag{8.17}$$

where $c \in \mathbb{R}$ is some constant:

$$\begin{aligned}
D_{\tilde{g}} &= D_g + \frac{c}{2}\left(\mathbb{E}_{\boldsymbol{x}\sim P(\cdot|0)}\left[\frac{1-P(0|\boldsymbol{x})}{P(0|\boldsymbol{x})} - 1\right] + \mathbb{E}_{\boldsymbol{x}\sim P(\cdot|1)}\left[\frac{1-P(1|\boldsymbol{x})}{P(1|\boldsymbol{x})} - 1\right]\right)\\
&= D_g + \frac{c}{2}\left(\mathbb{E}_{\boldsymbol{x}\sim P(\cdot|0)}\left[\frac{P(1|\boldsymbol{x})}{P(0|\boldsymbol{x})} - 1\right] + \mathbb{E}_{\boldsymbol{x}\sim P(\cdot|1)}\left[\frac{P(0|\boldsymbol{x})}{P(1|\boldsymbol{x})} - 1\right]\right)\\
&= D_g + \frac{c}{2}\Big(\mathbb{E}_{\boldsymbol{x}\sim P(\cdot|0)}[1] + \mathbb{E}_{\boldsymbol{x}\sim P(\cdot|1)}[1] - 2\Big) = D_g.
\end{aligned} \tag{8.18}$$

Relation to the Fisher information

In the following, we show that any g-dissimilarity with $g(1/2) = 0$ and a twice-differentiable g-function reduces to the Fisher information in lowest order in $\Delta\gamma$, which is the distance between the regions in parameter space that are being compared. For this, consider the case with $l = 1$ where we compare the distributions at two points in parameter space. The corresponding g-dissimilarity can be written in the form of an f-divergence

$$D_f[P(\cdot|0), P(\cdot|1)] = D_f[P(\cdot|\gamma), P(\cdot|\gamma + \Delta\gamma)]. \tag{8.19}$$

Recall from Chapter 5 that

$$D_f[P(\cdot|\gamma), P(\cdot|\gamma)] = 0 \tag{8.20}$$

if $f(1) = 0$ and

$$\left.\frac{\partial D_f[P(\cdot|\gamma), P(\cdot|\phi)]}{\partial \phi}\right|_{\phi=\gamma} = f'(1)\frac{\partial \mathbb{E}_{\boldsymbol{x}\sim P(\cdot|\gamma)}[1]}{\partial \gamma} = f'(1)\frac{\partial 1}{\partial \gamma} = 0. \tag{8.21}$$

The second-order derivative corresponds to

$$\begin{aligned}
\left.\frac{\partial^2 D_f[P(\cdot|\gamma), P(\cdot|\phi)]}{\partial \phi^2}\right|_{\phi=\gamma} &= f'(1)\mathbb{E}_{\boldsymbol{x}\sim P(\cdot|\gamma)}\left[\frac{1}{P(\cdot|\gamma)}\frac{\partial^2 P(\cdot|\gamma)}{\partial \gamma^2}\right] +\\
&\quad f''(1)\mathbb{E}_{\boldsymbol{x}\sim P(\cdot|\gamma)}\left[\left(\frac{\partial \ln\left[P(\cdot|\gamma)\right]}{\partial \gamma}\right)^2\right]\\
&= f''(1)\mathcal{F}(\gamma)
\end{aligned} \tag{8.22}$$

assuming $f'(1) = 0$, where $\mathcal{F}$ is the Fisher information.

Thus, for $l = 1$ and parameter values separated by $\Delta\gamma$, we can express any g-dissimilarity with $g(1/2) = 0$ as

$$D_g = \frac{g''(\frac{1}{2})}{32}\mathcal{F}(\gamma)\Delta\gamma^2 + \mathcal{O}(\Delta\gamma^3). \tag{8.23}$$

The fact that $f'(1)$ must be zero translates into the condition that $g'(1/2) = 0$ is not a fundamental restriction: we have some freedom in the choice of g function as described above. That is, we can replace $g \mapsto \tilde{g}$, where $\tilde{g}(x) = g(x) + c\left(\frac{1}{x} - 2\right)$ with $c = g'(1/2)/6$, retaining $D_g = D_{\tilde{g}}$ and ensuring that $\tilde{g}'(1/2) = 0$.

8.2.3 Utilized large language models

In this chapter, we study transitions in models of the Pythia, Mistral, and Llama families.

Pythia

Pythia [Biderman *et al.*, 2023] is a suite of 16 LLMs released in 2023 that were trained on public data in the same reproducible manner. The models range from 70 million (M) to 12 billion (B) parameters in size.

Mistral

From the Mistral family, we consider the base model Mistral-7B-v0.1 with 7.3B parameters and the corresponding fine-tuned Mistral-7B-Instruct model [Jiang *et al.*, 2023] released in 2023.

Llama

Llama 3 [AI@Meta, 2024] from Meta AI was released in 2024. We consider both the Llama-3 8B parameter base model and NVIDIA's chat-tuned Llama3-ChatQA-1.5-8B model [Liu *et al.*, 2024].

8.3 Results

We can distinguish between three fundamental ways in which a parameter γ may influence the output distribution of a language model:

i) As a variable within the input prompt. To this end, we scan through integers injected to the prompt in Section 8.3.1.

ii) As a hyperparameter controlling how a trained language model is applied at inference time. To this end, we vary the temperature in Section 8.3.2.

iii) As a training hyperparameter of the language model. To this end, we vary the number of training epochs in Section 8.3.3.

8.3.1 Transitions as a function of a variable in the prompt

First, we consider the parameter γ to be varied a part of the prompt, and all parameters of the language model itself are fixed. As a simple example, we start by considering the prompt *"γ is larger than 42. True or False?"* with an integer $\gamma \in \mathbb{N}$ as the control parameter. An LLM that understands the order of integers should output very different answers for $\gamma < 42$ versus $\gamma > 42$, i.e., its distribution over outputs should change drastically around $\gamma = 42$. Thus, in such a case we expect the dissimilarities to show a clear peak around $\gamma = 42$.

Figure 8.1(a) shows dissimilarities based on various g-functions for the Mistral-7B-Instruct model [Jiang *et al.*, 2023]. All dissimilarities show a clear peak around $\gamma = 42$, whereas they are relatively flat around 0 otherwise. This is a clear example of an abrupt transition between two distinct phases of behaviors of an LLM as a function

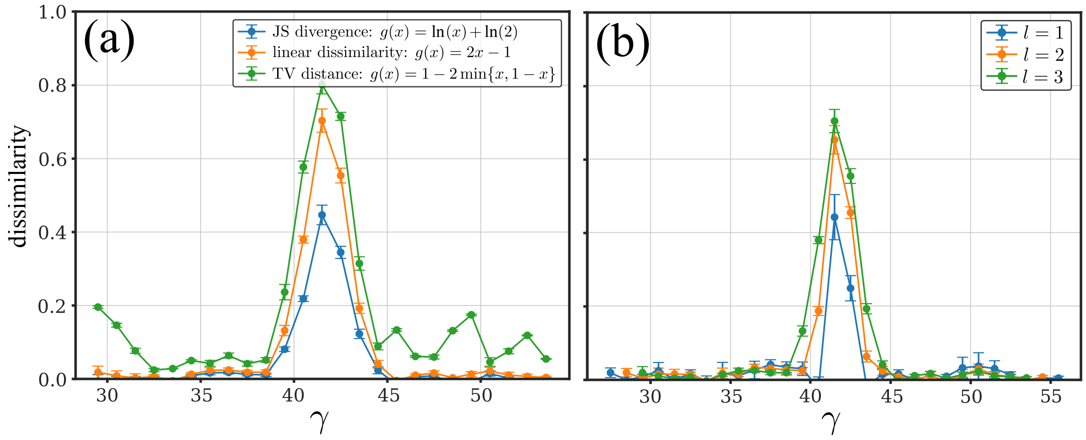

FIGURE 8.1: Mistral-7B-Instruct model applied to the integer ordering prompt *"γ is larger than 42. True or False?"*. (a) Different g-dissimilarities with $l = 3$. (b) Linear dissimilarity for different l-values. [Number of text outputs generated per parameter value γ: $|\mathcal{D}_\gamma| = 10280$. Number of generated output tokens: $N_{\text{tokens}} = 10$. Error bars indicate standard error of the mean over 4 batches, each with batch size 2056.] Similar results are obtained when using numbers different from 42 within the prompt and different semantic formulations of the prompt, see Figure H.1 in Appendix H.

of a tunable parameter. As compared to the linear dissimilarity, the logarithm-based JS divergence is arguably a bit sharper in that it decays more rapidly to baseline 0. The peak in the TV distance is the broadest due to the $\min\{\cdot\}$ operation appearing in its g-function. For the remainder of this chapter, we will focus on the linear dissimilarity as a compromise between sensitivity and numerical stability.

The transition is also clearly visible using different l settings, see Figure 8.1(b). Smaller l values are closer to the Fisher information limit, while larger values generally lead to higher distinguishability of distributions and therefore larger peaks at transition points. As we will see in more detail in Section 8.3.3, choosing larger l values also makes the dissimilarity measure less susceptible to outliers and sampling noise due to the averaging over several neighboring points in parameter points.

Interestingly, when performing the same analysis on base models such as the Llama3-8B and Mistral base models, as well as Pythia models [Biderman *et al.*, 2023] of various sizes, the resulting linear dissimilarity is flat, signaling the absence of any transition [see Figure 8.2(a)]. In contrast to Mistral-7B-Instruct and NVIDIA's chat-tuned Llama3-8B, these models do not show a clear peak around $\gamma = 42$. This points toward the fact that fine-tuning, i.e., alignment or instruction-tuning, is crucial for language models to be able to order integers.

A transition of a different origin can be observed in Figure 8.2(b), where the LLMs are probed using the prompt *"γ"* with γ again being an integer. Interestingly, all Pythia models show a peak between $\gamma = 2020$ and $\gamma = 2021$. This behavioral transition may originate from a transition in the tokenizers of these models, which encode numbers in a range below $\gamma = 2021$ with a single token and numbers in a range at and above $\gamma = 2021$ with two. This explanation is corroborated by the absence of the transition around $\gamma = 2021$ for the Llama and Mistral models, whose tokenizers translate a number into tokens following rules that are independent of the number's frequency.

The Mistral models and the base Llama3-8B model show a smaller peak around

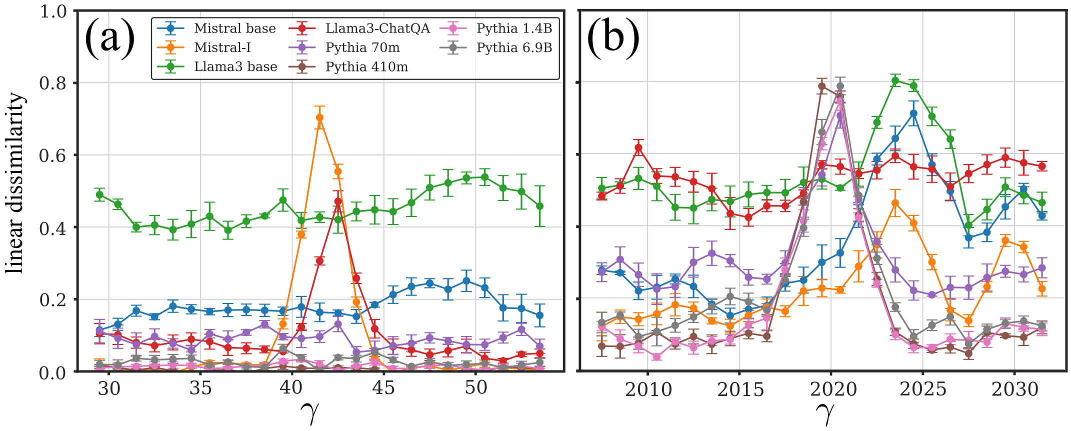

FIGURE 8.2: Benchmarking of various models using the linear dissimilarity with $l = 3$. (a) Test of ability to compare integers in value via the prompt "*γ is larger than 42. True or False?*". (b) Bare integers as prompt "γ" reveal transition in tokenizer encoding. [Same numerical settings as in Figure 8.1.]

$\gamma = 2023/2024$. Both models have only encountered training data from before and around that time given their release date in 2023/2024, which may explain the peak [Razeghi *et al.*, 2022]. This transition is absent in the Pythia models.

8.3.2 Transitions as a function of the model's temperature

Next, we consider transitions as a function of the temperature hyperparameter $\gamma = T$ controlling how the model's logits $\boldsymbol{z}$ are converted to probabilities

$$p_i = \frac{e^{z_i/T}}{\sum_j e^{z_j/T}} \tag{8.24}$$

for next-token prediction. The sum runs over all possible tokens $|\mathcal{X}|$. Per construction, at $T = 1$ language models predict probabilities $\{p_i\}_i$ to approximate the distribution to be learned. In the limit $T \to 0$, the model deterministically picks the most likely next token in each step. In contrast, for $T \to \infty$, the model samples the next token uniformly.

Relation to model systems from physics

This scenario resembles a one-dimensional lattice of spins that are coupled via long-range interactions, i.e., the one-dimensional Ising model (or Potts model if we consider spin variables with more than two possible states) [Dyson, 1969; Martínez-Herrera *et al.*, 2022], which has an order-disorder phase transition for couplings of infinite range.

In our case, the tokens take the role of the spins, and the coupling is mediated via the transformer's attention mechanism. The interaction only occurs in the forward direction for autoregressive models and is of finite range due to the finite context length of current models. While this cannot give rise to a strict discontinuity, rapid changes qualify for our weaker definition of a phase transition, and by scaling the system size appropriately one may observe a similar critical behavior.

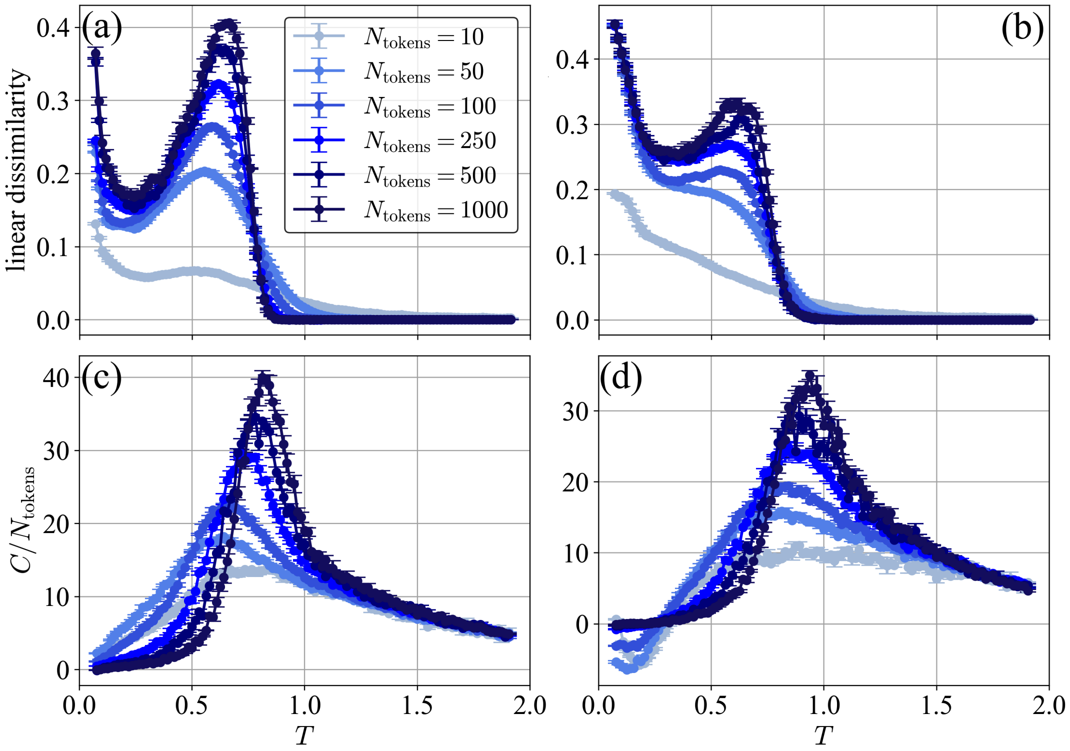

FIGURE 8.3: High-temperature transition of Pythia 70M model in response to (a),(c) the prompt *"There's measuring the drapes, and then there's measuring the drapes on a house you haven't bought, a"* and (b),(d) *"The opinions expressed by columnists are their own and do not represent the views of Townhall.com.\n\n"*. Both prompts are excerpts from OpenWebText. (a),(b) Linear dissimilarity measure ($l = 5$) and (c),(d) heat capacity for various numbers of generated output tokens N_{tokens} with the temperature range $[10^{-4}, 2]$. [Number of text outputs generated per parameter value T: $|\mathcal{D}_T| = 5000$, 5000, 5000, 1500, 500, and 500 for $N_{\text{tokens}} = 10$, 50, 100, 250, 500, and 1000, respectively. Error bars indicate the standard error of the mean over 5 batches, each of size $|\mathcal{D}_T|/5$]

Behavior of linear dissimilarity

Figures 8.3(a) and (b) show the linear dissimilarity as a function of temperature for the Pythia 70M model prompted with two different text excerpts from OpenWebText [Aaron Gokaslan and Vanya Cohen, 2019] (which serves as a proxy for the Pythia training dataset) for various lengths of the generated output, i.e., N_{tokens}. In both cases, the linear dissimilarity shows two distinct peaks corresponding to two transition points: one at a very low temperature $T_{\text{c},1} \approx 0.02$ and one at an intermediate temperature $T_{\text{c},2} \approx 0.75$. The behavior we observe as a function of N_{tokens} is analogous to one of the critical phenomena observed in physical systems: As the system size (i.e., the output length N_{tokens}) increases from 10 to 1000 tokens, we see the peaks becoming sharper and larger in height. This sharpening of the peak with increasing system size is reminiscent of finite-size scaling effects seen near critical points in models from statistical physics. Note that the locations of the critical points are still subject to strong finite-size effects and may, in general, be dependent on the prompt. In particular, we observe that $T_{\text{c},2}$ seems to increase when increasing N_{tokens}. The

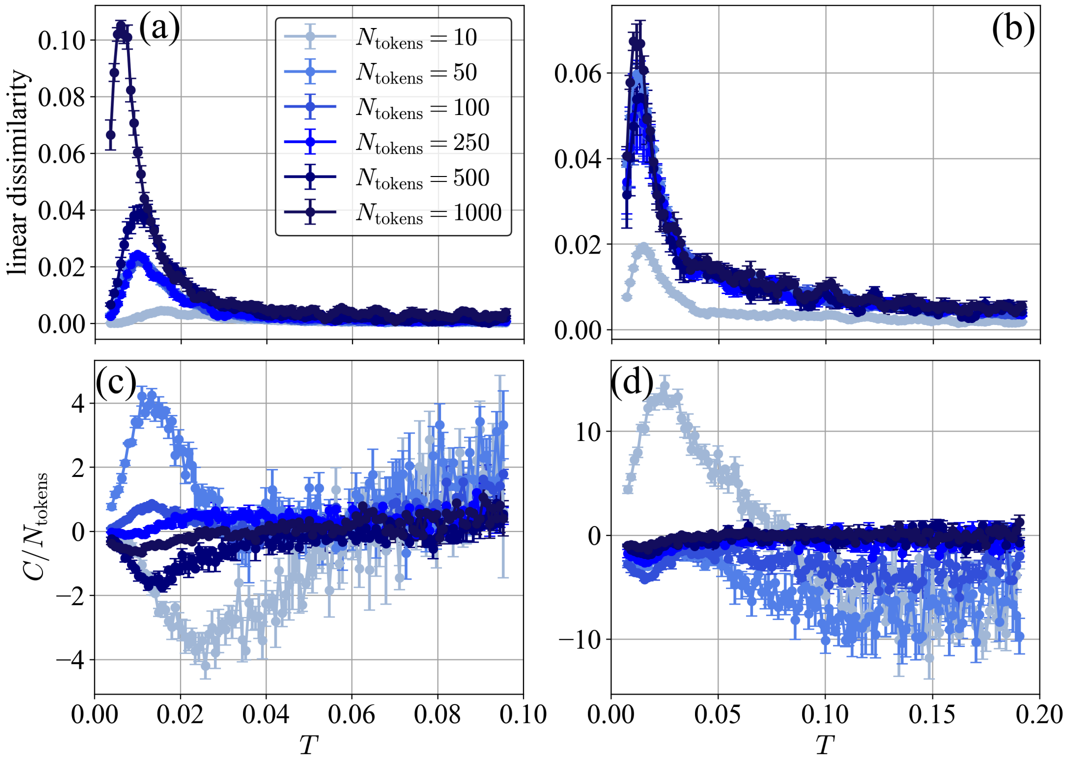

FIGURE 8.4: Low-temperature transition of Pythia 70M model in response to (a),(c) the prompt *"There's measuring the drapes, and then there's measuring the drapes on a house you haven't bought, a"* and (b),(d) *"The opinions expressed by columnists are their own and do not represent the views of Townhall.com.\n\n"*. Both prompts are excerpts from OpenWebText. The temperature ranges are (a),(c) $[10^{-4}, 10^{-1}]$ and (b),(d) $[10^{-4}, 2\cdot 10^{-1}]$. (a),(b) Linear dissimilarity measure ($l = 5$) and (c),(d) heat capacity for various numbers of generated output tokens N_{tokens}. [Number of text outputs generated per parameter value T: $|\mathcal{D}_T| = 5000$, 5000, 5000, 1500, 500, and 500 for $N_{\text{tokens}} = 10$, 50, 100, 250, 500, and 1000, respectively. Error bars indicate the standard error of the mean over 5 batches, each of size $|\mathcal{D}_T|/5$.]

low-temperature transition is shown in more detail in Figures 8.4(a) and (b).[7] Here, $T_{c,1}$ decreases as N_{tokens} increases.

Behavior of heat capacity

For comparison and to confirm the observed behavior we present a second analysis, independent of the dissimilarity-based indicators. It is based on taking inspiration from statistical mechanics, where the state of thermal systems is governed by the Boltzmann distribution. We want to view an LLM as such a thermal system at varying temperatures: Let $\boldsymbol{x} = (x_1, \ldots, x_{N_{\text{tokens}}})$ be a sequence of N_{tokens} tokens generated for a fixed prompt from an autoregressive LLM such as the ones considered in this article.

[7] The discrepancy between the linear dissimilarity in the high-temperature scan in Figure 8.3 and the linear dissimilarity in the low-temperature scan in Figure 8.4 arises from using different grid spacings ΔT.

The distribution of $\boldsymbol{x}$ is given by

$$P(\boldsymbol{x}|T) = Q(x_{N_{\text{tokens}}}|x_1,\ldots,x_{N_{\text{tokens}}-1};T)Q_T(x_{N_{\text{tokens}}-1}|x_1,\ldots,x_{N_{\text{tokens}}-2};T) \\ \cdots Q(x_2|x_1;T)Q(x_1;T), \tag{8.25}$$

denoting the fact that the tokens are sampled sequentially. In each step, a token is sampled from a Boltzmann distribution Q at temperature T,

$$Q(x_i|x_1,\ldots,x_{i-1};T) = e^{-E(x_i|x_1,\ldots,x_{i-1})/T}/Z_i(T). \tag{8.26}$$

Here, $Z_i(T) = \sum_{x_i} e^{-E(x_i|x_1,\ldots,x_{i-1})/T}$ is a normalization factor with the sum running over all possible ith tokens. The conditional energies, $E(x_i|x_1,\ldots,x_{i-1})$ are typically referred to as logits and learned from data. Note that while the distribution over individual tokens can be expressed as a Boltzmann distribution at varying temperatures, the overall distribution $P(\cdot|T)$ cannot – in order for a quantity to be a valid energy of a system, it cannot itself depend on temperature, i.e., change with temperature.

Nevertheless, we can define an energy scale for the entire system by viewing the overall probability distribution at $T = 1$ as a Boltzmann distribution

$$P(\boldsymbol{x}|T=1) = e^{-E(\boldsymbol{x})}/Z, \tag{8.27}$$

where Z is a normalization constant independent of $\boldsymbol{x}$ (the partition function). Recall that any valid probability distribution can be written in the form of Equation (8.27) with a suitably chosen energy function. Taking the logarithm of Equation (8.27) and reordering, we have

$$E(\boldsymbol{x}) = -\ln\left[P(\boldsymbol{x}|T=1)\right] - \ln(Z). \tag{8.28}$$

Using Equation (8.28), we can compute the total energy up to the constant $-\ln(Z)$ which serves as our reference point for the energy scale.

Now, let us view the LLM as a thermal system at varying temperatures where the negative logarithmic probability at $T = 1$, $-\ln\left[P(\boldsymbol{x}|T=1)\right]$, takes on the role of the energy E of a given text output $\boldsymbol{x}$. In physical systems governed by Boltzmann distributions, thermal phase transitions can be detected as peaks in the heat capacity $C(T) = \partial\mathbb{E}_{\boldsymbol{x}\sim P(\cdot|T)}\left[E(\boldsymbol{x})\right]/\partial T$, i.e., by looking at the temperature derivative of the mean total energy. We can do the same given that the heat capacity remains unchanged under the mapping $-\ln\left[P(\cdot|T=1)\right] - \ln(Z) \mapsto -\ln\left[P(\cdot|T=1)\right]$. We compute the derivative within the heat capacity numerically using a Savitzky–Golay filter [Savitzky and Golay, 1964].

Figures 8.3(c),(d) and 8.4(c),(d) show that the locations of peaks (i.e., dips) in these quantities are in qualitative agreement with the critical points highlighted by the linear dissimilarity. Note that in an LLM, text outputs are not truly sampled from a Boltzmann distribution governed by the total energy. Instead, each individual token is drawn from a Boltzmann distribution for its individual energy conditioned on the previous tokens only. This procedure corresponds to a greedy sampling strategy. The resulting sampling mismatch can lead to the counterintuitive phenomenon of the mean energy of the system increasing with decreasing temperature corresponding to a negative "heat capacity", cf. Figures 8.4(c) and (d). With increasing N_{tokens}, the signal of the heat capacity at low temperature seems to vanish in size and shift toward $T = 0$. In contrast, the high-temperature peak grows (exhibiting finite-size scaling effects) and shifts toward larger temperatures.

Discussion

Intuitively, the two critical points we have found mark transitions between three distinct regimes of LLM behavior: "frozen" at low temperatures $T < T_{\mathrm{c},1}$, "unfrozen and sensible" at intermediate temperatures $T_{\mathrm{c},1} < T < T_{\mathrm{c},2}$, and "random" at high temperatures $T_{\mathrm{c},2} < T$. In Figures 8.3 and 8.4 we have investigated the output distributions corresponding to two specific prompts. While we find that the temperature behavior does depend on the prompt, there seems to be a trend: many distinct prompts lead to a transition at $T \approx 1$ (i.e., on the order of the natural temperature scale) and at $T \ll 1$, particularly when N_{tokens} is sufficiently large. Moreover, the transitions are also visible when considering larger Pythia models, see Figure H.2 in Appendix H.

We expect that the high-temperature transition point $T_{\mathrm{c},2}$ attains a non-zero value in the limit $N_{\text{tokens}} \to \infty$.[8] In contrast, the low-temperature transition point is expected to tend to zero $T_{\mathrm{c},1} \to 0$ as $N_{\text{tokens}} \to \infty$. These scaling behaviors are to be confirmed in future investigations involving a larger number of output tokens.

The low-temperature transition we discovered here is reminiscent of the transition we have observed in the Ising model at low-temperature using SL and PBM as well as the topological crossover in the IGT, see Chapter 3. In both cases, the system transitions from occupying a single state (its ground state) to multiple states, which is visible as a large change in the underlying probability distribution. The high-temperature transition is reminiscent of the order-disorder transition in the Ising model.

The transition at low temperatures has recently been investigated in [Bahamondes, 2023] for GPT-2 [Radford *et al.*, 2019] using physics-inspired quantities. Moreover, they speculated on the existence of a phase transition at higher temperatures. This high-temperature transition in GPT-2 was recently found and analyzed by Nakaishi *et al.* [2024] using correlations in part-of-speech tags. Their analysis revealed signs of critical behavior that complement the scaling behavior we observe in Figure 8.3.

8.3.3 Transitions as a function of the training epoch

Finally, we search for transitions as a function of the training epoch ($\gamma =$ epoch), i.e., we compare the output distributions of models at different stages during training and see whether there are certain epochs at which these statistics change drastically. Such temporal analyses are rare given that they require access to models at checkpoints during training [Liu *et al.*, 2021; Gurnee and Tegmark, 2023; Chen *et al.*, 2023]. Here, we analyze the Pythia suite of models for which such checkpoints are publicly available.

Millidge [2023] analyzed the weight distribution of the Pythia models, and similar weight-based analyses of other NNs during training have also been performed in previous works [Shwartz-Ziv and Tishby, 2017; Achille *et al.*, 2019; Chen *et al.*, 2023]. To study the previously observed transitions [Millidge, 2023], we analyze changes in the weight distributions in the same manner as for the output distributions (see Section 8.2), i.e., to characterize phase transitions using dissimilarities the lists of model weights are converted to distributions via histogram binning (10000 bins for the range -3 to 3). In this case, sums over all elements of the state space can be carried out explicitly (instead of being estimated via sampling).

[8] In general, LLMs have a finite context length, i.e., a finite number of tokens over which interaction is mediated and correlations can be taken into account. In the case of the Pythia suite, for example, the context length is 2048. Hence, the limit $N_{\text{tokens}} \to \infty$ is strictly speaking not properly defined and one would need to consider models with increasing context length as the number of tokens is increased.

The results for the Pythia 70M model with $l = 6$ are shown in Figure 8.5(a) as colored lines, each corresponding to the distribution of the weights of a particular query-key-value (QKV) layer. Different layers show transitions at roughly 20000 (layer 5), 40000 (layers 3), 50000 (layer 4), and 80000 (layer 4) epochs, matching the observations of Millidge [2023] who analyzed these transitions using the norm of the weights as an "order parameter". We also observe a large peak around epoch 0, i.e., at the start of the training, highlighting that the LLM learns most rapidly at the beginning stages. In the long run, the dissimilarity curves approach 0, signaling that the weight distributions become less and less distinguishable overall.

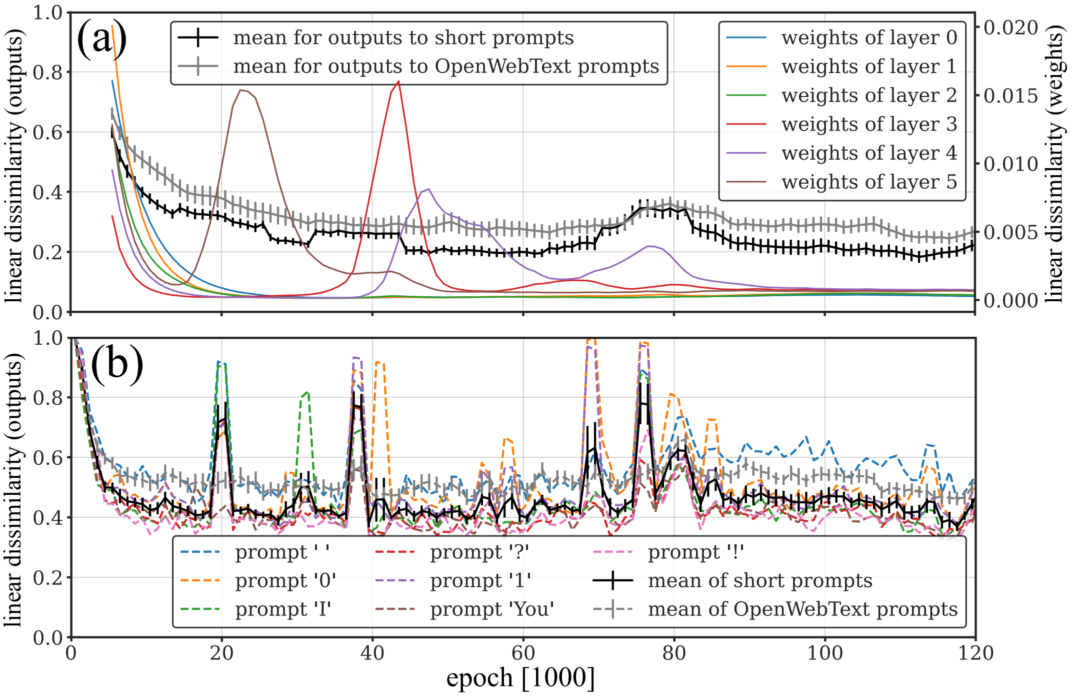

FIGURE 8.5: Linear dissimilarity by epoch for the Pythia 70M model with checkpoints taken every 1000 epochs. (a) Computed at $l = 6$ for both weights and responses to 20 random prompts from OpenWebText (gray) and 7 short prompts (black) shown in panel (b). (b) Computed at $l = 1$ for several prompts. For reference, the mean linear dissimilarity over short prompts and OpenWebText prompts with $l = 1$ is also shown. [Number of text outputs generated per parameter value γ and prompt: $|\mathcal{D}_\gamma| = 1536$. Number of generated output tokens: $N_{\text{tokens}} = 10$. Error bars indicate the standard error of the mean over all corresponding prompts. Error bars for the individual prompts in panel (b) are almost negligible and thus omitted to avoid visual clutter.]

Complementarily, in the same plot, we show dissimilarities derived from the LLM output distributions. The grey line corresponds to an average of the dissimilarities obtained by using entries from OpenWebText [Aaron Gokaslan and Vanya Cohen, 2019] as prompts. The black line corresponds to the average of results obtained from a selection of short, generic, single-token prompts (" ", "0", "I", "?", "1", "You", and "!"). Both dissimilarity curves show a peak around epoch 0 as well as a small bump around 80000 epochs that is potentially related to the rapid change of layer 4 around the same time. Intuitively, we would expect that the rapid changes in the weight distributions also affect the output distributions. However, our analysis shows that the transitions in the weights of intermediate layers at intermediate epochs do not seem to affect the output distribution drastically.

Figure 8.5(b) shows the linear dissimilarity of the output distribution as a function of the training epoch for $l = 1$. The output distributions associated to the short prompts seem more sensitive as compared to the long examples from OpenWebText: their mean dissimilarity shows clear peaks near epochs 20000, 40000, and 80000. Further analysis shows that these correspond to outliers where the output distribution changes drastically only at a single point during training and returns back (close) to its original value immediately after. As such, these peaks do not mark transitions between two macroscopic phases of behavior. We explicitly verified this by inspecting the dissimilarity between the points directly to the left and right of the outlier epoch. Our analysis provides correlational evidence linking these outliers to the transitions observed in the layer weights shown in panel (a). Note that the larger l value used in panel (a) averages out the signal stemming from these outliers. Such a reduced susceptibility to outliers can be an advantage of using $l > 1$, in particular when searching for macroscopic transitions.

Some peak locations in the dissimilarity curves are prompt-dependent, indicating that learning progresses differently for different types of behavior. Here we have used rather generic prompts, resulting in an analysis of the LLM's general behavior during training. However, in principle, conditioning on the prompt allows one to analyze whether and when specific knowledge emerges [Liu *et al.*, 2021; Gurnee and Tegmark, 2023]. As an outlook, one can imagine automatically monitoring changes across a multitude of prompts on different topics and testing different abilities at scale, without the need to design individual metrics for each prompt.

8.4 Related works

Before concluding, let us discuss how our method relates to other approaches for studying transitions in LLMs.

Generic performance-based analysis

Many previous works found transitions in LLM behavior by locating sharp changes in generic performance measures, such as sudden drops in training loss [Olsson *et al.*, 2022; Chen *et al.*, 2023]. While this may capture transitions in the overall behavior, such an approach cannot resolve transitions in specific LLM behavior. In particular, it may miss algorithmic transitions where the same performance is reached but by different means [Zhong *et al.*, 2023].

Prompt-specific success metrics

Other works have found transitions by looking at success metrics tailored toward specific prompts [Brown *et al.*, 2020; Hendrycks *et al.*, 2020; Austin *et al.*, 2021; Liu *et al.*, 2021; Radford *et al.*, 2021; Srivastava *et al.*, 2022; Wei *et al.*, 2022]. Recalling the example studied in Section 8.3.1, this would correspond to assigning a score of 1 if the LLM provided the correct answer to the question "Is γ larger than 42?" and 0 otherwise. Similarly, one could compute such a score in a temporal analysis (Section 8.3.3) or for detecting transitions as a function of another hyperparameter (Section 8.3.2).

A downside of this approach is that it is restricted to prompts that allow for a clear score to be assigned. In particular, choosing an appropriate scoring function may require lots of human engineering. In some examples, such as the tokenizer transition studied in Figure 8.2(b) with the prompt "γ", it may be unclear how to craft an appropriate metric. Moreover, discontinuous metrics can artificially induce transitions even where the underlying behavior varies smoothly [Schaeffer *et al.*, 2023]. Similarly, success metrics may miss transitions where the same performance is reached but by different means [Zhong *et al.*, 2023].

Measures based on model internals

The aforementioned approaches are based on the model output. Many works have also detected transitions based on changes in the internal structure of models, such as its trainable parameters [Millidge, 2023; Chen *et al.*, 2023]. This is similar to the weight-based analysis we have performed in Section 8.3.3. However, access to model internals may not always be available. Moreover, the design of measures that capture specific transitions in the internal behavior requires lots of human input [Räuker *et al.*, 2023; Conmy *et al.*, 2023; Zhong *et al.*, 2023], e.g., using insights from the field of *mechanistic interpretability*.

8.5 Summary

In this chapter, we have proposed a method for automating the detection of phase transitions in LLMs and demonstrated that it successfully reveals a variety of transitions. Leveraging access to the LLMs' next-token probability distributions, the proposed dissimilarity measures can efficiently quantify distribution shifts without fine-tuning or adaption to the specific scenario at hand – recall that we have utilized the same g-dissimilarity across all examples. Because the method is solely based on analyzing a model's output distribution and access to the model weights is not required, it enables *black-box interpretability* studies.

Specific findings

Let us summarize the key findings we have obtained in this chapter using our method:

- The instruction-tuned Llama and Mistral models seem to have the capability to order integers whereas all considered base models do not.

- Changes in integer tokenization can be visible in the text output as sharp transitions.

- Three distinct phases of behavior as a function of an LLM's temperature can be mapped out: a deterministic "frozen" phase near zero temperature, an intermediate "coherent" phase, and a "disordered" phase at high temperatures.

- An LLM's "heat capacity" with respect to the temperature can be negative, i.e., the LLM's mean energy can decrease as its temperature is increased.

- Rapid changes in the distribution of weights during training may signal instabilities during training causing short-term changes in the text output. Changes in the weights of later layers may cause more long-lasting effects.

- Different prompts result in different transition times during training, suggesting that distinct types of behavior can be learned rapidly at distinct times in training.

8.6 Outlook

Future large-scale investigations are needed to fully understand how the uncovered transitions depend on variables such as the specific prompt or the selected model. In particular, due to computational resource constraints, the size of the studied language models has been limited. Similarly, it will be of interest to increase the number of generated output tokens and utilize models with larger context lengths to extend the finite-size scaling analysis performed in Section 8.3.2.

Our current understanding of most of the phase transitions studied in this chapter is limited. Let us discuss the temperature-induced transition at $T_{\mathrm{c},2}$, for example. When fitting data with Boltzmann machines and introducing a fictitious temperature parameter, the corresponding heat capacity has been observed to possess a peak near $T = 1$ for various datasets [Mora and Bialek, 2011; Stephens *et al.*, 2013; Nguyen *et al.*, 2017]. Similarly, it is known that the generative model's Fisher information peaks at this location. The divergence of these two quantities may, however, not explicitly signal a phase transition in the underlying dataset. Rather, a high susceptibility indicates that distinct parameter values of the generative model can be effectively distinguished given the data at hand. In contrast, low susceptibilities suggest that the corresponding likelihood differs only marginally. Consequently, it is expected that the parameters of a reconstructed model lie in a region of the parameter space with high susceptibility – a region in which models with distinct parameters can be clearly distinguished based on the data. While LLMs are not Boltzmann machines, they are generative models trained to fit text data via maximum likelihood at $T = 1$. Moreover, connections between self-attention and the Potts model have recently started to be explored [Rende *et al.*, 2024]. Given these similarities, it remains an exciting open question whether the high-temperature transition in LLMs is of similar origin and whether $T_{\mathrm{c},2} \to 1$ as $N_{\mathrm{tokens}} \to \infty$.

The method we propose in this chapter to detect phase transitions is not only applicable to language models but can be straightforwardly adapted to any generative model with an explicit, tractable density. This is particularly exciting given the development of powerful generative models for distinct modalities, such as images, videos, and speech.

If one can draw samples from the output distribution but does not have explicit access to the underlying probabilities, then the dissimilarity measures can still be approximated using NN-based classifiers tailored toward the particular data type, such as natural language.[9] We will utilize this approach to detect transitions in real-world news data in Chapter 9.

The results and figures presented in this chapter have been in parts published in [Arnold *et al.*, 2024a]. The corresponding code is open source at [Arnold *et al.*, 2024b].

[9] Recall that LBC and other data-driven methods for detecting phase transitions have originally been proposed using discriminative NNs.

Chapter 9

Phase Transitions in Real-World News Data

The results presented in this chapter are based on the following manuscript:

Machine learning change points in real-world news data,
C. Zsolnai, N. Lörch, and J. Arnold,
manuscript in preparation (2025).

9.1 Motivation

Identifying points in time where temporal data changes abruptly is key for understanding the underlying dynamics and formulating effective responses. As such, the task of *change point detection* finds applications in a vast range of domains including economics [Pepelyshev and Polunchenko, 2015], medicine [La Rosa *et al.*, 2008; Maladi *et al.*, 2013], environmental science [Reeves *et al.*, 2007; Itoh and Kurths, 2010; Gong *et al.*, 2023; Beaulieu *et al.*, 2024], speech recognition [Rybach *et al.*, 2009; Gupta, 2015], and linguistics [Kulkarni *et al.*, 2015]. In the context of news data, change point detection can expose shifts in topics, sentiments, trends, and patterns of interest. For example, it can identify influential events such as breaking news, political developments, or natural disasters. Understanding when these events occur and their impact on the public discourse provides valuable insights into society at large. Changes in news data may also reflect changes related to the news venue itself, such as their publication rate, changes in staff, or changes in reader engagement. Detecting change points in massive real-world textual data is a formidable task complicated by the fact that the data is high-dimensional [Grundy *et al.*, 2020] and correlated. Moreover, the data is often sparse with only a few documents per time point being available.

The problem of offline change-point detection[1] can be solved by computing a dissimilarity score between probability distributions governing the data of two consecutive time segments. If the dissimilarity score is sufficiently large, the time point indicating the split between the two segments is deemed a change point. The key difficulty lies in the fact that the underlying probability distributions are typically unknown because we only have access to samples, and performing density estimation in a high-dimensional space is a hard problem.

In this chapter, we tackle the problem of estimating dissimilarity scores using the variational representations discussed in Chapter 6 that we exploit using NNs

[1] Offline change point detection refers to identifying changes or abrupt shifts in the statistical properties of a time series *after* the entire dataset has been collected and is available for analysis. In contrast, *online* change point detection identifies changes in the statistical properties of a time series as data arrives, in real-time or near-real-time. In this chapter, we solely focus on the former problem.

tailored toward natural language processing. That is, we utilize the discriminative LBC (including the multitasking approach proposed in Chapter 7) to estimate the TV distance between the distributions underlying neighboring time segments from data. Here, change points take the role of critical points and time t takes the role of a tuning parameter γ. We showcase the efficacy of this approach for detecting change points datasets of both synthetically generated and real-world news articles.

9.2 Methodology

In the following, we formally describe the task of detecting change points and our approach to solving it. We are given a set of samples $\mathcal{D}_t = \{\boldsymbol{x}|\boldsymbol{x} \in \mathcal{X}, \boldsymbol{x} \sim P(\cdot|t)\}$ representative of the distribution $P(\cdot|t)$ at a discrete set of points in time $t \in \Gamma \subset \mathbb{R}$. In this chapter, $\boldsymbol{x}$ corresponds to a news article. Because we will eventually work with real-world data, $|\mathcal{D}_t|$ may be different for different points in time. Let us denote the set of points t^{bp} lying halfway in-between sampled points as $\Gamma' = \{t_1^{\mathrm{bp}}, t_2^{\mathrm{bp}}, \dots, t_{K-1}^{\mathrm{bp}}\}$, where $K = |\Gamma|$. Each bipartition point $t^{\mathrm{bp}} \in \Gamma'$ is a candidate for a change point dividing the time axis into two segments, $\Gamma_0(t^{\mathrm{bp}})$ and $\Gamma_1(t^{\mathrm{bp}})$, each comprised of the l time points $t \in \Gamma$ closest to t^{bp} with $t < t_{\mathrm{bp}}$ and $t > t^{\mathrm{bp}}$, respectively.[2] The two segments $\Gamma_0(t^{\mathrm{bp}})$ and $\Gamma_1(t^{\mathrm{bp}})$ are each characterized by the datasets $\mathcal{D}_0(t^{\mathrm{bp}}) = \{\boldsymbol{x}|\boldsymbol{x} \in \mathcal{D}_t, t \in \Gamma_0(t^{\mathrm{bp}})\}$ and $\mathcal{D}_1(t^{\mathrm{bp}}) = \{\boldsymbol{x}|\boldsymbol{x} \in \mathcal{D}_t, t \in \Gamma_1(t^{\mathrm{bp}})\}$, respectively. These datasets are viewed as being representative of the probability distributions underlying the two segments, $P(\cdot|0;t^{\mathrm{bp}})$ and $P(\cdot|1;t^{\mathrm{bp}})$, respectively.[3] To assess whether t^{bp} is a change point or not, we compute a dissimilarity score between the distributions underlying the two segments $D(t^{\mathrm{bp}}) = D\left[P(\cdot|0;t^{\mathrm{bp}})|P(\cdot|1;t^{\mathrm{bp}})\right] \geq 0$. The higher the dissimilarity score, the more likely the point t^{bp} is a change point. Thus, we can identify change points as significant maxima in $D(t^{\mathrm{bp}})$ with $t^{\mathrm{bp}} \in \Gamma'$.[4] In principle, to quantify the dissimilarity any *statistical distance* may be used (recall Chapter 5). Here, we consider the TV distance

$$D_{\mathrm{TV}}[p,q] = \frac{1}{2}\sum_{\boldsymbol{x}\in\mathcal{X}} |p(\boldsymbol{x}) - q(\boldsymbol{x})| \tag{9.1}$$

which belongs to the broader class of statistical distances known as *f-divergences*. The hyperparameter l determines the timescale on which changes are detected. If, for example, Γ contains daily points in time and $l = 7$, we detect variations on a weekly basis.

Estimating statistical distances

To estimate the TV distance, we note that $D_{\mathrm{TV}}[p,q] = 1 - 2p_{\mathrm{err}}^{\mathrm{opt}}$ where $p_{\mathrm{err}}^{\mathrm{opt}}$ is the Bayes-optimal average error probability when trying to decide whether a given sample $\boldsymbol{x}$ has been drawn from p or q, i.e., when performing the task of single-shot binary symmetric hypothesis testing (cf. Chapter 5). Consequently, the average error rate extracted from any classifier lower bounds the TV distance as $1 - 2p_{\mathrm{err}} \leq 1 - 2p_{\mathrm{err}}^{\mathrm{opt}} = D_{\mathrm{TV}}[p,q]$. Here, the two relevant distributions are $p \mapsto P(\cdot|0;t^{\mathrm{bp}})$ and $q \mapsto P(\cdot|1;t^{\mathrm{bp}})$.

[2] Note that for bipartition points t^{bp} close to the border of the sampled time interval Γ less than l time points may be available in one of the two segments.

[3] In previous chapters, we had $P(\cdot|0;t^{\mathrm{bp}}) = \frac{1}{|\Gamma_0(t^{\mathrm{bp}})|}\sum_{t\in\Gamma_0(t^{\mathrm{bp}})} P(\cdot|t)$ and $P(\cdot|1;t^{\mathrm{bp}}) = \frac{1}{|\Gamma_1(t^{\mathrm{bp}})|}\sum_{t\in\Gamma_1(t^{\mathrm{bp}})} P(\cdot|t)$. This factorized form treats each day equally and assumes that distinct days are uncorrelated. This may not be the case for real-world news data and hence, we do not assume such a form explicitly.

[4] Recall the discussion regarding the vagueness in detecting “phase transitions” within a finite-sized system at the start of Section 8.2.

For each bipartition point t_k^{bp}, $k \in \{1, 2, \ldots, K-1\}$, one can train a parametric binary classifier $\hat{y}_{\boldsymbol{\theta}^{(k)}} : \mathcal{X} \to [0, 1]$ to minimize an unbiased binary cross-entropy loss

$$\mathcal{L}_{\mathrm{train}}(\boldsymbol{\theta}^{(k)} | t_k^{\mathrm{bp}}) = -\frac{1}{2} \sum_{y \in \{0,1\}} \frac{1}{|\mathcal{T}_y(t_k^{\mathrm{bp}})|} \sum_{\boldsymbol{x} \in \mathcal{T}_y^{(k)}} \left(y \ln \left[\hat{y}^{(k)}_{\boldsymbol{\theta}^{(k)}}(\boldsymbol{x}) \right] + (1-y) \ln \left[1 - \hat{y}^{(k)}_{\boldsymbol{\theta}^{(k)}}(\boldsymbol{x}) \right] \right), \tag{9.2}$$

making it distinguish between samples drawn from the two distributions. Here, $\mathcal{T}_y(t_k^{\mathrm{bp}}) \subset \mathcal{D}_y(t_k^{\mathrm{bp}})$ is the corresponding training set.[5] Its error rate can be estimated as

$$\tilde{p}_{\mathrm{err}}^{(k)} = \frac{1}{2} \sum_{y \in \{1,0\}} \frac{1}{|\mathcal{D}_y(t_k^{\mathrm{bp}})|} \sum_{\boldsymbol{x} \in \mathcal{D}_y(t_k^{\mathrm{bp}})} \mathrm{err}^{(k)} \left[\hat{y}^{(k)}_{\boldsymbol{\theta}^{(k)}}(\boldsymbol{x}) \right], \tag{9.3}$$

where the error function $\mathrm{err}^{(k)}$ is 0 if the sample $\boldsymbol{x}$ is classified correctly and 1 otherwise.[6] Based on this estimate, we can approximate the TV distance as $\tilde{D}_{\mathrm{TV}}(t_k^{\mathrm{bp}}) = 1 - 2\tilde{p}_{\mathrm{err}}^{(k)}$. In the infinite-data limit, the optimal classifier under the loss in Equation (9.2) attains the Bayes-optimal error rate (cf. Chapter 4), thus $\tilde{D}_{\mathrm{TV}}(t_k^{\mathrm{bp}}) \to D_{\mathrm{TV}}(t_k^{\mathrm{bp}})$ asymptotically.

In case we choose $l = \infty$, i.e., we split the entire time range into two distinct segments, we use the multi-tasking approach to LBC introduced in Chapter 7:[7] Instead of training a distinct binary classifier for each tentative change point $t^{\mathrm{bp}} \in \Gamma'$ separately, we train a classifier with trainable parameters $\boldsymbol{\theta}$ that has $K-1$ outputs corresponding to all nontrivial tentative change points with a loss function proportional to $\sum_{t^{\mathrm{bp}} \in \Gamma'} \mathcal{L}_{\mathrm{train}}(\boldsymbol{\theta} | t^{\mathrm{bp}})$.

9.2.1 Choice of classifier

As a parametric classifier, we use a simple feedforward NN with a single hidden layer comprised of 64 nodes. We consider two distinct numerical representations for news articles $\boldsymbol{x}$.

Term frequency-inverse document frequency

The first one is based on term frequency-inverse document frequency (TF-IDF) [Salton and Buckley, 1988]. TF-IDF is a measure of the importance of a term (word) to a document in a corpus, taking into account the fact that some words appear more frequently in general, e.g., words such as "and" or "or". The TF-IDF score is a function of a term x, news article $\boldsymbol{x}$, and collection of news articles $\mathcal{D} = \{\boldsymbol{x} \in \mathcal{D}_t | t \in \Gamma\}$:

$$\mathrm{tfidf}(x, \boldsymbol{x}, \mathcal{D}) = \mathrm{tf}(x, \boldsymbol{x}) \cdot \mathrm{idf}(x, \mathcal{D}), \tag{9.4}$$

where

$$\mathrm{tf}(x, \boldsymbol{x}) = \frac{\sum_{x' \in \boldsymbol{x}} \delta_{x,x'}}{\sum_{x' \in \boldsymbol{x}} 1} \tag{9.5}$$

[5] In case of the validation loss, $\mathcal{T}_y(t_k^{\mathrm{bp}})$ gets replaced by the corresponding validation set $\mathcal{V}_y(t_k^{\mathrm{bp}})$.

[6] Here, we always choose the entire dataset for evaluation, i.e., $\mathcal{D}_y(t_k^{\mathrm{bp}}) = \mathcal{E}_y(t_k^{\mathrm{bp}})$.

[7] In principle, one may also use multi-tasking for finite l. In practice, however, while the corresponding indicator curves often highlight the most important change point correctly, the training seems to suffer from local minima. As such, we observe it to be more difficult to reach Bayes' error using an NN with multi-tasking when l is finite. Trying to alleviate these issues, e.g., through appropriate modifications to the neural network architecture, is left for future work. Generally, the individual classification tasks are expected to become more and more independent as l decreases. Consequently, the advantage of multitasking is also expected to diminish.

measures the relative frequency of term x appearing within the document $\boldsymbol{x}$ and

$$\operatorname{idf}(x, \mathcal{D}) = \ln\left[\frac{|\mathcal{D}|}{|\{\boldsymbol{x}|\boldsymbol{x} \in \mathcal{D} \wedge x \in \boldsymbol{x}\}|}\right] \tag{9.6}$$

is a measure of how much information the term carries, i.e., a measure of how infrequently it appears across all documents. Having computed the TF-IDF scores for all terms within all documents of the dataset, each document is represented as a vector containing the TF-IDF scores of all its terms $\boldsymbol{x} \mapsto \{\operatorname{tfidf}(x, \boldsymbol{x}, \mathcal{D})\}_{x \in \boldsymbol{x}}$.

The key downside of this simple representation is that it ignores in-document correlations between terms. The TF-IDF representation is purely frequency-based and does not take the context in which a term appears into account. Moreover, the resulting representation can be quite high-dimensional.

Transformer-based embedding model

The second representation *does* take in-document correlations between terms into account. It is obtained using the all-MiniLM-L6-v2 transformer-based embedding language model (LM) that maps text to a 364-dimensional vector space.[8] The embedding model is trained to map similar text inputs to similar latent representations.

Training details

For training, we use the Adam optimizer [Kingma and Ba, 2014] with a learning rate of 10^{-4}, a batch size of 64, and default settings otherwise. We split the entire data $\mathcal{D}$ into a training $\mathcal{T}$ and validation set $\mathcal{V}$ by grouping all news articles according to their publication date. Next, for each unique publication date, a random 80% of news articles are added to the training set while the remaining 20% is assigned to the validation set. Each training runs for at least 200 epochs. Afterward, we check the validation loss every 10 epochs and stop the training if it stops improving or when 20000 training epochs are reached.

9.2.2 Detecting change points via topic extraction

To independently confirm whether the change points that we detect are appropriate, we consider an alternative method for detecting change points as a benchmark. This method is based on the intuition that change points in news may often be related to topical changes, and there exist ML methods that extract topics from text data in an unsupervised fashion. In this chapter, we utilize latent Dirichlet allocation (LDA) as a topic extraction method [Blei *et al.*, 2003].[9]

Here, each "topic" corresponds to a probability distribution $P_{\text{LDA}}(x|\text{topic})$ over all unique terms appearing in a collection of news articles, $x \in \{x \in \boldsymbol{x}|\boldsymbol{x} \in \mathcal{D}\}$. LDA assigns each document $\boldsymbol{x}$ a distribution over topics $P_{\text{LDA}}(\text{topic}|\boldsymbol{x})$. Averaging over all the articles present at days underlying segment 0/1 in parameter space, we obtain a distribution over topics conditioned on a segment $y \in \{0, 1\}$:[10]

$$P_{\text{LDA}}(\text{topic}|y) = \frac{1}{|\mathcal{D}_y|} \sum_{\boldsymbol{x} \in \mathcal{D}_y} P_{\text{LDA}}(\text{topic}|\boldsymbol{x}). \tag{9.7}$$

[8] This model truncates any input text longer than 256 tokens.

[9] In LDA, the number of topics is a hyperparameter that has to be set beforehand.

[10] Later on, we will also visualize the distribution over topics conditioned on a given time point $P_{\text{LDA}}(\text{topic}|t) = \frac{1}{|\mathcal{D}_t|} \sum_{\boldsymbol{x} \in \mathcal{D}_t} P_{\text{LDA}}(\text{topic}|\boldsymbol{x})$.

Given that there are only a handful of topics, we can explicitly calculate the TV distance between the distributions over topics underlying the two segments

$$D_{\mathrm{TV}}[P_{\mathrm{LDA}}(\cdot|0), P_{\mathrm{LDA}}(\cdot|1)] = \frac{1}{2} \sum_{\text{topics}} \left| P_{\mathrm{LDA}}(\text{topic}|0) - P_{\mathrm{LDA}}(\text{topic}|1) \right|. \tag{9.8}$$

We may view the TV distance between distributions over topics as an approximation of the TV distance between distributions over articles. In fact, due to the data-processing inequality (see Section 5.2) we have

$$D_{\mathrm{TV}}[P_{\mathrm{LDA}}(\cdot|0), P_{\mathrm{LDA}}(\cdot|1)] \leq D_{\mathrm{TV}}[P(\cdot|0), P(\cdot|1)], \tag{9.9}$$

where equality holds if and only if the topic is a sufficient statistic. Equation (9.9) captures the intuitive fact that the more informative the topics are for classifying the articles into the two segments, the larger the corresponding TV distance between distributions over topics, and the better the approximation to the true TV distance.

By repeating this process for all possible segments, we can obtain a dissimilarity score as a function of the bipartition parameter. Once again, change points can be detected as local maxima.

9.2.3 The Guardian news articles dataset

Year	Category	Year	Category
2000	UK News	2014	World
2000	US News	2015	UK News
2000	World	2015	US News
2001	UK News	2016	UK News
2001	US News	2016	US News
2001	World	2016	World
2002	World	2019	UK News
2007	UK News	2019	US News
2007	US News	2019	World
2007	World	2020	UK News
2008	World	2020	US News
2010	UK News	2020	World
2010	US News	2021	UK News
2010	World	2021	US News
2011	US News	2021	World
2012	UK News	2022	US News
2012	US News	2022	World
2012	World		

TABLE 9.1: All sets of news articles (197170 in total) from The Guardian considered in this chapter. Each dataset contains all publicly accessible news articles in a given year and category after our filtering stage. In the filtering stage, we discard all non-English articles, articles that are mainly composed of non-textual data (such as video or audio), and articles that are less than 1000 words in length (we concatenate the article titles and the main text).

Our ultimate goal is to detect change points in real-world news data. In this chapter, we mainly focus on *The Guardian* – a British daily newspaper that has a free, publicly available application programming interface (API) to query newspapers on specific dates. On The Guardian, each article is assigned a category, such as "US News", "UK News", "Technology", "Business", and so on. Here, we focus on the three categories of "UK News", "US News", and "World" given the large range of topics that are covered in these categories and the large number of daily articles. In total, we have collected 197170 publicly accessible news articles with publishing dates spanning from the year 2000 up to the year 2022. We organize this data into distinct sets corresponding to all the publicly accessible articles within a given year and category. These datasets specified by their year and category are summarized in Table 9.1.

Artificially generated datasets

To test our method and demonstrate its ability to detect change points, we would like to create a scenario in which we know the underlying change point that needs to be detected *a priori*.

To this end, we generate a first benchmark dataset (*benchmark 1*) composed of artificial "news articles" on a given topic via an LLM (ChatGPT-3.5). We explicitly induce a change point by changing from one topic to another (e.g., from "George Bush" to "New Year celebrations"). To make the generated text more realistic, we provide several randomly selected news articles from the Guardian newspaper as examples within the prompt. We consider a time interval of one month (30 days) and generate 10 artificial news articles per day. The corresponding topics and change points are listed in Table 9.2. Note that these are quite few articles per day. However, in contrast to the real dataset, the same number of articles is present on each day.

Topic 1	**Topic 2**	**Change point**
Cannes film festival	Italian supreme court	2001-03-20
George Bush	New Year celebrations	2007-03-10
Christmas	Pakistan	2010-03-10
European Union	North Korea	2012-03-20
Olympic games	Eiffel tower	2014-03-05
Oktoberfest	Brexit	2016-03-05
Komodo dragons	Canada	2019-03-31
Solar panels	Iran	2020-03-15
Syrian economy	Reindeers	2021-03-05
COVID-19 pandemic	Russian war against Ukraine	2022-02-15

TABLE 9.2: Major topics and their corresponding change points (in Year-Month-Day format) within the *benchmark 1* dataset generated via ChatGPT.

Artifically induced split points

As a second benchmark (*benchmark 2*), we create a dataset that is composed of real news articles from the Guardian newspaper. However, here we explicitly induce a change point by switching between news categories (e.g., from "UK News" to "US News"). As a time interval, we consider a whole year. The corresponding categories and change points are listed in Table 9.3.

Category 1	Category 2	Change point
US News	UK News	2000-04-14
US News	World	2010-04-15
UK News	US News	2015-05-07
UK News	World	2019-01-18
US News	UK News	2019-03-27
UK News	World	2019-10-30
US News	World	2022-02-16
US News	World	2022-03-17

TABLE 9.3: News categories and their corresponding change points (in Year-Month-Day format) within the *benchmark 2* dataset.

9.3 Experiments

In this section, we are going to present and discuss the results of our methods for change point detection being applied to the three datasets introduced in Section 9.2.3 with increasing complexity.

9.3.1 Results for benchmark 1

Figure 9.1 shows a set of results for a dataset from the first benchmark where a transition between artificial news articles on "George Bush" and "New Year celebrations" is induced. Here, all three methods yield an approximation of the TV distance that shows a peak at the artificial change point.

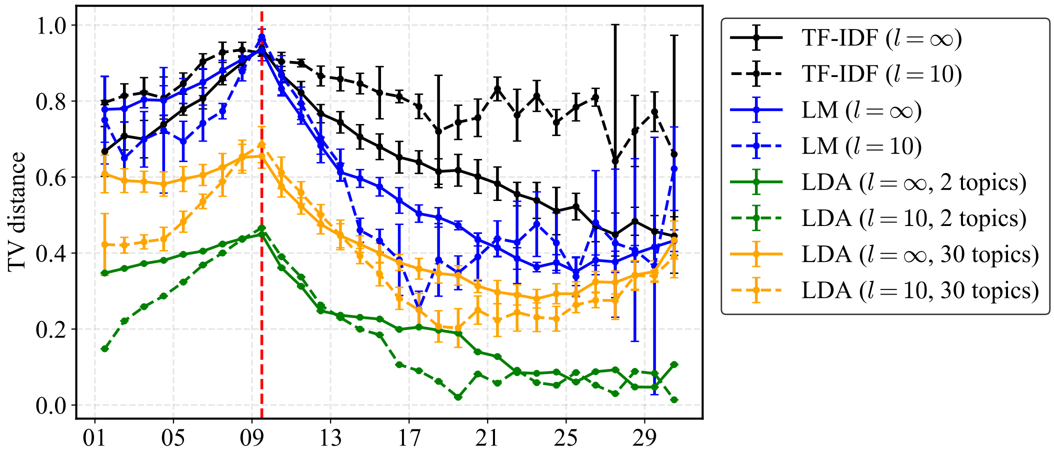

FIGURE 9.1: Result of change point detection methods applied to a dataset from benchmark 1, where artificial news articles were generated on the topic "George Bush" (left) and "New Year celebrations" (right). The error bars correspond to the standard deviation over 5 independent training runs. The artificially induced change point is highlighted by a vertical red line.

We can observe that the NN-based classifiers with a TF-IDF representation yield the largest estimate for the TV distance, followed by NN classifiers with LM representation, and finally LDA. Within LDA, the largest estimates are obtained when more topics are included. These observations can be made across all analyzed datasets and are related to the loss of information which increases as we move from a TF-IDF

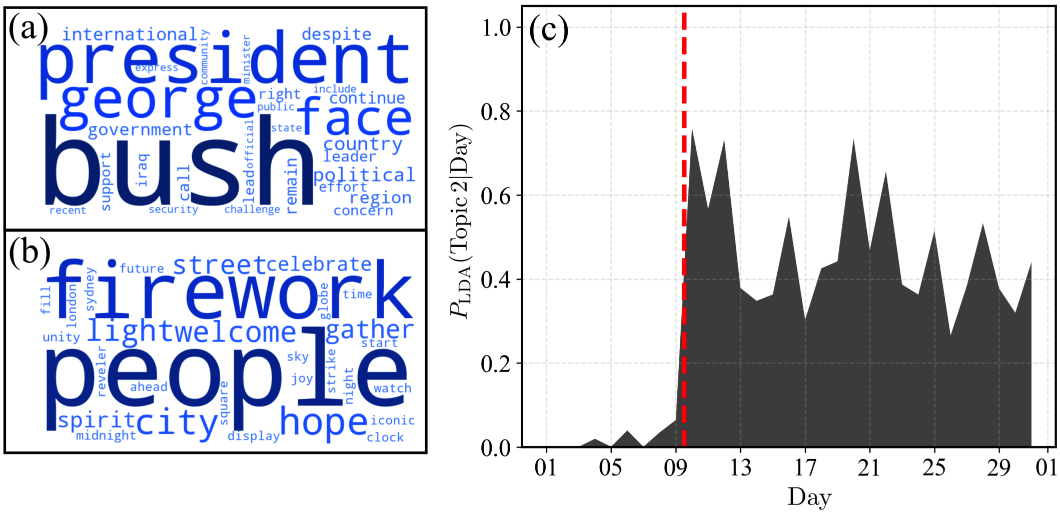

Figure 9.2: Result of LDA (2 topics) applied to a dataset from benchmark 1, where artificial news articles were generated on the topic "George Bush" (left; topic 1) and "New Year celebrations" (right; topic 2). Word cloud visualization depicting the terms in (a) topic 1 and (b) topic 2, where the size of a word corresponds to its relative weight within the topic $P_{\mathrm{LDA}}(x|\text{topic})$. (c) Probability of an article being categorized as topic 2 for each day in the 30-day time interval. The artificially induced change point is highlighted by a vertical red line.

representation to an LM representation, and finally to the few topics extracted via LDA. Note that the dimensionality of the representation also decreases in this order. That is, the TF-IDF representation seems to preserve the most information, i.e., results in the news articles being most distinguishable. In practice, this also results in a reduced computation time for training LM-based classifiers compared to ones based on the TF-IDF representation.

Figure 9.2 shows the topics that LDA detects for this dataset as well as the corresponding topic distribution over time. LDA does indeed correctly identify the two key themes ("George Bush" and "New Year celebrations") within the articles.

Table 9.4 shows the mean deviations of the peak location from the artificial change point across all 10 datasets considered in the first benchmark (see Table 9.2). We find that all methods qualitatively highlight the change point. Here, the TV distance estimated via an NN-based classifier in combination with the LM embedding yields the best results, performing slightly better than the TF-IDF representation and LDA. Note that although the dataset features articles with two dominant themes, LDA with 30 topics is also capable of detecting the underlying change points (see Figure 9.1). This showcases the viability of choosing a large number of topics *a priori* if knowledge about the underlying dataset is limited.

Regarding the choice of l, we find that for each method, the best results can be achieved with $l = 10$ (being $1/3$ of the total time interval), corresponding to a detection of changes on a timescale of 10 days. If l is too small, the indicator curves are prone to be highly noisy due to the small amount of news articles present in each region.

9.3.2 Results for benchmark 2

Figure 9.3 shows results for a dataset from the second benchmark where a transition between news articles from the "UK News" category to news articles from the "US

Method	Mean deviation [days]
TF-IDF ($l = 1$)	7.9 ± 1.2
LM ($l = 1$)	6.0 ± 1.1
LDA ($l = 1$, 2 topics)	2.2 ± 0.7
LDA ($l = 1$, 30 topics)	3.0 ± 0.8
TF-IDF ($l = 2$)	5.3 ± 1.0
LM ($l = 2$)	5.2 ± 1.0
LDA ($l = 2$, 2 topics)	2.2 ± 0.8
LDA ($l = 2$, 30 topics)	2.5 ± 0.9
TF-IDF ($l = 5$)	1.2 ± 0.3
LM ($l = 5$)	1.7 ± 0.5
LDA ($l = 5$, 2 topics)	2.0 ± 0.7
LDA ($l = 5$, 30 topics)	2.1 ± 0.8
TF-IDF ($l = 10$)	1.6 ± 0.5
LM ($l = 10$)	$\mathbf{0.6 \pm 0.2}$
LDA ($l = 10$, 2 topics)	1.5 ± 0.6
LDA ($l = 10$, 30 topics)	2.1 ± 0.7
TF-IDF ($l = \infty$)	3.8 ± 0.9
LM ($l = \infty$)	1.0 ± 0.3
LDA ($l = \infty$, 2 topics)	1.8 ± 0.7
LDA ($l = \infty$, 30 topics)	4.0 ± 0.9

TABLE 9.4: Mean absolute deviation of location of maximum in estimated TV distance from the location of the artificial split point over all datasets within *benchmark* 1, see Table 9.2. We report the standard error over all datasets, where we perform 5 independent training runs for each dataset. Recall that these datasets each span a month in time (30 days).

News" category is induced in 2019. Again, all three methods yield an approximation of the TV distance that shows a peak at the artificial change point.

Table 9.5 shows the mean deviations of the peak location and the artificial change point across all 8 datasets considered in the second benchmark (see Table 9.3). Here, we find that our method based on NN classifiers with LM representations performs on par with LDA, highlighting the split point to within a few days given data on a whole year's time interval. As in Section 9.3.1, we find that an intermediate l yields the best performance, here $l = 100$. As a consequence, we will proceed with utilizing this setting for the remainder of this chapter. The large mean deviation of LDA with 2 topics as well as TF-IDF is largely caused by a single dataset (transition from "US News" to "World" category in the year 2010) where these methods yield an indicator with a second peak that corresponds to the global maximum, highlighting another significant event within the "World" news category.[11] Note that this peak is also present for the other two methods, albeit as a local maximum. In Table 9.3, we report the results obtained when excluding this year from the dataset in brackets.

9.3.3 Results for real-world news dataset

Having verified that our methods are indeed able to detect artificially induced change points, we now apply them to the full Guardian news article dataset. It is an open question what kind of change points are hidden in this dataset. Intuitively, we would

[11] An analysis of the news articles around that time (end of November 2010), including LDA topics, suggests that this peak is related to the WikiLeaks diplomatic cables release, also known as "Cablegate".

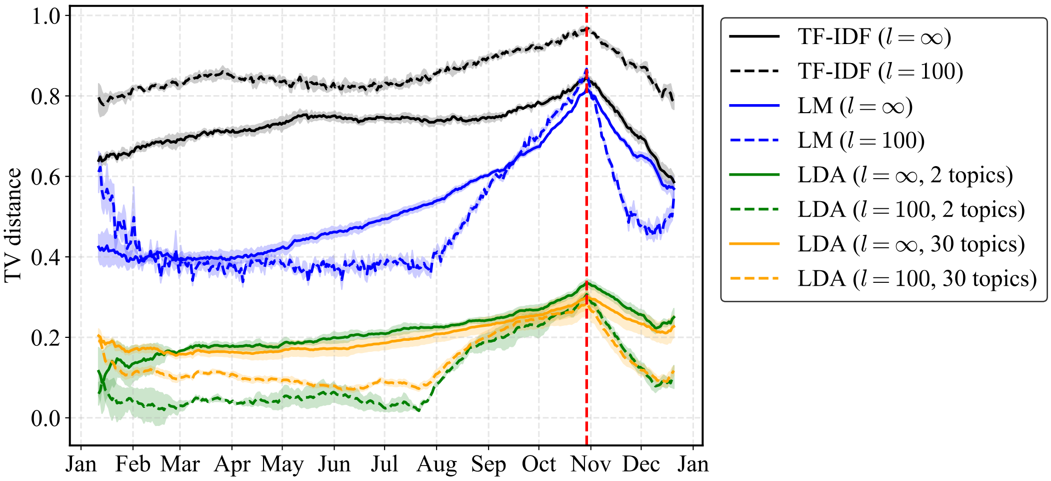

FIGURE 9.3: Result of change point detection methods applied to a dataset from benchmark 2 (transition from "UK News" to "US News" category in the year 2019). The error bands correspond to the standard deviation over 5 independent training runs. The first and last 10 days of the year have been excluded from the analysis as the corresponding indicators suffer from large edge effects caused by insufficient data. The artificially induced change point is highlighted by a vertical red line.

Method	**Mean deviation [days]**
TF-IDF ($l = 10$)	31 ± 11
LM ($l = 10$)	9 ± 3
LDA ($l = 10$, 2 topics)	33 ± 12
LDA ($l = 10$, 30 topics)	4 ± 1
TF-IDF ($l = 100$)	37 ± 13 (16 ± 8)
LM ($l = 100$)	$\mathbf{3 \pm 2}$
LDA ($l = 100$, 2 topics)	35 ± 12 (8 ± 3)
LDA ($l = 100$, 30 topics)	$\mathbf{3 \pm 1}$
TF-IDF ($l = \infty$)	11 ± 4
LM ($l = \infty$)	14 ± 4
LDA ($l = \infty$, 2 topics)	44 ± 12
LDA ($l = \infty$, 30 topics)	23 ± 4

TABLE 9.5: Mean deviation of location of maximum in estimated TV distance compared to the location of the true artificial split point over all datasets within *benchmark* 2, see Table 9.3. We report the standard error over all datasets, where we perform 5 independent training runs for each dataset. Recall that these datasets each span a whole year in time. The mean deviation in brackets is obtained by excluding the second dataset from 2010. Note that the first and last 10 days of the year have been excluded from the analysis as the corresponding indicators suffer from large edge effects caused by insufficient data.

speculate that significant events, such as the terrorist attack on the 11th of September 2001 or the outbreak of COVID-19 certainly show up as change points. And indeed, these events can be detected as peaks in the approximate TV distance.

Figure 9.4 shows an example result of the methods applied to the dataset of news articles within the 2001 "World" category. Here, all four estimates of the TV distance show a clear peak around the 11th of September – the day on which Islamist terrorists attacked the United States.

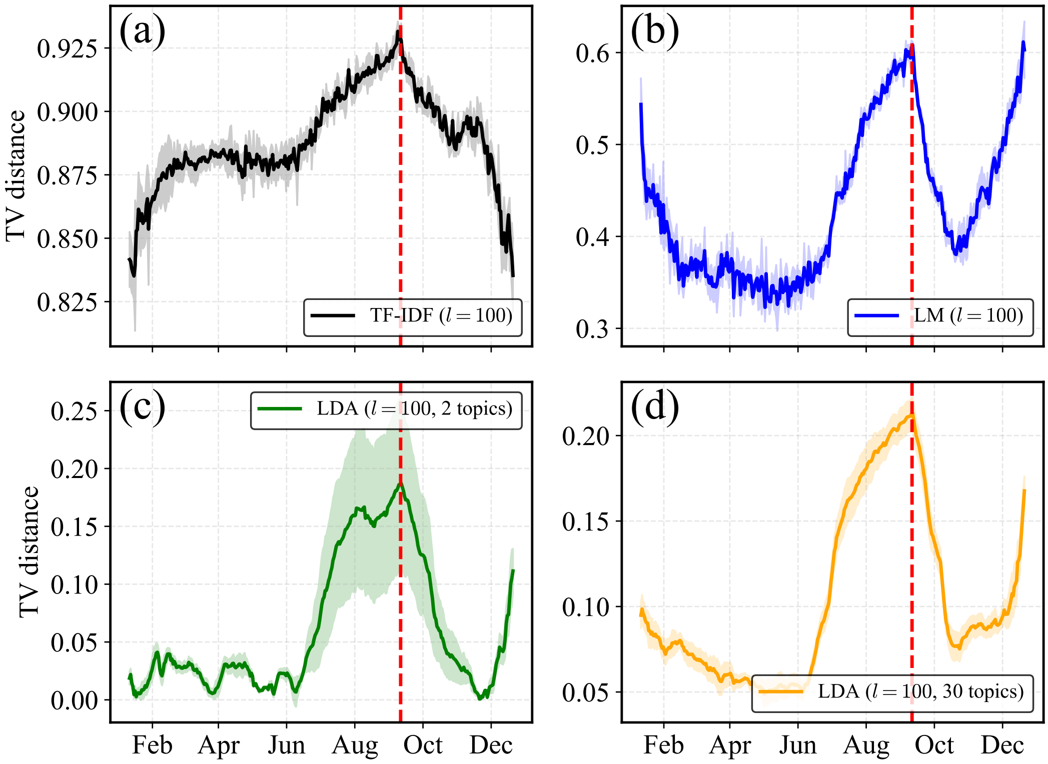

FIGURE 9.4: Result of change point detection methods applied to news articles from The Guardian newspaper in 2001 within the category "World". The error bands correspond to the standard deviation over 5 independent training runs. The first and last 10 days of the year have been excluded from the analysis as the corresponding indicators suffer from large edge effects caused by insufficient data. The 11th of September is highlighted by a vertical dashed line.

Note that the TV distance approximations around the start and end of the year are expected to be most sensitive to finite-sample statistics and should be trusted least. Hence, we omit the signal for time points at the very edges of the interval.

Another example of a significant historical event being highlighted is the COVID-19 pandemic, see Figure 9.5 which shows the TV distance approximations for the news article within the 2020 "World" category. Because the spread of the coronavirus is a gradual event, it is also expected that its news coverage will ramp up more gradually (compared to the news coverage of a terrorist attack). Here, we have chosen the day on which the World Health Organization officially declared the coronavirus outbreak a pandemic as a reference point. However, the coronavirus was already discussed in the news before this date, which may explain the large indicator values within that time range, see Figures 9.5(b)-(d).

Further analysis of the news articles within this category reveals that the second peak in the indicators around the end of May/start of June (see dotted vertical line in Figure 9.5) may be related to the conflict between mainland China and Hong Kong on which the Guardian newspaper frequently reported.

Other events that can be identified as peaks in the approximate TV distance of one of the three methods include the presidential elections in 2012 and 2016, the

Brexit referendum in 2016, the withdrawal of US troops from Afghanistan in 2021 culminating in the fall of Kabul, or the invasion of Russia into Ukraine in 2022.

Note that significant events can influence the public discourse and news cycle in drastically different ways. Thus, these events are also expected to result in distinct indicator signals, i.e., are expected to be easier or harder to detect with certain methods, highlighting the challenge of analyzing real-world news data. Compare, for example, a presidential election to a terrorist attack. The date of the former is typically well-known in advance, whereas the latter usually comes as a surprise. As such, the frequency with which the news reports on the presidential election is expected to ramp up leading up to the election date, and ramp down afterward. Hence, the change is more subtle and largely semantically: before the election date the winner is not known and multiple candidates are in the race, whereas after the election date, the winners and losers are known. In contrast, terrorist attacks only spark discourse after they happen as they come as a surprise.

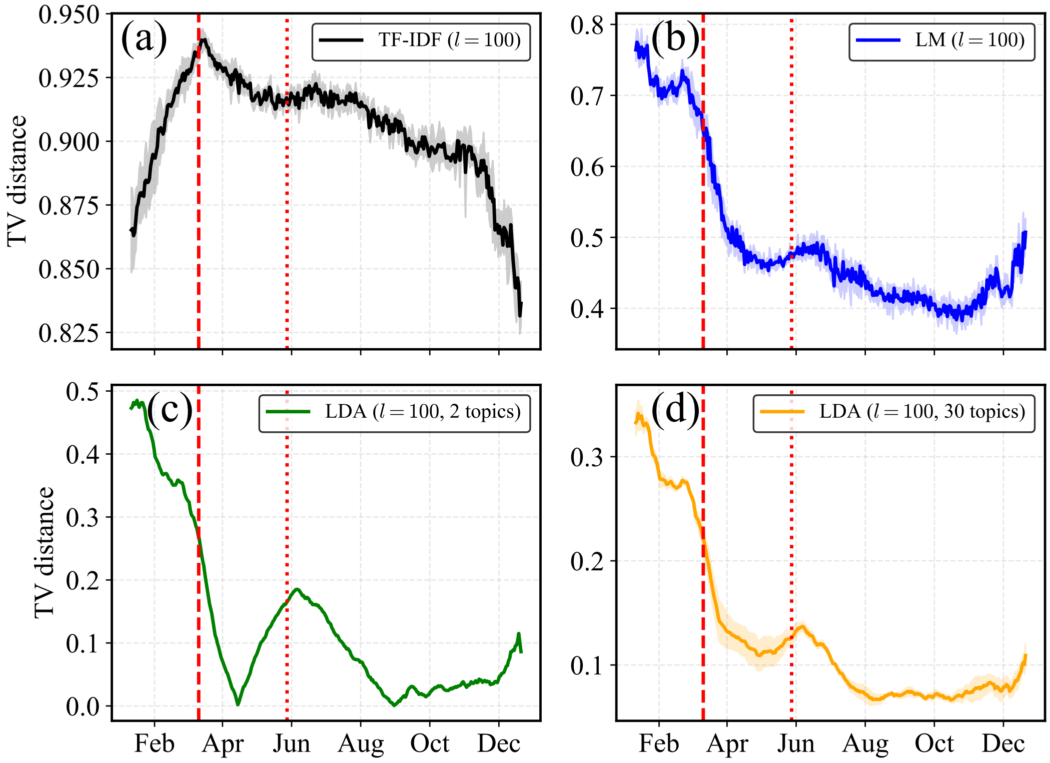

Figure 9.5: Result of change point detection methods applied to news articles from The Guardian newspaper in 2020 within the category "World". The error bands correspond to the standard deviation over 5 independent training runs. The first and last 10 days of the year have been excluded from the analysis as the corresponding indicators suffer from large edge effects caused by insufficient data. The vertical dashed line highlights the 11th of March 2020 on which the World Health Organization officially declared the coronavirus outbreak a pandemic. The dotted dashed line highlights the 28th of May on which China's National People's Congress put in place new controversial security laws for Hong Kong.

Do the change-point-detection methods highlight significant events?

From the previous set of results, it may seem that the methods regularly detect significant events. In this section, we try to quantify what constitutes a "significant event". We can then compare the dates at which such significant events occur with the location of the maxima in the approximate TV distance to evaluate the performance of our change-point-detection methods at identifying these events. This provides further insight into what signals these methods are sensitive to and what changes are indeed reflected in The Guardian news dataset.

To this end, we rely on Wikipedia's "Year in Review" pages where contributors list events deemed most significant for each year. In the following, we use these events as a "ground truth". This choice is motivated by the fact that these lists are the most comprehensive, publicly available representations of significant events spanning the years 2000 to 2022. While an event may be deemed significant by a contributor, it may not be represented within The Guardian dataset. Hence, from Wikipedia's list of significant events, we select the one whose description has the most number of common tokens with the titles of The Guardian news articles within a given year and category. In 2001, for example, this does indeed highlight 9/11 as the most significant event.

We find that for all three methods, the mean deviation of the location of the maximum of the approximate TV distance and the significant event across the various Guardian news article datasets is on the order of 100 days, which is high compared to the total time interval of a year. From this, we conclude that the methods do not necessarily always detect significant events – as we have defined them above – as change points.

Finding explanations for the observed TV distances remains a hard task. In particular, as noted in Section 9.1, the factors that may influence the news discourse are highly diverse and may even be related to the internal workings of The Guardian newspaper itself. As such, large changes in the TV distance may not necessarily be related to significant events. In contrast to the model systems typically studied in physics, i.e., the models we analyzed in the first part of this thesis, prior theoretical insights that may serve as ground truth are lacking.

9.4 Related works

For a thorough review of the field of change-point-detection methods, see [Aminikhanghahi and Cook, 2017; Van den Burg and Williams, 2020; Truong *et al.*, 2020]. Interestingly, various works have approached the problem of detecting change points via computing a dissimilarity based on statistical divergences, such as the KL divergence, JS divergence, or the Pearson χ^2 divergence, in the past. However, many of these rely on estimating the underlying densities via histogram binning [Afgani *et al.*, 2008; Basterrech and Woźniak, 2022], which is bound to fail in the multivariate case, i.e., for inputs living in high-dimensional spaces due to the curse of dimensionality. Another line of work uses a parametric approach for estimating the densities by making strong distributional assumptions, such as Gaussianity [Siegler *et al.*, 1997; Jabari *et al.*, 2019], which may not be appropriate.

An important insight that led to an improved number of methods for change-point detection using dissimilarities based on statistical divergences is the fact that these can be estimated given only the ratio between the two probability distributions.[12]

[12] Recall that the central object in an f-divergence $D_f[p,q]$ is the ratio p/q entering the function f.

Since one can calculate this density ratio knowing the densities underlying the two intervals but not the other way around, estimating the density ratio is expected to be an easier problem[13]. In [Sugiyama *et al.*, 2008; Kawahara and Sugiyama, 2009; Liu *et al.*, 2013; Haque *et al.*, 2017], nonparametric Gaussian kernel models of the density ratio have been proposed for this purpose.

In recent years there has been a surge in the use of NNs for change-point detection. Gupta [2015], for example, utilized NNs trained to distinguish between French and English radio. Other approaches relied on an autoencoder-based dissimilarity measure [De Ryck *et al.*, 2021], self-supervised contrastive learning [Deldari *et al.*, 2021], or newly proposed NN architectures [Ebrahimzadeh *et al.*, 2019]. LLMs have only started to be explored recently for this task [Tevissen *et al.*, 2023]. NNs have also been used as density ratio models [Khan *et al.*, 2019; Moustakides and Basioti, 2019; Chen *et al.*, 2021], replacing the kernel models in earlier works. Nevertheless, it remains challenging to fully leverage the power of deep learning for this change-point detection.

To that end, it has been noted that the density ratio is also the central object in binary classification [Qin, 1998; Cheng and Chu, 2004; Bickel *et al.*, 2009; Sugiyama *et al.*, 2012; Menon and Ong, 2016] and an estimate for the density ratio can be extracted from any classifier.[14] This discriminative approach to density ratio estimation has found success in various applications, including reinforcement learning [Liu *et al.*, 2018], energy-based models [Gutmann and Hyvärinen, 2012], generative adversarial networks [Nowozin *et al.*, 2016], as well as change-point detection [Hido *et al.*, 2008; Wang *et al.*, 2023].

In [He *et al.*, 2022; Zhao *et al.*, 2024], a modified version of LBC has been used for detecting change points in various types of data, including a real-world Twitter dataset. The work of Zhao *et al.* [2024] is perhaps the closest to the approach we presented in this chapter. Our work differs from their approach in several ways. First, Zhao *et al.* [2024] did not properly correct for the bias resulting from the different dataset sizes within the two segments, which may cause erroneous change-point signals (recall our findings from Chapter 4). Second, they train a separate classifier for each tentative change point, making the procedure more computationally intensive compared to our multi-tasking approach.[15] Third, they did not point out the fundamental connection between the TV distance and the indicator of LBC. Fourth, in this chapter, we analyzed the Guardian news dataset which is more multi-faceted and spans a longer time range compared to the Twitter datasets analyzed in [Zhao *et al.*, 2024] that focused on COVID-19 and the 2017 French Election.

9.5 Summary

In this chapter, we demonstrated a method capable of detecting change points in real-world news data that is based on approximating the TV distance between the distributions of underlying time segments variationally via NN-based classifiers. This

[13] This is in the spirit of Vapnik's principle [Vapnik, 1998] stating that when solving a problem of interest, one should not solve a more general problem as an intermediate step.

[14] A Bayes-optimal binary classifier predicts $P(0|\boldsymbol{x}) = P(\boldsymbol{x}|0)/\left[P(\boldsymbol{x}|0) + P(\boldsymbol{x}|1)\right] = 1/\left[1 + P(\boldsymbol{x}|1)/P(\boldsymbol{x}|0)\right]$, and similarly for $P(1|\boldsymbol{x})$. Thus, the ratio of the Bayes-optimal predictions corresponds to the density ratio $P(0|\boldsymbol{x})/P(1|\boldsymbol{x}) = P(\boldsymbol{x}|0)/P(\boldsymbol{x}|1) = r(\boldsymbol{x})$. The ratio of the estimated class probabilities from any classifier may thus serve as an estimate of the density ratio $\hat{r}(\boldsymbol{x}) = \tilde{P}(0|\boldsymbol{x})/\tilde{P}(1|\boldsymbol{x})$. This estimate can be used for further downstream tasks. For example, an estimate of any f-divergence D_f can be constructed as $D_f\left[P(\cdot|0), P(\cdot|1)\right] \approx 1/|\mathcal{D}_1| \sum_{\boldsymbol{x}\in\mathcal{D}_1} f\left[\hat{r}(\boldsymbol{x})\right]$.

[15] Similar to our multitasking approach to LBC, Wang *et al.* [2023] capitalized on the similarity between the distributions of neighboring segments via so-called "variational continual learning".

allows us to capitalize on the rapid development of NN architectures tailored toward natural language processing, including transformer-based language models.

9.6 Outlook

In the future, it will be interesting to capitalize on the method we proposed in this chapter to gain novel quantitative insights into the public discourse captured by news articles. One might, for example, compare different US presidential elections based on their approximate TV distance. Which presidential election in recent history was most disruptive to the news cycle? Similarly, one may compare the average value of the approximate TV distance for different years. Which year between 2000 and 2022 was the most turbulent viewed by this metric? We believe that our method is in a unique position to aid in answering such questions by providing access to quantitative measures.

In Section 9.3.3, we touched upon the fact that distinct events, such as terrorist attacks and presidential elections, influence the news in drastically different ways. In the future, it will be of interest to develop a classification of change points in news that captures such differences.

It is known that estimates of the density ratio extracted from classifiers can be inaccurate when the two distributions are very different [Rhodes *et al.*, 2020; Choi *et al.*, 2021]. In such cases, the discriminative task (given a finite set of samples) becomes trivial and the classifier can achieve perfect accuracy, leading to imperfectly calibrated output probabilities, and thus density ratio estimates. In future work, one may think of constructing more accurate estimators of the relevant statistical distances.

Our method explicitly ignores correlations between inputs collected at different points in time, i.e., it assumes that the articles within a given time segment are independently and identically distributed. It is known that the presence of correlations can lead to false detection signals if not properly taken into account [Shi *et al.*, 2022]. Similarly, it leads to difficulties in identifying abrupt changes in the frequency domain [De Ryck *et al.*, 2021]. Intuitively, such changes are not as important for textual data. This is confirmed *a posteriori* given that our method successfully captures meaningful transitions in real-world news data. In future studies venturing beyond this realm of applications, this is a limitation that may need to be addressed.

Here, we have not introduced any postprocessing procedure for detecting peaks in the dissimilarity measure, i.e., for identifying discrete change points. Instead, we relied on visual inspection of the dissimilarity score, which is common practice [De Ryck *et al.*, 2021]. One way this can be overcome is by setting a threshold value c such that t^{bp} is considered a change point if $D(t^{\mathrm{bp}}) \geq c$. This threshold value needs to be carefully calibrated as it depends on the application scenario as well as the chosen dissimilarity function [Liu *et al.*, 2013].

A key strength of our method is the fact that it can utilize NN architectures tailored for solving supervised learning tasks given data of various forms. This makes it potentially applicable for a variety of different fields where change point detection is required, such as audio, video, and image segmentation.

Finally, in many applications where timely responses are crucial, online change point detection is desirable. It is interesting to think about how one could extend the method we presented here to work in an online scenario.

Part III

Conclusion and Appendices

Chapter 10

Conclusion and Outlook

In recent years, we have witnessed remarkable advances across multiple scientific frontiers ranging from the development of quantum computers to the deployment of increasingly sophisticated artificial intelligence systems. One thing that these domains have in common is the vast size of the underlying state space. Despite this complexity, we would like to understand and characterize distinct behavioral regimes within these systems. Up to now, scientists have largely relied on prior system knowledge and their human intuition to tackle this challenge. It was *their* task to compress the underlying state space by identifying a few key low-dimensional quantities that accurately capture the macroscopic behaviors of the system.

In this thesis, we have explored the possibility of utilizing ML methods to automate this process. Such approaches bear the potential to enable new phases of behavior and phase transitions to be discovered autonomously from readily available data without much prior system knowledge or human supervision. We have laid a particular focus on the three popular methods – supervised learning (SL), learning by confusion (LBC), and the prediction-based method (PBM) – that typically utilize neural networks as predictive models at their core. We demonstrated that by carefully analyzing and improving these methods for detecting phase transitions, we can not only better understand their working principle in physical systems, but also extend their utility to entirely new domains.

In the following, let us summarize the key contributions we have presented over the course of the two main parts of this thesis and discuss core research directions for future work. We refer the reader to the summary and outlook section of each individual chapter for a more in-depth discussion.

Part I: Learning to Detect Phase Transitions in Physical Systems

Summary

In the first part of this thesis, we developed a deeper understanding of NN-based methods for detecting phase transitions from a probabilistic perspective. This started in Chapter 3 where we derived analytical expressions for the optimal predictions and indicators of SL, LBC, and PBM, revealing that these methods inherently rely on detecting changes in the probability distributions governing the system's state. Remarkably, these analytical expressions enabled a novel computational procedure for detecting phase transitions directly from data without training neural networks, resulting in significant speedups. Moreover, they allowed us to explain the successes and failures documented in a variety of previous NN-based studies.

In Chapter 4, we formulated the three methods in a fully probabilistic manner. This allowed us to generalize the ML methods to higher-dimensional parameter spaces featuring multiple phases. Similarly, it enabled us to generalize the computational procedure from Chapter 3 – the so-called generative approach – to work with any generative model with an explicit, tractable density. This, in turn, enabled larger system sizes to be studied by using variational methods, such as tensor networks, to obtain approximations of the probability distributions governing the system's measurement statistics. Similarly, it makes the ML methods applicable to study LLMs as we explored in Chapter 8. The probabilistic formulation also exposed previously hidden assumptions within the ML methods, allowing us to further improve their ability to detect phase transitions through appropriate modifications. Recall that in Chapter 3, we found that the ML methods may not correctly highlight the phase transition when employed with an optimal predictive model. The modifications in Chapter 4 largely remedied this issue.

In fact, in Chapter 6, we were able to prove that the indicators of all three (modified) methods approximate the square root of the system's Fisher information from below. This establishes a fundamental connection between the ML and information-theoretic paradigms for studying critical phenomena. This result puts the understanding from Chapter 3 that the methods inherently rely on detecting changes in the probability distributions governing the system's state on a firm footing. Given that the Fisher information is known to highlight first- and second-order phase transitions, this result explains the previous successes of these methods at detecting such phase transitions. Similarly, it explains the fundamental shortcomings of these methods in detecting higher-order (> 2) phase transitions. The connection also highlights the key advantage of the ML methods: The Fisher information is a difficult quantity to compute as it typically requires knowledge of the system's underlying probability distribution. Using discriminative models, the ML methods of SL, LBC, and PBM allow one to obtain an underapproximation of the system's Fisher information from samples of the distributions alone. The resulting approximation remains operationally useful in detecting phase transitions.

In Chapter 6, we have also shown that the optimal indicator of LBC is related to the TV distance. Similarly, the optimal value of the loss function can be related to the JS divergence. Using LBC with a predictive model trained from a finite amount of data yields an underapproximation of the TV distance and JS divergence, and in turn of the Fisher information. A trained classifier may, in fact, be used to approximate any f-divergence from data. While the use of the TV distance and JS divergence have been motivated historically through their role in LBC, other f-divergences may have more suitable properties.

Finally, in Chapter 7, we proposed a multi-task implementation of LBC that provides significant speedups compared to its original formulation by training a single classifier instead of multiple separate ones.

In summary, we have significantly improved the computational cost and fundamental capability of ML methods to detect phase transitions from data. As such, this thesis constitutes an important step toward enabling automated scientific discoveries in self-driven laboratories of the future. This is particularly exciting given the advent of programmable quantum simulators [Bohrdt *et al.*, 2021; Ebadi *et al.*, 2021; Semeghini *et al.*, 2021; Scholl *et al.*, 2021; Altman *et al.*, 2021; Miles *et al.*, 2023] and digital quantum computers [Smith *et al.*, 2019; Barratt *et al.*, 2021; Satzinger *et al.*, 2021; Herrmann *et al.*, 2022; Noel *et al.*, 2022; Kim *et al.*, 2023; Bluvstein *et al.*, 2024; Acharya *et al.*, 2024].

Outlook

Computational aspects.—We still lack a general understanding of the computational resources that are required to accurately determine phase transitions using ML methods: How many samples are needed to obtain accurate estimates of the critical point? Could adaptive sampling strategies help to reduce this amount? Which NN-based approach – be it discriminative or generative – is most appropriate for approximating a given indicator of phase transitions?

More generally, which approach is most appropriate for approximating the Fisher information from data remains an open question: How powerful are approaches based on approximating f-divergences? What f-divergence should be utilized? Are there other known methods for estimating the Fisher information that may be better suitable to detect phase transitions from data? Or are the approximations to the Fisher information obtained by the ML methods developed in the context of physics useful more generally?

Detecting higher-order phase transitions and topological phases of matter.—The connection of the indicators of SL, LBC, and PBM to the Fisher information is conceptually appealing. However, it also seems to limit their capability of detecting higher-order transitions. Can we generalize these methods to be able to detect higher-order transitions? What would be the core information-theoretic quantity to be estimated?

In this thesis, we have focused on phase transitions between two topologically trivial phases or between a topologically non-trivial and a topologically trivial phase. Such transitions may be detected simply through the detection of the topologically trivial phase and the departure from the latter. In the future, it will be interesting to investigate transitions between two topologically non-trivial phases of matter where such a strategy does not work anymore. In this case, the ML methods may have a hard time when employing generic IC-POVMs, given that lots of samples are required to capture nonlocal features using such measurements. Finding ways to deal with these cases remains a challenging task for future work.

One way to tackle this problem may be to extend the methods to be able to compute functions of multiple measurement outcomes (rather than just one), allowing it to be more effective at capturing correlations. In the quantum case, this has been used effectively to be able to approximate observables nonlinear in the density matrix [Huang *et al.*, 2022b; Kim *et al.*, 2024]. Another complementary strategy would involve the use of variational quantum circuits. While we have been limited to a fixed choice of POVM in this thesis, the introduction of variational circuits may allow one to simultaneously search over the space of measurements. This may not only reduce the number of samples required to resolve the phase transitions due to the ability to capture nonlocal order, but also yield a protocol for approximating the quantum Fisher information.

Unification of methods.—In this thesis, we have showcased how information theory can be used to gain a deep understanding of the ML methods of SL, LBC, and PBM and view them in a unifying light. Can we gain a deeper understanding of other ML methods for detecting phase transitions in a similar fashion? The methods based on PCA and autoencoders briefly discussed in Chapter 2, for example, can be viewed as solving a compression task. This connection can provide a starting point for an information-theoretic analysis. We believe that such an approach may ultimately help us to group and distinguish methods based on the fundamental quantities they approximate and tasks they try to solve, rather than the particularities of the NN architecture that is being employed.

Part II: Venturing Beyond Physics

Summary

In the second part of this thesis, we demonstrated the broader applicability of ML methods developed for detecting phase transitions in physics: Using our multi-tasking approach to LBC, in Chapter 7 we detected rapid changes in images generated via a text-to-image generative diffusion model as a function of an integer in its prompt. Similarly, in Chapter 8, we showed how f-divergences approximated in a generative fashion detect abrupt changes in the behavior of large language models as a function of various control parameters, including variables in the prompt, temperature, and the number of training epochs. Finally, in Chapter 9 we demonstrated the application of discriminative LBC to detect significant historical events in news articles from *The Guardian* newspaper over time.

Looking ahead, we envision these methods evolving into general-purpose tools for analyzing complex systems, complementing human insight with ML capabilities to reveal hidden patterns of change. Their broad applicability makes them particularly promising for tackling emerging challenges in domains where lots of data is readily available.

Outlook

Broadening the application domain.—In this thesis, we have showcased applications of the methods to text data and image data. A key strength of our methods is their flexibility due to their utilization of neural nets – be it as a discriminative or generative model. In future work, extensions to other data modalities, such as video and audio, remain to be explored.

Deeper understanding of discovered transitions.—While the ML methods we considered in this thesis can detect the location of phase transitions in parameter space, they do not *a priori* explain the origin of these transitions. Given that the rapid changes in generative models and news data have largely been newly discovered in this thesis, our understanding of these phenomena is fairly limited. It remains an exciting task for future work to develop a theory for these transitions.

In the case of LLMs, for example, the origin and nature of the transition as a function of its temperature remain to be fully analyzed. Can this transition be replicated in a simpler toy model? Is the dissimilarity truly divergent in the thermodynamic limit? In the case of news, is it possible to classify and categorize significant events based on their influence on the news cycle?

Appendix A

An Alternative Approach Toward Supervised Learning

Here, we review our approach to supervised learning (SL) introduced in Chapter 2 (Section 2.5.1) and put it into context. Carrasquilla and Melko [2017] originally proposed to identify the estimated critical value of the tuning parameter in SL as $\operatorname{argmin}_{\gamma\in\Gamma} |\hat{y}(\gamma) - 0.5|$. In all systems analyzed in Section 3.6, this yields similar results compared to our approach based on identifying the peak location of the mean prediction's derivative [Equation (2.9)]. Note that the latter approach has, e.g., already been mentioned as an alternative in [Broecker *et al.*, 2017]. Looking at Figure 3.19(a), we observe that these two procedures would yield slightly different estimated critical values for the MBL phase transition. This discrepancy is even more prominent for the Mott insulator to superfluid transition in the two-dimensional Bose-Hubbard model whose Hamiltonian is given by

$$H = -J\sum_{\langle ij\rangle}(b_i^\dagger b_j + \text{h.c.}) + \sum_i \frac{U}{2} n_i(n_i - 1) - \mu n_i, \tag{A.1}$$

where J is the nearest-neighbor hopping strength, U is the on-site interaction strength, and μ is the chemical potential. This model undergoes a quantum phase transition at zero temperature from a Mott insulating phase to a superfluid phase as the tuning parameter J/U is increased at a fixed chemical potential. This gives rise to the characteristic Mott lobes [Fisher *et al.*, 1989; Jaksch *et al.*, 1998], see Figure A.1(a).

We perform mean-field calculations based on a Gutzwiller ansatz in which the ground-state wave function is written as a product state

$$|\Psi_{\text{MF}}\rangle = \prod_i |\phi_i\rangle \tag{A.2}$$

with

$$|\phi_i\rangle = \sum_{n=0}^{n_{\max}} f_n |n_i\rangle, \tag{A.3}$$

where $|n_i\rangle$ denotes the Fock state with n bosons at site i [Krauth *et al.*, 1992]. We minimize the expectation value of the Hamiltonian with respect to the Gutzwiller coefficients $\{|f_n|^2\}_{n=0}^{n_{\max}}$ using simulated annealing [Comparin, 2017; Huembeli *et al.*, 2018] with a maximum number of bosons per site of $n_{\max} = 20$. As such, the Gutzwiller coefficients $\{|f_n|^2\}_{n=0}^{n_{\max}}$ represent the relevant probability distributions governing the data. Note that the simulated annealing algorithm can get stuck in local energy minima. To counteract this noise, we average the Gutzwiller coefficients obtained from 500 independent simulated annealing runs.

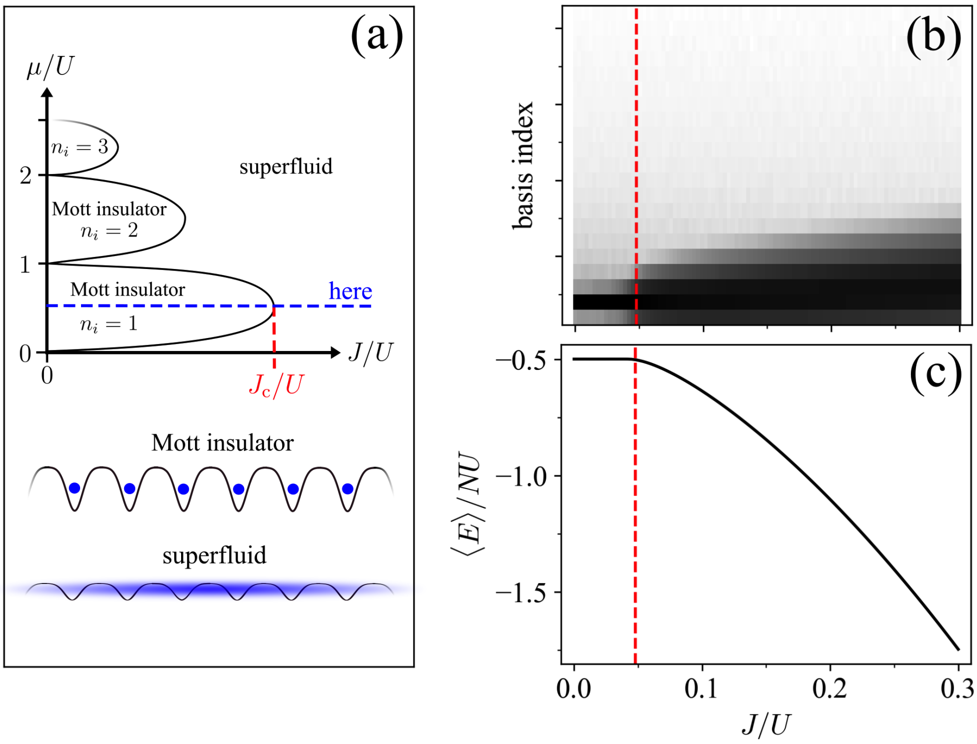

FIGURE A.1: Results for the Mott insulating to superfluid phase transition in the (two-dimensional) Bose-Hubbard model with the dimensionless coupling strength as a tuning parameter $\gamma = J/U$ ranging from $\gamma_1 = 0$ to $\gamma_K = 0.3$ in steps of $\Delta\gamma = 0.03$, where $\mu/U = 0.5$. The reference value for the critical value of the tuning parameter $J_c/U = 1/(5.8z)$ with $z = 4$ [Zwerger, 2003] is highlighted by a red dashed line. (a) Illustration of the two-dimensional phase diagram of the Bose-Hubbard model containing three Mott lobes. Here, we analyze the quantum phase transition from a Mott-insulating state to a superfluid state occurring at the tip of the first Mott lobe ($\mu/U = 0.5$). A sketch of the two distinct phases is shown at the bottom. (b) Probability distributions governing the input data (indices of Fock basis states $\{|n_i\rangle\}_{i=1}^{n_{\max}}$) as a function of tuning parameter, where the color scale denotes the probability. (c) Average energy per site (N sites in total) as a function of the tuning parameter. Notice the drop in the average energy as the system undergoes the quantum phase transition.

At the tip of the first Mott lobe ($\mu/U = 0.5$) the phase transition occurs at $J_c/U = 1/(5.8z)$ [see Figure A.1(a)], where z is the coordination number (here $z = 4$) [Zwerger, 2003]. The phase transition can be revealed by looking at the average boson number per site $\langle n\rangle$, see Figure A.2(d). The Mott insulator is characterized by an integer density enforced by the Mott energy gap $\propto U$. As a result of the energy gap, the Mott insulator is incompressible. In contrast, the superfluid phase is compressible and is characterized by strong number fluctuations (even at low temperatures).

Figure A.2 shows the results of SL, LBC, and PBM using Bayes-optimal predictive models obtained following the approach introduced in Chapter 3. Here, both SL and PBM correctly identify the quantum phase transition, whereas LBC fails. Looking at Figure A.1(b), we see that a large change in the underlying probability distributions occurs at the quantum phase transition. In [Liu and van Nieuwenburg, 2018], the

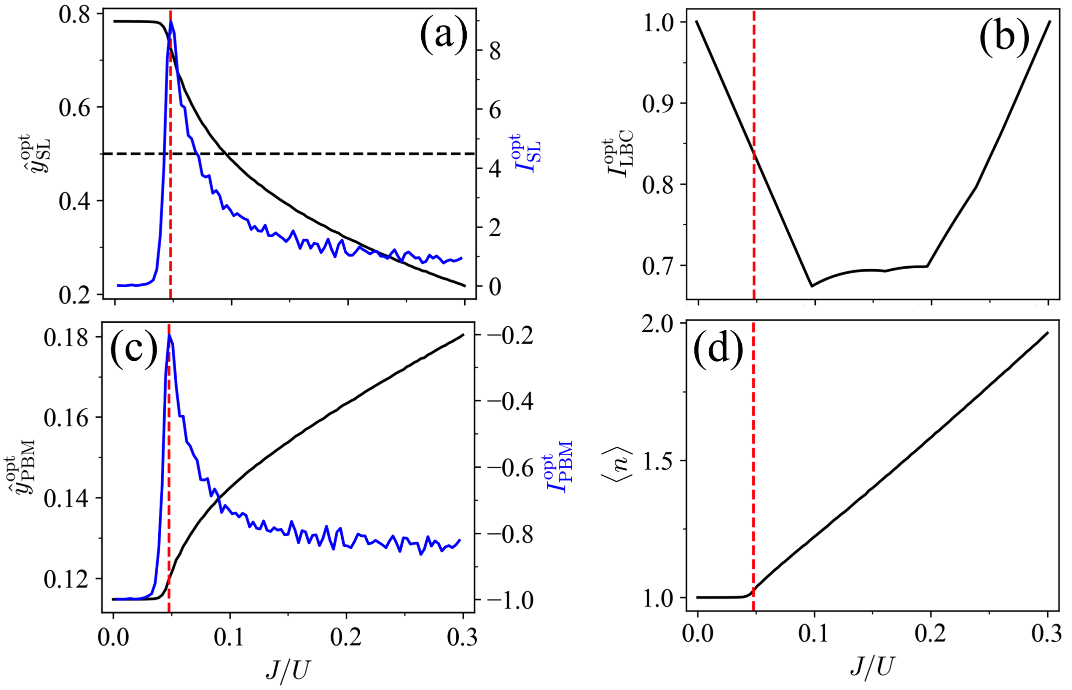

FIGURE A.2: Results for the Mott insulating to superfluid phase transition in the (two-dimensional) Bose-Hubbard model with the dimensionless coupling strength as a tuning parameter $\gamma = J/U$ ranging from $\gamma_1 = 0$ to $\gamma_K = 0.3$ in steps of $\Delta\gamma = 0.03$, where $\mu/U = 0.5$. In SL, the data obtained at γ_1 and γ_K constitutes our training set, i.e., $r_{\mathrm{I}} = 1$ and $l_{\mathrm{II}} = K$. The reference value for the critical value of the tuning parameter $J_{\mathrm{c}}/U = 1/(5.8z)$ with $z = 4$ [Zwerger, 2003] is highlighted by a red dashed line. (a) Mean optimal prediction $\hat{y}_{\mathrm{SL}}^{\mathrm{opt}}$ in SL (black, solid) and the corresponding indicator $I_{\mathrm{SL}}^{\mathrm{opt}}$ (blue). The value $\hat{y}_{\mathrm{SL}}^{\mathrm{opt}} = 0.5$ is highlighted by a black dashed line. (b) Optimal indicator of LBC, $I_{\mathrm{LBC}}^{\mathrm{opt}}$ (black). (c) Mean optimal prediction $\hat{y}_{\mathrm{PBM}}^{\mathrm{opt}}$ in PBM (black) and the corresponding indicator $I_{\mathrm{PBM}}^{\mathrm{opt}}$ (blue). (d) Average occupation number per site $\langle n \rangle$ as a function of the tuning parameter.

Mott insulating to superfluid transition in the Bose-Hubbard model was correctly highlighted using LBC with NNs. However, in this case, the Gutzwiller coefficients directly served as input, whereas here, the individual Fock basis states (i.e., their indices) constitute the input. Note that the phase transition would not be predicted with a high accuracy using SL if we estimated the predicted critical temperature as the value of the tuning parameter for which $\hat{y}_{\mathrm{SL}}^{\mathrm{opt}} = 0.5$, see black dashed line in Figure A.2(a). This motivates our approach to SL compared to the procedure originally proposed in [Carrasquilla and Melko, 2017].

Appendix B

Assumptions for Optimal Supervised Learning

Let us review the assumption of $\bar{\mathcal{E}} = \bar{\mathcal{T}}$ underlying the derivation for the optimal predictions and corresponding indicator of SL in Section 3.2.1. In general, if $\bar{\mathcal{E}} \neq \bar{\mathcal{T}}$ the optimal predictions of SL can be expressed

$$\hat{y}_{\mathrm{SL}}^{\mathrm{opt}'}(\gamma) = \sum_{\boldsymbol{x} \in \bar{\mathcal{T}} \cap \bar{\mathcal{E}}} \tilde{P}^{(\mathcal{E})}(\boldsymbol{x}|\gamma) \hat{y}_{\mathrm{SL}}^{\mathrm{opt}}(\boldsymbol{x}) + \sum_{\boldsymbol{x} \in \bar{\mathcal{E}} \setminus \bar{\mathcal{T}}} \tilde{P}^{(\mathcal{E})}(\boldsymbol{x}|\gamma) \hat{y}_{\mathrm{SL}}(\boldsymbol{x}). \tag{B.1}$$

The first contribution in Equation (B.1) comes from predictions for inputs contained in the training data, which are determined through minimization of the corresponding loss function [see Equation (3.6)]. The second contribution comes from predictions for inputs not contained in the training data, which are *a priori* only restricted to the unit interval $\hat{y}_{\mathrm{SL}}(\boldsymbol{x}) \in [0, 1]$. Therefore, this contribution to Equation (B.1) is bounded by the probability of drawing an input at γ that is not present in the training data. When using SL with NNs, the predictions for inputs not contained in the training data [second contribution in Equation (B.1)] will be most susceptible to noise inherent to NN training and hyperparameter choices. As such, its physical relevance is questionable. It may be possible to obtain better bounds for this second contribution when using SL with NNs, e.g., based on the theory of neural tangent kernels [Jacot *et al.*, 2018].

Let us explicitly discuss the classical systems analyzed in Chapter 3, which are governed by a Boltzmann distribution [Equations (3.45) and (3.46)]. Because the probability of drawing a particular configuration sample (or energy) at any non-zero temperature is non-zero, the assumption of $\bar{\mathcal{E}} = \bar{\mathcal{T}}$ holds given a sufficient number of samples. When computing the optimal indicator of SL numerically, we work with a finite number of samples. Thus, we may encounter an input during evaluation that has not been part of the training data $\boldsymbol{x} \notin \bar{\mathcal{T}}$. In practice, we can verify on-the-fly whether this is the case. If so, we set $\hat{y}_{\mathrm{SL}}(\boldsymbol{x}) = 0$ in the second part of Equation (B.1). Thereby, we effectively ignore the contribution to the predictions of SL from inputs not present in the training data. Note that because these predictions correspond to inputs with low probability, they are also most susceptible to finite-sample statistics. This procedure is further justified by the fact that the optimal predictions $\hat{y}_{\mathrm{SL}}^{\mathrm{opt}}$ obtained in this manner track the ground-state probability with high accuracy [see Figures 3.5(a), 3.8(a), and 3.12(a)]. That is, the optimal predictions closely match the expression in Equation (3.65) valid in the case where deviations due to finite-sample statistics vanish.

In the quantum case, it is typically not straightforward to determine *a priori* whether the assumption of $\bar{\mathcal{E}} = \bar{\mathcal{T}}$ is met for a given system and choice of basis. Here, when calculating the optimal predictions and indicators numerically, we use the same procedure as described for the classical case. In our study, we only encounter samples during evaluation where $\boldsymbol{x} \notin \bar{\mathcal{T}}$ for the XXZ model. The error resulting from neglecting the second contribution in Equation (B.1) is marginal, as the probability of drawing such inputs across the parameter range is found to be small. Note that the optimal indicator of SL obtained in such a manner correctly reveals the quantum phase transition in the XXZ (see Fig 3.14). The optimal predictions calculated via this procedure correspond to the probability of measuring the ferromagnetic ground state (see Section 3.6.4). For the above reasons, we expect that the optimal predictions of SL are capable of revealing phase transitions even if $\bar{\mathcal{E}} \neq \bar{\mathcal{T}}$.

A relevant scenario in which the assumption that $\bar{\mathcal{E}} = \bar{\mathcal{T}}$ is violated occurs when the system transitions between multiple phases as the tuning parameter is varied. Then, inputs drawn in the phases present in the middle of the sampled range of the tuning parameter may not be present in the two boundary phases. By dropping the second contribution in Equation (B.1), we may still faithfully detect the transition between the first and second phase (with the phases being arranged in order of increasing values of γ). However, all subsequent phase boundaries will then likely be missed. In the future, it will be of interest to lift the assumption of $\bar{\mathcal{E}} = \bar{\mathcal{T}}$ underlying the optimal predictions through appropriate interpolation schemes [Jacot *et al.*, 2018; Greplova *et al.*, 2020; Huang *et al.*, 2022b], which would allow for the generalization capabilities of SL (based on optimal predictors) to be explored.

Appendix C

Influence of Parameter Range on Optimal Indicators

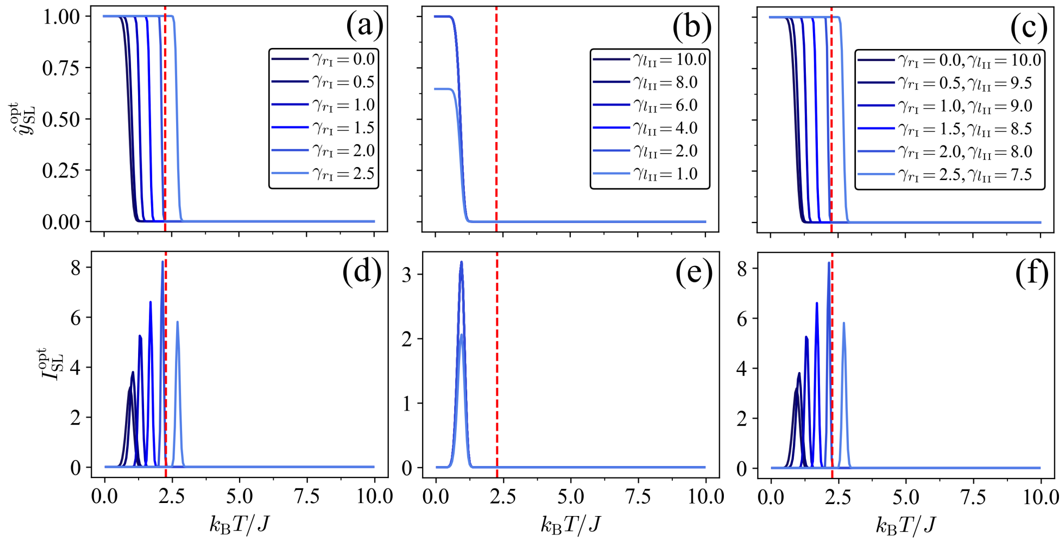

FIGURE C.1: (a)-(c) Mean optimal prediction $\hat{y}_{\mathrm{SL}}^{\mathrm{opt}}$ of SL and (d)-(f) the corresponding indicator $I_{\mathrm{SL}}^{\mathrm{opt}}$ for the Ising model ($L = 60$) with dimensionless temperature as tuning parameter $\gamma = k_{\mathrm{B}}T/J$, where $\gamma_1 = 0$, $\gamma_K = 10$, and $\Delta\gamma = 0.05$, for various choices of regions I and II ranging from $\gamma_{l_{\mathrm{I}}}$ to $\gamma_{r_{\mathrm{I}}}$ and $\gamma_{l_{\mathrm{II}}}$ to $\gamma_{r_{\mathrm{II}}}$, respectively. The critical temperature of the Ising model is highlighted by a red dashed line. In panels (a),(d) region I is varied, in (b),(e) region II is varied, and in (c),(f) both regions are varied simultaneously.

In this appendix, we discuss the influence of the choice of sampled parameter range on the optimal predictions and indicators of SL, LBC, and PBM. In particular, in the case of SL, we discuss how the results vary depending on the choice of regions I and II constituting the training data. We will focus on the approaches to these methods discussed in Chapter 3. As an example, we consider the Ising model (as discussed in Section 3.6).

Figure C.1 shows the optimal predictions and indicators of SL as a function of the tuning parameter for various different choices of regions I and II (with default values $l_{\mathrm{I}} = r_{\mathrm{I}} = 1$ and $l_{\mathrm{II}} = r_{\mathrm{II}} = K$), i.e., the training data. In particular, Figures C.1(a) and (d) show how the result varies if region I, ranging from $\gamma_{l_{\mathrm{I}}}$ to $\gamma_{r_{\mathrm{I}}}$ with $\gamma_{l_{\mathrm{I}}} = 0$, is extended to encompass a large range of temperatures. For small deviations in $\gamma_{r_{\mathrm{I}}}$, i.e., $\gamma_{r_{\mathrm{I}}} \approx 0$, we have $\forall E \neq E_{\mathrm{gs}} : P_{\mathrm{I}}(E_{\mathrm{gs}}) \gg P_{\mathrm{I}}(E)$. Therefore, the optimal predictions still approximately match the analytical expression stated in Equation (3.65).

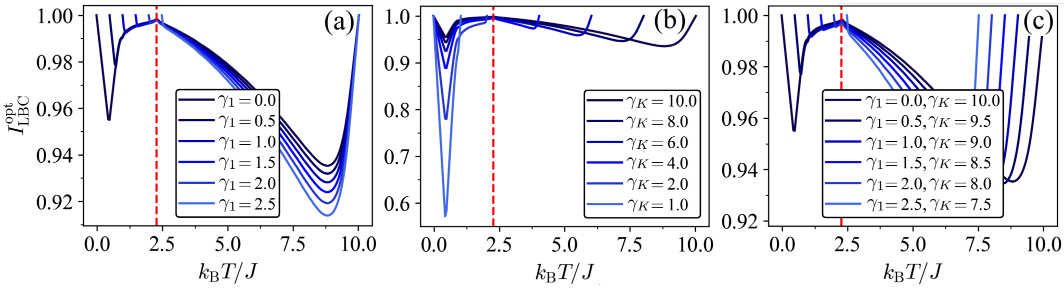

FIGURE C.2: Optimal indicator $I_{\mathrm{LBC}}^{\mathrm{opt}}$ of LBC for the Ising model ($L = 60$) with dimensionless temperature as tuning parameter $\gamma = k_{\mathrm{B}}T/J$ for various choices of γ_1 and γ_K with $\Delta\gamma = 0.05$. The critical temperature of the Ising model is highlighted by a red dashed line. In panel (a) γ_1 is varied and $\gamma_K = 10$, in (b) γ_K is varied and $\gamma_1 = 0$, and in (c) both borders of the parameter range are varied simultaneously.

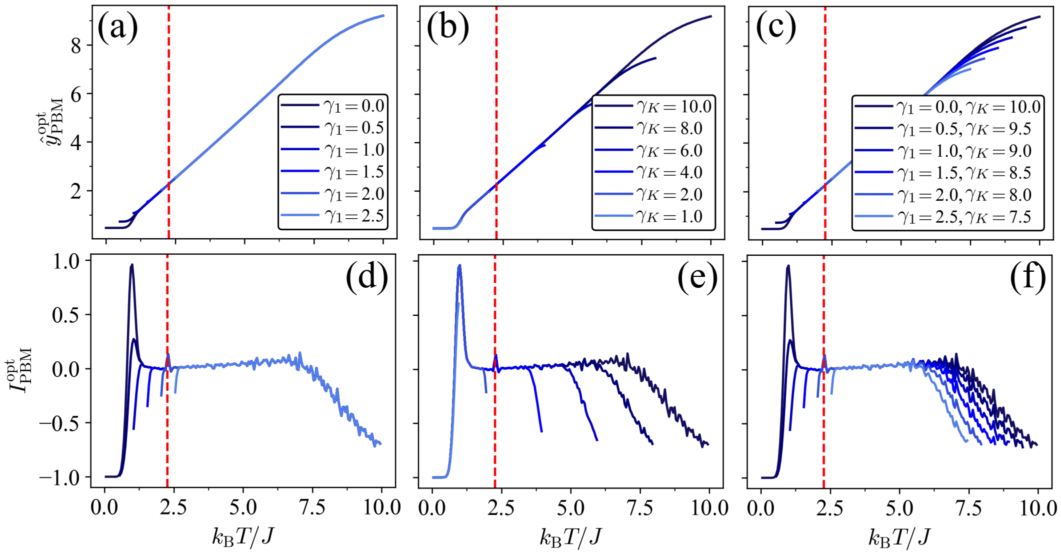

FIGURE C.3: (a)-(c) Mean optimal prediction $\hat{y}_{\mathrm{PBM}}^{\mathrm{opt}}$ of PBM and (d)-(f) the corresponding indicator $I_{\mathrm{PBM}}^{\mathrm{opt}}$ for the Ising model ($L = 60$) with dimensionless temperature as tuning parameter $\gamma = k_{\mathrm{B}}T/J$ for various choices of γ_1 and γ_K with $\Delta\gamma = 0.05$. The critical temperature of the Ising model is highlighted by a red dashed line. In panels (a),(d) γ_1 is varied and $\gamma_K = 10$, in (b),(e) γ_K is varied and $\gamma_1 = 0$, and in (c),(f) borders of the parameter range are varied simultaneously.

Recall that the results for the Ising model, IGT, and XY model shown in Section 3.6 were obtained with $\gamma_{l_{\mathrm{I}}} = \gamma_{r_{\mathrm{I}}} > 0$. As $\gamma_{r_{\mathrm{I}}}$ is increased further, low-lying energy states become populated in region I and start to contribute. Given that they are assigned the label 1, this shifts the corresponding peak in the indicator signal toward higher temperatures. The peak in the indicator signal can even occur at temperatures larger than the critical temperature if region I extends beyond the critical temperature (see Figure C.1). Interestingly, the critical temperature is marked by the choice of region I for which the peak in the optimal indicator is largest, i.e., the corresponding optimal predictions change most rapidly. This is akin to LBC, where multiple bipartitions of the parameter range are investigated. In this case, the critical temperature of the Ising model is correctly highlighted by the (non-trivial) partition where the optimal model achieves the highest classification accuracy corresponding to a sharp classification boundary.

Figures C.1(b) and (e) show the results when varying region II while region I is chosen according to $\gamma_{l_{\mathrm{I}}} = \gamma_{r_{\mathrm{I}}} = 0$. From Equation (3.69), we have that

$$\hat{y}_{\mathrm{SL}}^{\mathrm{opt}}(T) = \frac{P(E_{\mathrm{gs}}|T)}{1 + P_{\mathrm{II}}(E_{\mathrm{gs}})} \propto P(E_{\mathrm{gs}}|T), \tag{C.1}$$

i.e., we know that this change only rescales $\hat{y}_{\mathrm{SL}}^{\mathrm{opt}}(T)$. Therefore, the peak position does not shift. A significant change in the optimal predictions occurs only once $\gamma_{l_{\mathrm{II}}} \ll \gamma_c$ such that $P_{\mathrm{II}}(E_{\mathrm{gs}}) \gg 0$. Figures C.1(c) and (f) show the results when varying both region I and II. Here, both effects described above are at play. Overall, this demonstrates that the results obtained by SL can vary substantially if regions I and II are *not* chosen deep within the two phases. However, this generally constitutes the basis of SL. What is considered "deep within a phase" depends on the particular physical system at hand, with the crucial point being that the probability distribution underlying the input data only changes marginally in such a region.

For completeness, we also show the results for LBC and PBM when the overall parameter range is varied in Figures C.2 and C.3, respectively. Both methods are robust even against large changes in the chosen parameter range, and characteristic signals only disappear once they fall out of the corresponding parameter range.

Appendix D

Reproducing Optimal Predictions and Indicators Using Neural Networks

Figures D.1-D.6 show the predictions and indicators of the three ML methods obtained using NNs after long training for all six physical systems considered in Section 3.6. This showcases explicitly that the optimal predictions and indicators discussed in Chapter 3 can indeed be reproduced by NN-based models. For details on the NN-based training, see Section 3.7.1.

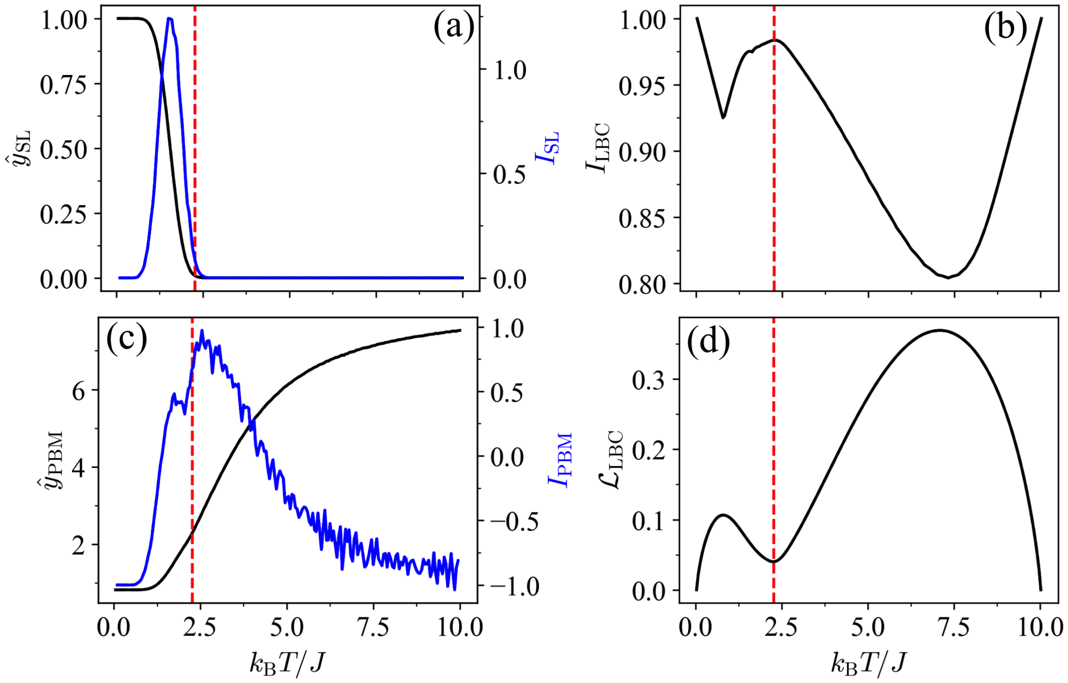

FIGURE D.1: Results for the Ising model ($L = 10$) using NNs. The NNs used in SL, LBC, and PBM were trained for 10000, 1000, and 5000 epochs, respectively. The tuning parameter ranges from $\gamma_1 = 0.05$ to $\gamma_K = 10$ with $\Delta\gamma = 0.05$. (a) Mean prediction $\hat{y}_{\mathrm{SL}}(\gamma)$ obtained using the analytical expression (black, solid) or an NN (black, dashed), as well as the corresponding indicator $I_{\mathrm{SL}}(\gamma)$ (blue). Here, we choose $r_{\mathrm{I}} = 1$ and $l_{\mathrm{II}} = K$. (b) The indicator of LBC, I_{LBC}, obtained using the analytical expression (black, solid) or an NN (black, dashed). (c) Mean prediction $\hat{y}_{\mathrm{PBM}}(\gamma)$ of PBM obtained using the analytical expression (black, solid) or an NN (black, dashed), as well as the corresponding indicator $I_{\mathrm{PBM}}(\gamma)$ (blue). (d) Value of the loss function in LBC, $\mathcal{L}_{\mathrm{LBC}}$, for each bipartition point γ^{bp} obtained using the analytical expression (black, solid) or evaluated after NN training (black, dashed). In all three models, the NNs were comprised of three hidden layers with 64 nodes each, and the learning rate was set to 0.001.

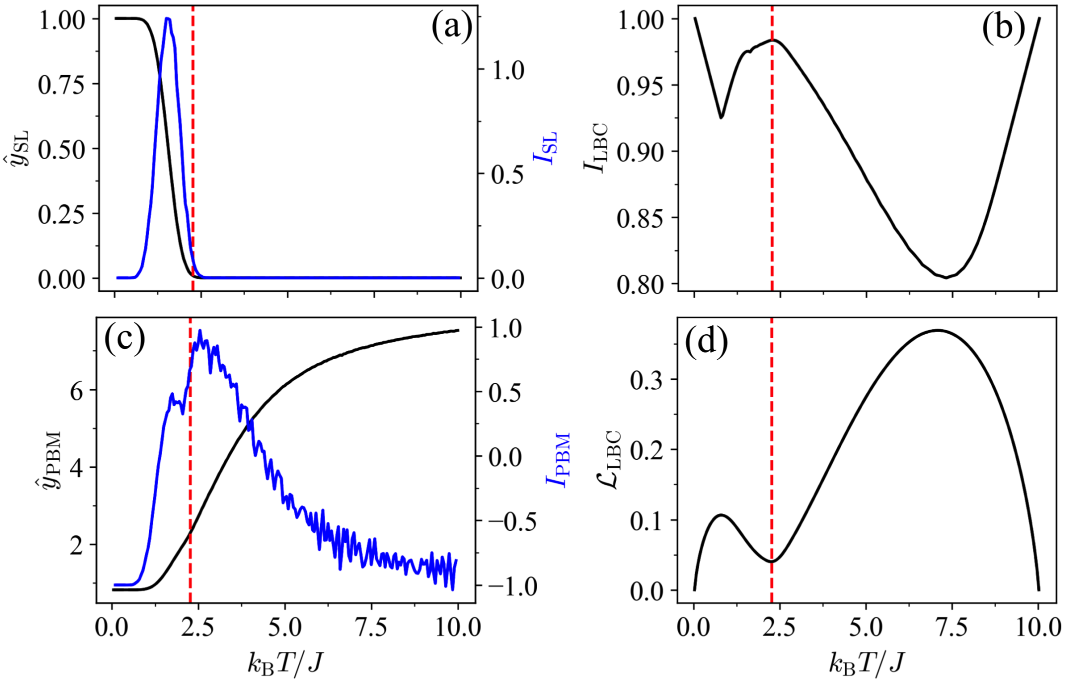

FIGURE D.2: Results for the IGT ($L = 4$) using NNs. The NNs used in SL, LBC, and PBM were trained for 10000, 1000, and 5000 epochs, respectively. The tuning parameter ranges from $\gamma_1 = 0.05$ to $\gamma_K = 5$ with $\Delta\gamma = 0.05$. (a) Mean prediction $\hat{y}_{\mathrm{SL}}(\gamma)$ obtained using the analytical expression (black, solid) or an NN (black, dashed), as well as the corresponding indicator $I_{\mathrm{SL}}(\gamma)$ (blue). Here, we choose $r_{\mathrm{I}} = 1$ and $l_{\mathrm{II}} = K$. (b) The indicator of LBC, I_{LBC}, obtained using the analytical expression (black, solid) or an NN (black, dashed). (c) Mean prediction $\hat{y}_{\mathrm{PBM}}(\gamma)$ of PBM obtained using the analytical expression (black, solid) or an NN (black, dashed), as well as the corresponding indicator $I_{\mathrm{PBM}}(\gamma)$ (blue). (d) Value of the loss function in LBC, $\mathcal{L}_{\mathrm{LBC}}$, for each bipartition point γ^{bp} obtained using the analytical expression (black, solid) or evaluated after NN training (black, dashed). In all three models, the NNs were comprised of three hidden layers with 64 nodes each, and the learning rate was set to 0.001.

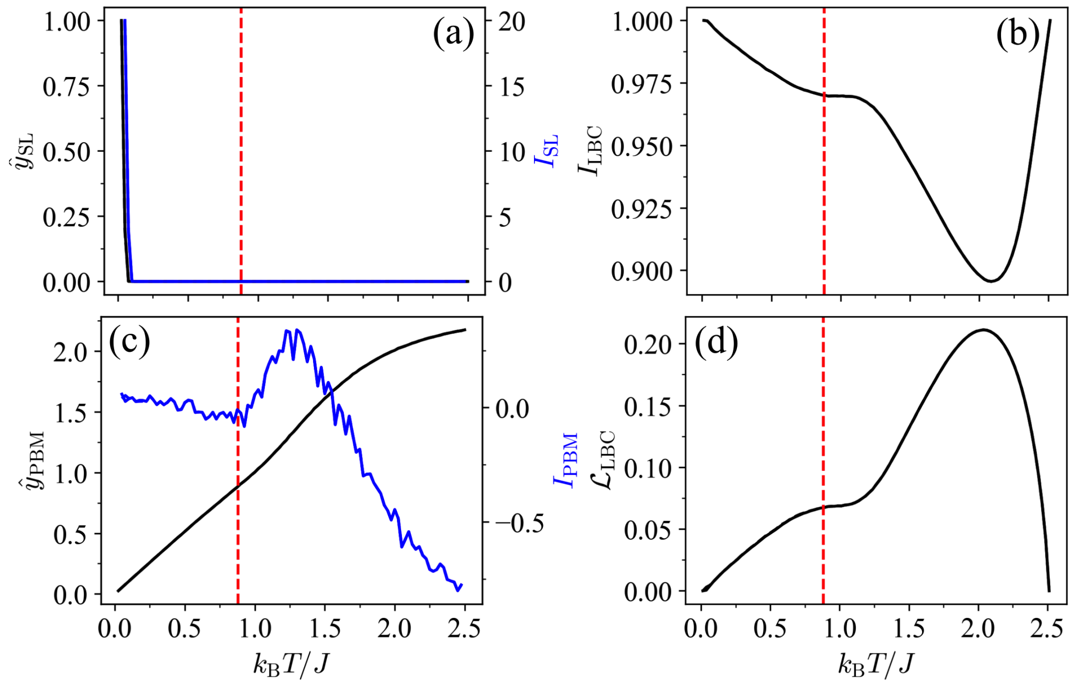

Figure D.3: Results for the XY model ($L = 10$) using NNs. The NNs used in SL, LBC, and PBM were trained for 10000, 1000, and 10000 epochs, respectively. The tuning parameter ranges from $\gamma_1 = 0.025$ to $\gamma_K = 2.5$ with $\Delta\gamma = 0.025$. The critical value of the tuning parameter $\gamma_{\mathrm{c}} = k_{\mathrm{B}}T_{\mathrm{c}}/J$ is highlighted in red. (a) Mean prediction $\hat{y}_{\mathrm{SL}}(\gamma)$ obtained using the analytical expression (black, solid) or an NN (black, dashed), as well as the corresponding indicator $I_{\mathrm{SL}}(\gamma)$ (blue). Here, we choose $r_{\mathrm{I}} = 1$ and $l_{\mathrm{II}} = K$. (b) The indicator of LBC, I_{LBC}, obtained using the analytical expression (black, solid) or an NN (black, dashed). (c) Mean prediction $\hat{y}_{\mathrm{PBM}}(\gamma)$ of PBM obtained using the analytical expression (black, solid) or an NN (black, dashed), as well as the corresponding indicator $I_{\mathrm{PBM}}(\gamma)$ (blue). (d) Value of the loss function in LBC, $\mathcal{L}_{\mathrm{LBC}}$, for each bipartition point γ^{bp} obtained using the analytical expression (black, solid) or evaluated after NN training (black, dashed). In all three models, the NNs were comprised of three hidden layers with 64 nodes each, and the learning rate was set to 0.001.

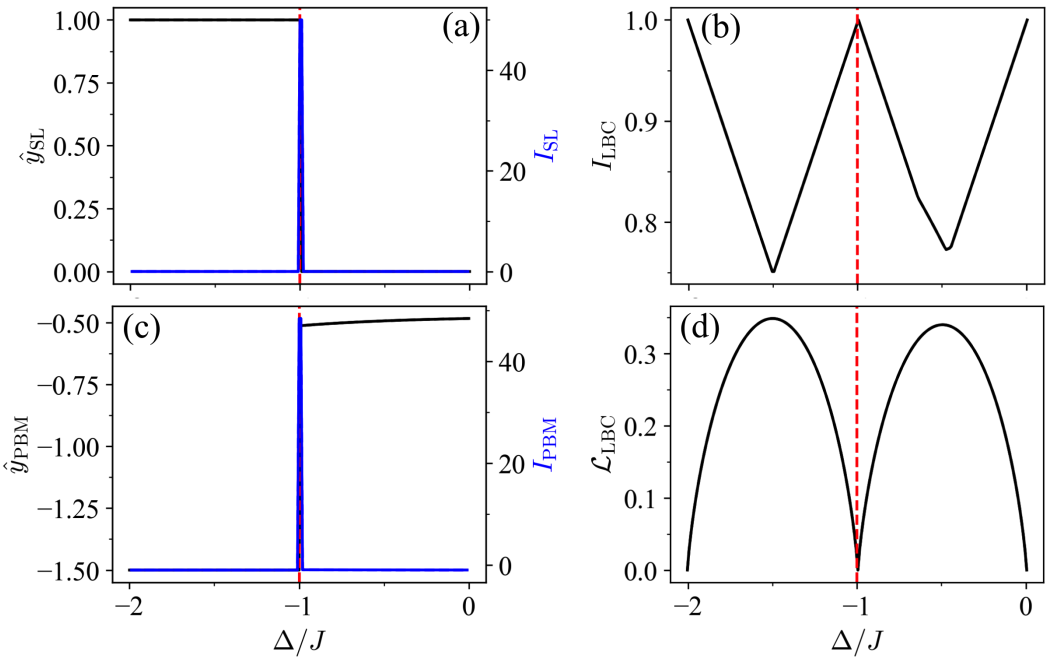

Figure D.4: Results for the XXZ chain ($L = 4$) using NNs. The NNs used in SL, LBC, and PBM were trained for 10000, 1000, and 5000 epochs, respectively. The tuning parameter ranges from $\gamma_1 = -2$ to $\gamma_K = 0$ with $\Delta\gamma = 0.01$. The critical value of the tuning parameter $\gamma_{\rm c} = \Delta_{\rm c}/J$ is highlighted in red. (a) Mean prediction $\hat{y}_{\rm SL}(\gamma)$ obtained using the analytical expression (black, solid) or an NN (black, dashed), as well as the corresponding indicator $I_{\rm SL}(\gamma)$ (blue). Here, we choose $r_{\rm I} = 1$ and $l_{\rm II} = K$. (b) The indicator of LBC, $I_{\rm LBC}$, obtained using the analytical expression (black, solid) or an NN (black, dashed). (c) Mean prediction $\hat{y}_{\rm PBM}(\gamma)$ of PBM obtained using the analytical expression (black, solid) or an NN (black, dashed), as well as the corresponding indicator $I_{\rm PBM}(\gamma)$ (blue). (d) Value of the loss function in LBC, $\mathcal{L}_{\rm LBC}$, for each bipartition point $\gamma^{\rm bp}$ obtained using the analytical expression (black, solid) or evaluated after NN training (black, dashed). Here, the NNs were comprised of three hidden layers with 64 nodes each, and the learning rate was set to 0.001.

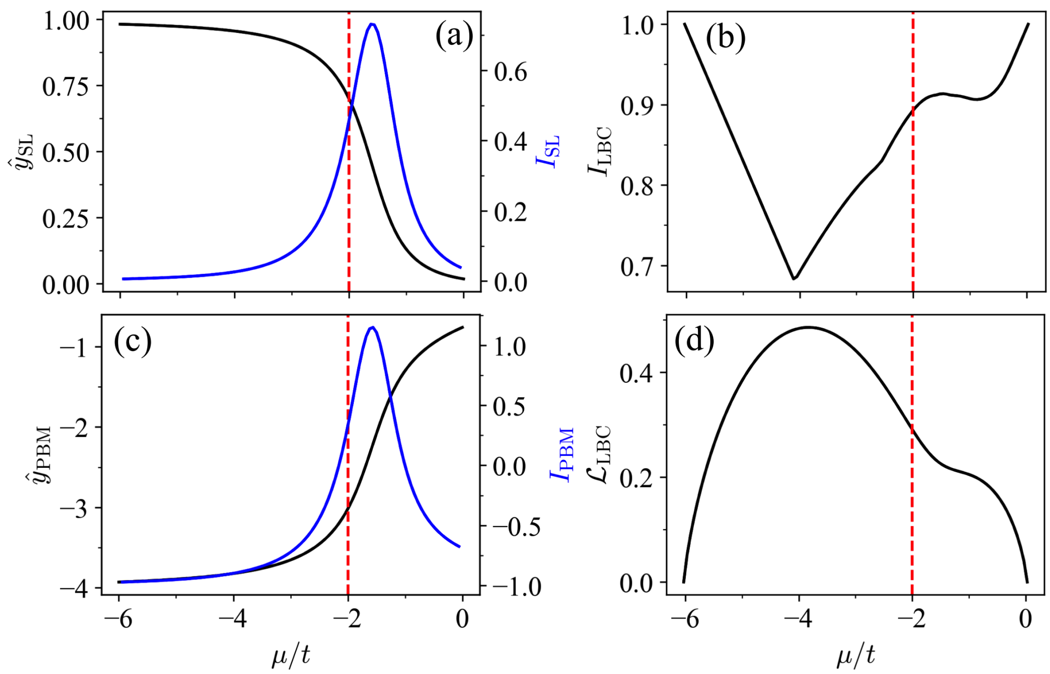

FIGURE D.5: Results for the Kitaev chain ($L = 10$) using NNs. The NNs used in SL, LBC, and PBM were trained for 5000, 500, and 1000 epochs, respectively. The tuning parameter ranges from $\gamma_1 = -6$ to $\gamma_K = 0$ with $\Delta\gamma = 0.06$. The critical value of the tuning parameter $\gamma_{\mathrm{c}} = \mu_{\mathrm{c}}/t$ is highlighted in red. (a) Mean prediction $\hat{y}_{\mathrm{SL}}(\gamma)$ obtained using the analytical expression (black, solid) or an NN (black, dashed), as well as the corresponding indicator $I_{\mathrm{SL}}(\gamma)$ (blue). Here, we choose $r_{\mathrm{I}} = 1$ and $l_{\mathrm{II}} = K$. (b) The indicator of LBC, I_{LBC}, obtained using the analytical expression (black, solid) or an NN (black, dashed). (c) Mean prediction $\hat{y}_{\mathrm{PBM}}(\gamma)$ of PBM obtained using the analytical expression (black, solid) or an NN (black, dashed), as well as the corresponding indicator $I_{\mathrm{PBM}}(\gamma)$ (blue). (d) Value of the loss function in LBC, $\mathcal{L}_{\mathrm{LBC}}$, for each bipartition point γ^{bp} obtained using the analytical expression (black, solid) or evaluated after NN training (black, dashed). Here, we use two hidden layers with 128 nodes each, followed by three hidden layers with 64 nodes each, and the learning rate was set to 0.001.

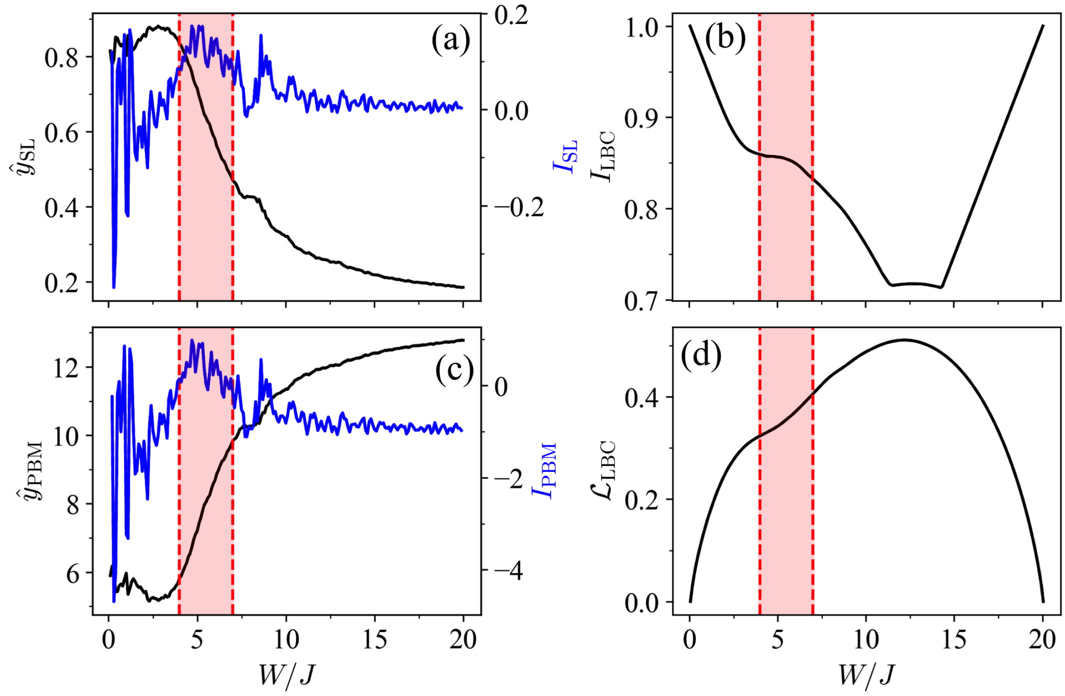

Figure D.6: Results for the many-body localization phase transition in the Bose-Hubbard model ($L = 6$) using NNs. The NNs used in SL, LBC, and PBM were trained for 10000, 300, and 1000 epochs, respectively. The tuning parameter ranges from $\gamma_1 = 0.1$ to $\gamma_K = 20$ with $\Delta\gamma = 0.1$. The critical value of the tuning parameter $\gamma_c = W_c/J$ is highlighted in red. (a) Mean prediction $\hat{y}_{\mathrm{SL}}(\gamma)$ obtained using the analytical expression (black, solid) or an NN (black, dashed), as well as the corresponding indicator $I_{\mathrm{SL}}(\gamma)$ (blue). Here, we choose $r_{\mathrm{I}} = 1$ and $l_{\mathrm{II}} = K$. (b) The indicator of LBC, I_{LBC}, obtained using the analytical expression (black, solid) or an NN (black, dashed). (c) Mean prediction $\hat{y}_{\mathrm{PBM}}(\gamma)$ of PBM obtained using the analytical expression (black, solid) or an NN (black, dashed), as well as the corresponding indicator $I_{\mathrm{PBM}}(\gamma)$ (blue). (d) Value of the loss function in LBC, $\mathcal{L}_{\mathrm{LBC}}$, for each bipartition point γ^{bp} obtained using the analytical expression (black, solid) or evaluated after NN training (black, dashed). Here, we use two hidden layers with 128 nodes each, followed by three hidden layers with 64 nodes each, and the learning rate was set to 0.001.

Appendix E

Discriminative Cooperative Networks

In this appendix, we discuss an alternative scheme for mapping out phase diagrams in two-dimensional parameter spaces using LBC proposed by Liu and van Nieuwenburg [2018] that we mention in Chapter 4. Instead of a brute-force search of the entire parameter space for all possible phase boundaries, the predicted phase boundary is modeled as a parametrized curve using an active contour model called *snake* [Kass *et al.*, 1988]. This snake partitions the parameter space locally and is driven via internal forces that, e.g., prevent bending and stretching, as well as external forces aiming to minimize the overall classification error. In Liu and van Nieuwenburg [2018], the external force is generated in an interplay between a guesser network and learner network, together called *discriminative cooperative networks*, that are optimized via a joint cost function. The guesser provides labels for the data which the learner should reproduce. In each step, the guesser tries to provide a better set of labels based on the predictions of the learner to cooperatively minimize the classification error (i.e., the corresponding loss).

Let us first consider a one-dimensional parameter space. Following [Liu and van Nieuwenburg, 2018], we use the following sigmoidal guesser

$$\tilde{P}_{\boldsymbol{\theta}_G}(0|\gamma) = \frac{1}{1 + e^{(\gamma_G - \gamma)/\sigma_G}}, \tag{E.1}$$

where $\tilde{P}_{\boldsymbol{\theta}_G}(1|\gamma) = 1 - \tilde{P}_{\boldsymbol{\theta}_G}(0|\gamma)$. The guesser is characterized by two parameters $\boldsymbol{\theta}_G = (\gamma_G, \sigma_G)$, where γ_G corresponds to the guessed transition point and σ_G determines the sharpness of the transition. The learner is given by $\tilde{P}_{\boldsymbol{\theta}_L}(y|\boldsymbol{x})$. The guesser and learner are optimized jointly using a cross-entropy loss function

$$\mathcal{L}(\boldsymbol{\theta}) = -\frac{1}{|\mathcal{Y}|}\sum_{y\in\mathcal{Y}}\frac{1}{|\Gamma_y|}\sum_{\gamma\in\Gamma_y}\frac{1}{|\mathcal{D}_\gamma|}\sum_{\boldsymbol{x}\in\mathcal{D}_\gamma}\tilde{P}_{\boldsymbol{\theta}_G}(y|\gamma)\ln\left[\tilde{P}_{\boldsymbol{\theta}_L}(y|\boldsymbol{x})\right], \tag{E.2}$$

where $\boldsymbol{\theta} = (\boldsymbol{\theta}_G, \boldsymbol{\theta}_L)$. The parameters of the two networks can now be optimized using gradient descent on the loss function in Equation (E.2). This corresponds to a discriminative approach and has been used in [Liu and van Nieuwenburg, 2018].

Moving to two-dimensional parameter spaces, one can utilize the snake as a parametrization for the guesser. The snake refers to a discretized curve of linked nodes,

$$\boldsymbol{r}(s) = \Big(\gamma_1(s), \gamma_2(s)\Big), \tag{E.3}$$

parametrized by $s \in [0, 1]$ (assuming an open snake), see Figure E.1(a) for an illustration. The snake moves to minimize its total energy $E_{\text{tot}} = E_{\text{int}} + E_{\text{ext}}$. The internal energy

$$E_{\text{int}} = \int_0^1 \left(\alpha \left|\left| \frac{\partial \boldsymbol{r}}{\partial s} \right|\right|_2^2 + \beta \left|\left| \frac{\partial^2 \boldsymbol{r}}{\partial s^2} \right|\right|_2^2 \right) ds \tag{E.4}$$

is introduced to make the snake smoother, where the hyperparameter α penalizes the stretching of the snake and β penalizes its bending. The snake can sense its surroundings at each node through $2l$ sampled points perpendicular to the snake within a distance $l\sigma$ [see Figure E.1(a)]. The overall guesser function is comprised of local guesser functions evaluated at each node that sense the direction perpendicular to the snake. The external energy E_{ext} of the snake corresponds to the overall loss obtained by summing the cross-entropy losses of all individual one-dimensional guessers [cf. Equation (E.2)]. This energy gives rise to an external force $-\delta E_{\text{ext}}/\delta \mathbf{r}$ pointing perpendicular to the snake at each node.

Here, we replace the learner network at each node with a corresponding generative classifier. This renders updating the learner in each step obsolete. To construct the generative classifier, we derive the (empirically) optimal predictor based on the loss in Equation (E.2). Following the procedure outline in Section 4.5, we obtain

$$P_{\text{emp}}(y|\boldsymbol{x}) = \frac{\sum_{\gamma \in \Gamma_y} \tilde{P}_{\boldsymbol{\theta}_G}(y|\gamma) \frac{\mathcal{D}_\gamma(\boldsymbol{x})}{|\mathcal{D}_\gamma|} \frac{1}{|\mathcal{Y}|} \frac{1}{|\Gamma_y|}}{\sum_{y' \in \mathcal{Y}} \sum_{\gamma' \in \Gamma_{y'}} \tilde{P}_{\boldsymbol{\theta}_G}(y'|\gamma') \frac{\mathcal{D}_{\gamma'}(\boldsymbol{x})}{|\mathcal{D}_{\gamma'}|} \frac{1}{|\mathcal{Y}|} \frac{1}{|\Gamma_{y'}|}}. \tag{E.5}$$

In the infinite-data limit, this converges to

$$P(y|\boldsymbol{x}) = \frac{\sum_{\gamma \in \Gamma_y} \tilde{P}_{\boldsymbol{\theta}_G}(y|\gamma) P(\boldsymbol{x}|\gamma) P(y) P(\gamma|y)}{\sum_{y' \in \mathcal{Y}} \sum_{\gamma' \in \Gamma_{y'}} \tilde{P}_{\boldsymbol{\theta}_G}(y'|\gamma') P(\boldsymbol{x}|\gamma') P(y) P(\gamma'|y')}, \tag{E.6}$$

with a uniform prior over the classes, $P(y) = 1/|\mathcal{Y}|$ for any $y \in \mathcal{Y}$, and $P(\gamma|y) = 1/|\Gamma_y|$ if $\gamma \in \Gamma_y$ and zero otherwise. Thus, we can obtain a generative classifier by modeling $P(\boldsymbol{x}|\gamma)$ in Equation (E.6). Note that the expression in Equation (4.37) for binary labels is recovered as a special case.

In our implementation, we fix the width parameter σ_G of the sigmoid guesser to be $\sigma_G = \sigma/10$, where σ is the width parameter of the snake. We reduce σ exponentially during the optimization according to

$$\sigma_k = \sigma_{\text{end}} + (\sigma_{\text{start}} - \sigma_{\text{end}})\kappa^k, \tag{E.7}$$

where k denotes the current epoch, $\sigma_{\text{start/end}}$ are the start and end values, and κ is the decay rate. We set $\sigma_{\text{end}} = \Delta\gamma$, where $\Delta\gamma$ is the spacing between neighboring sampled points in parameter space and $\sigma_{\text{start}} = 5\sigma_{\text{end}}$. This choice yields a strong gradient signal early on that becomes weaker but more accurate at later stages, which facilitates the convergence of the snake to the true underlying phase boundary. We construct models for $P(\boldsymbol{x}|\boldsymbol{\gamma})$ at points $\boldsymbol{\gamma}$ not contained within the initial sampled set Γ from $\{\tilde{P}(\cdot|\boldsymbol{\gamma})\}_{\boldsymbol{\gamma}\in\Gamma}$ using bilinear interpolation. Gradients of the energy with respect to the node positions are calculated using finite differencing. We minimize the snake's total energy using gradient-based optimization with Adam [Kingma and Ba, 2014].

The results obtained using this scheme with a generative learner for the anisotropic Ising model (described in Section 4.4.1) are shown in Figure E.1(b). While the snake eventually finds the underlying phase boundary, there are several downsides to this scheme. First, the number of unique classification tasks that need to be solved over the course of the training is $N_{\text{epochs}} \cdot N_{\text{nodes}} = 400 \cdot 20 = 8000$. This is larger than the

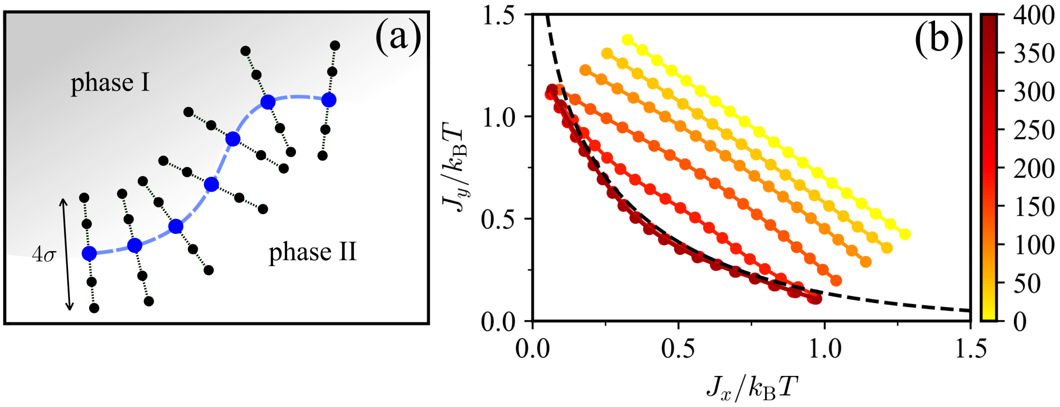

FIGURE E.1: (a) Schematic illustration of the snake model. The blue circles represent the snake nodes (here 7). The normal (sensing) direction at each node is shown as a black dashed line and the relevant $2l$ sampled points within a distance $2l\sigma$ are shown as black circles (here $l = 2$). (b) Snake boundary for the anisotropic Ising model on a square lattice ($L = 20$; $J_x/k_\mathrm{B}T, J_y/k_\mathrm{B}T \geq 0$) obtained using a generative classifier where the color bar denotes the training epoch. The snake consists of 20 nodes with $l = 4$ and hyperparameters $\alpha = 0.002$, $\beta = 0.4$, $\kappa = 0.9$, and learning rates of 10^{-4} and $5 \cdot 10^{-4}$ associated with E_int and E_ext, respectively. The set Γ is composed of a uniform grid with 30 points for each axis and $|\mathcal{D}_{\boldsymbol{\gamma}}| = 10^5$ for all $\boldsymbol{\gamma} \in \Gamma$. Onsager's analytical solution for the phase boundary is shown as a black dashed line.

number of unique classification tasks that need to be solved within the LBC extension proposed in Section 4.2.2, which is given by $|\Gamma| \cdot 2 = 900 \cdot 2 = 1800$. As such, there is no gain in computation time compared to a brute-force search of the parameter space (as performed within the LBC method described in Section 4.2.2). The snake scheme also has a multitude of hyperparameters: α, β, σ_start, κ, l, learning rates, as well as the initial snake positioning and snake topology (e.g., whether the snake is open or closed and whether certain nodes should remain fixed throughout training). Thus, to run the snake scheme one has to perform hyperparameter tuning. This involves additional computational effort or prior knowledge of the underlying phase diagram. In particular, we find the results to depend heavily on α, β, as well as the learning rates and initial position of the snake. Moreover, the scheme is not reliable in the presence of more than two phases as the snake will typically converge to one of the phase boundaries, missing the remaining ones [see Figure E.1(b)]. To get around this, the scheme must be run multiple times with different snake initializations (possibly guided by prior knowledge of the phase diagram).

Appendix F

Background Subtraction in Learning by Confusion

In this appendix, we discuss the attempt of Bohrdt *et al.* [2021] to construct a simplified indicator for phase transitions for LBC. Here we refer to LBC and its optimal predictions and indicators as defined in Chapter 3. If no phase transition is present, the (original) indicator of LBC exhibits a characteristic V-shape. Whereas the presence of a phase transition, a W-shape is expected. Ideally, we would like the indicator to be flat around zero in the absence of a phase transition and show a single peak in the presence of a transition. In [Bohrdt *et al.*, 2021], an attempt has been made to construct such a modified indicator by subtracting the V-shaped indicator signal in the case of indistinguishable data (case 1 in Section 3.3) as a baseline. However, we find that this procedure biases the transition point toward the center of the parameter range under consideration and hence does not seem viable.

Figure F.1 shows the optimal indicator in LBC for all physical systems considered in Section 3.6, as well as a modified version where the V-shaped indicator signal characteristic of indistinguishable data is subtracted. Note that this V-shaped indicator signal is computed separately for each system, i.e., parameter range. For all systems, we find that the modified indicator peaks near the center of the parameter range under consideration, whereas the original indicator signal peaks near the phase transition (red dashed line). This bias arises because the subtracted signal is lowest near the center of the parameter range. As such, the bias can be easily missed if the transition point is indeed located in the center of the chosen parameter range, see Figure F.1(d).

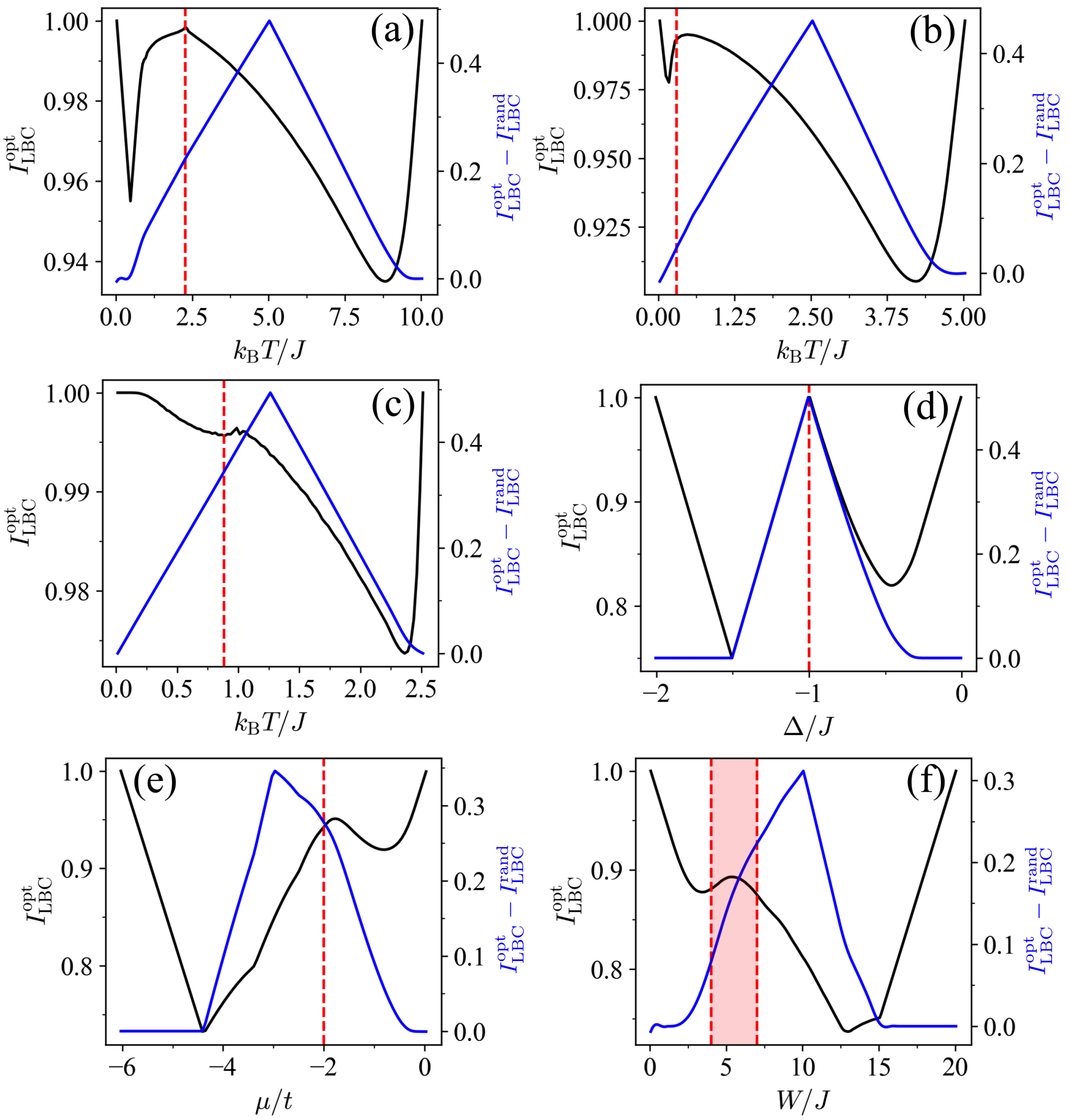

FIGURE F.1: Optimal indicator of LBC before (black) and after (blue) background subtraction for the (a) Ising model ($L = 60$), (b) IGT ($L = 28$), (c) XY model ($L = 60$), (d) XXZ chain ($L = 14$), (e) Kitaev chain ($L = 20$), and (f) Bose-Hubbard model ($L = 8$). The corresponding critical values of the tuning parameters are highlighted in red. For a discussion of the models and other numerical settings, see Section 3.6.

Appendix G

Relation Between Generative Adversarial Network Fidelity and the Fisher Information

In this appendix, we show that the generative adversarial network (GAN) fidelity that has been proposed in [Singh *et al.*, 2021] as an indicator of phase transitions is related to the square root of the system's Fisher information. The GAN fidelity is defined as

$$F_{\text{GAN}}(\gamma) = \frac{1}{\Delta\gamma}\mathbb{E}_{z\sim p}\Big[\text{discr}_{\boldsymbol{\theta}_D}\Big(\text{gen}_{\boldsymbol{\theta}_G}(z|\gamma),\gamma\Big) - \text{discr}_{\boldsymbol{\theta}_D}\Big(\text{gen}_{\boldsymbol{\theta}_G}(z|\gamma),\gamma+\Delta\gamma\Big)\Big]. \tag{G.1}$$

The function $\text{discr}_{\boldsymbol{\theta}_D}(\boldsymbol{x},\gamma)$ is a *discriminator* trained to output 1 for samples $\boldsymbol{x}$ drawn from the probability distribution $P(\cdot|\gamma)$ and 0 otherwise, whereas the function $\text{gen}_{\boldsymbol{\theta}_G}(\cdot|\gamma) : \mathcal{Z} \to \mathcal{X}$ is a *generator* trained to produce samples from $P(\cdot|\gamma)$ when evaluated with $z \sim p$, where p is a simple prior distribution over the latent variable $z \in \mathcal{Z}$.

Let us analyze the ideal case in which both the generator and discriminator are optimal, i.e., implement optimal strategies for generating and discriminating samples, respectively. In this case, we have

$$F^{\text{opt}}_{\text{GAN}}(\gamma) = \frac{1}{\Delta\gamma}\mathbb{E}_{\boldsymbol{x}\sim P(\cdot|\gamma)}\left[\text{discr}^{\text{opt}}\left(\boldsymbol{x},\gamma\right) - \text{discr}^{\text{opt}}\left(\boldsymbol{x},\gamma+\Delta\gamma\right)\right], \tag{G.2}$$

with

$$\text{discr}^{\text{opt}}\left(\boldsymbol{x},\gamma\right) = \frac{P(\boldsymbol{x}|\gamma)}{\sum_{\gamma'\in\Gamma}P(\boldsymbol{x}|\gamma')}, \tag{G.3}$$

where $\sum_{\gamma'\in\Gamma}P(\boldsymbol{x}|\gamma')$ is a normalization factor dependent on $\boldsymbol{x}$.

Taking the limit $\Delta\gamma \to 0$ in Equation (G.2), we have

$$\begin{aligned} F^{\text{opt}}_{\text{GAN}}(\gamma) &= -\mathbb{E}_{\boldsymbol{x}\sim P(\cdot|\gamma)}\left[\frac{1}{\sum_{\gamma'\in\Gamma}P(\boldsymbol{x}|\gamma')}\frac{\partial P(\boldsymbol{x}|\gamma)}{\partial\gamma}\right] \\ &\leq \sum_{\boldsymbol{x}\in\mathcal{X}}\text{discr}^{\text{opt}}(\boldsymbol{x},\gamma)\left|\frac{\partial P(\boldsymbol{x}|\gamma)}{\partial\gamma}\right| \\ &\leq \sum_{\boldsymbol{x}\in\mathcal{X}}\left|\frac{\partial P(\boldsymbol{x}|\gamma)}{\partial\gamma}\right| \\ &\leq \sqrt{\mathcal{F}}, \end{aligned} \tag{G.4}$$

where we have used the fact that $0 \leq \text{discr}^{\text{opt}}(\boldsymbol{x},\gamma) \leq 1$ for the second inequality and the Cauchy-Schwarz inequality for the last step.

Appendix H

Additional Scans of Transitions in Large Language Models

In this appendix, we provide additional evidence for transitions in LLM (cf. Chapter 8). Figure H.1 shows the results obtained when using (a) numbers γ^* different from 42 within the prompt *"γ is larger than γ^*. True or False?"* and (b) different semantic formulations of the original prompt *"γ is larger than 42. True or False?"*. In both cases, we observe a similar behavior as reported in Figure 8.1: The base Mistral model results in a flat linear dissimilarity, indicating that the model does not have a grasp on the natural ordering of integers. In contrast, when using the instruct-tuned Mistral model, the linear dissimilarity shows a peak at $\gamma^* - 0.5$ or $\gamma^* + 0.5$ [where $\gamma^* = 42$ for panel (b)], highlighting its ability to correctly order integers.

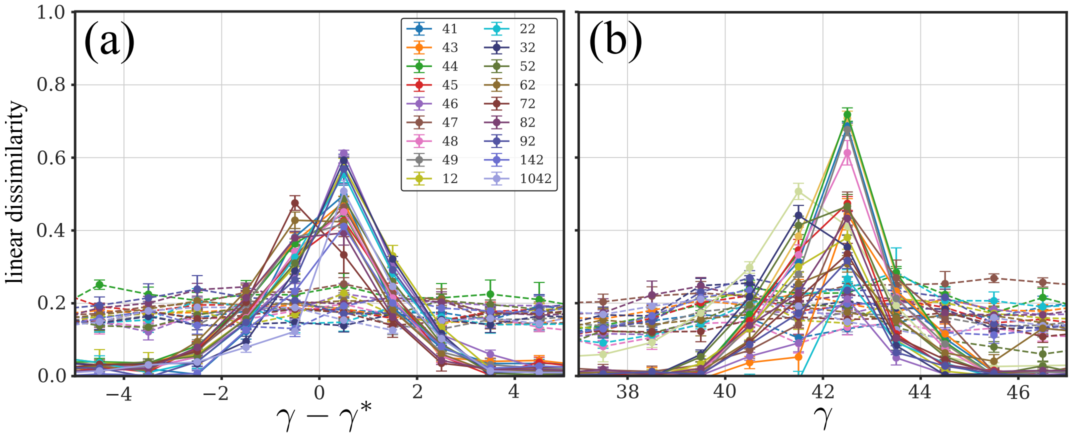

FIGURE H.1: Additional results for integer scan of Figure 8.1 in Chapter 8 using the same numerical settings with the Mistral-7B-Instruct model (full lines) and the Mistral-7B-base model (dashed lines). (a) Linear dissimilarity as a function of $\gamma - \gamma^*$ for the prompt *"γ is larger than γ^*. True or False?"*, where γ^* takes on a representative range of values (see legend). (b) Linear dissimilarity as a function of γ for various rephrasings of the original prompt *"γ is larger than 42. True or False?"* obtained from ChatGPT4. Examples include *"Is γ greater than 42?"* and *"Would you consider γ to be more than 42?"*.

Figure H.2 shows the linear dissimilarity and heat capacity as a function of temperature for Pythia models of various sizes with $N_{\text{tokens}} = 500$. The high-temperature transition is clearly visible across all four model sizes. The plot also highlights that the LLM size, i.e., the number of its trainable parameters, is not the quantity that is analogous to the number of constituents in a physical system. The heat capacity,

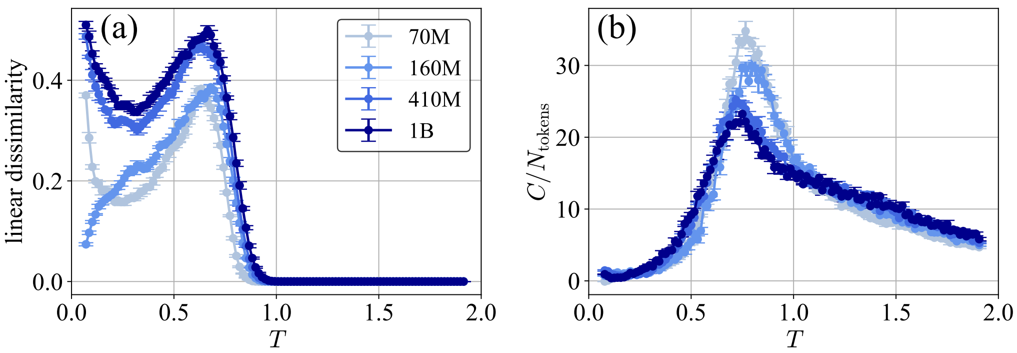

FIGURE H.2: High-temperature transition for Pythia models of various sizes in response to the prompt *"There's measuring the drapes, and then there's measuring the drapes on a house you haven't bought, a"* – an excerpt from OpenWebText [Aaron Gokaslan and Vanya Cohen, 2019]. The temperature range is $[10^{-4}, 2]$ and the number of output tokens is fixed to $N_{\text{tokens}} = 500$. (a) Linear dissimilarity measure ($l = 5$) and (b) heat capacity. [Number of text outputs generated per parameter value T: $|\mathcal{D}_T| = 400$. Error bars indicate the standard error of the mean over 4 batches (with 100 text outputs each).]

for example, is not observed to be larger for larger model sizes. Instead, in this context, the number of output tokens N_{tokens} takes on a role equivalent to the number of constituents and is the quantity relevant for performing finite-size scaling analyses.

Bibliography

Aaron Gokaslan and Vanya Cohen (2019): *OpenWebText Corpus*. Online; last accessed on 06/01/2025.

Abasto, D. F., A. Hamma, and P. Zanardi (2008): *Fidelity analysis of topological quantum phase transitions*. Phys. Rev. A **78**, 010301.

Acharya, R., L. Aghababaie-Beni, I. Aleiner, T. I. Andersen, M. Ansmann, F. Arute, K. Arya, A. Asfaw, N. Astrakhantsev, J. Atalaya, *et al.* (2024): *Quantum error correction below the surface code threshold.* arXiv:2408.13687.

Achiam, J., S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, *et al.* (2023): *GPT-4 technical report.* arXiv:2303.08774.

Achille, A., M. Rovere, and S. Soatto (2019): *Critical learning periods in deep neural networks*. In: *Int. Conf. Learn. Represent. – ICLR 2019.*

Afgani, M., S. Sinanovic, and H. Haas (2008): *Anomaly detection using the Kullback-Leibler divergence metric*. In: *2008 First International Symposium on Applied Sciences on Biomedical and Communication Technologies* (IEEE), pages 1–5.

AI@Meta (2024): *Llama 3 model card*. Online; last accessed on 21/05/2024.

Alet, F. and N. Laflorencie (2018): *Many-body localization: An introduction and selected topics*. C. R. Phys. **19**, 498.

Alicea, J. (2012): *New directions in the pursuit of Majorana fermions in solid state systems*. Rep. Prog. Phys. **75**, 076501.

Alishahi, A., G. Chrupała, and T. Linzen (2019): *Analyzing and interpreting neural networks for NLP: A report on the first BlackboxNLP workshop*. Nat. Lang. Eng. **25**, 543.

Altman, E., K. R. Brown, G. Carleo, L. D. Carr, E. Demler, C. Chin, B. DeMarco, S. E. Economou, M. A. Eriksson, K.-M. C. Fu, M. Greiner, K. R. Hazzard, R. G. Hulet, A. J. Kollár, B. L. Lev, *et al.* (2021): *Quantum simulators: Architectures and opportunities*. PRX Quantum **2**, 017003.

Alves, G. O. and G. T. Landi (2022): *Bayesian estimation for collisional thermometry.* Phys. Rev. A **105**, 012212.

Amico, L., R. Fazio, A. Osterloh, and V. Vedral (2008): *Entanglement in many-body systems*. Rev. Mod. Phys. **80**, 517.

Aminikhanghahi, S. and D. J. Cook (2017): *A survey of methods for time series change point detection*. Knowl. Inf. Syst. **51**, 339.

Anthropic (2023): *Model card and evaluations for Claude models*. Online; last accessed on 15/04/2024.

Arnold, J., F. Holtorf, F. Schäfer, and N. Lörch (2024a): *Phase transitions in the output distribution of large language models.* arXiv:2405.17088.

Arnold, J., F. Holtorf, F. Schäfer, and N. Lörch (2024b): *Code for: Phase transitions in the output distribution of large language models*. Online; last accessed on 03/01/2025.

Arnold, J., N. Lörch, F. Holtorf, and F. Schäfer (2023a): *Machine learning phase transitions: Connections to the Fisher information.* arXiv:2311.10710.

Arnold, J. and F. Schäfer (2022a): *Code for: Replacing neural networks by optimal analytical predictors for the detection of phase transitions*. Online; last accessed on 03/01/2025.

Arnold, J. and F. Schäfer (2022b): *Replacing neural networks by optimal analytical predictors for the detection of phase transitions*. Phys. Rev. X **12**, 031044.

Arnold, J., F. Schäfer, A. Edelman, and C. Bruder (2023b): *Code for: Mapping out phase diagrams with generative classifiers*. Online; last accessed on 03/01/2025.

Arnold, J., F. Schäfer, A. Edelman, and C. Bruder (2024c): *Mapping out phase diagrams with generative classifiers*. Phys. Rev. Lett. **132**, 207301.

Arnold, J., F. Schäfer, M. Žonda, and A. U. J. Lode (2021): *Interpretable and unsupervised phase classification*. Phys. Rev. Res. **3**, 033052.

Arnold, J., F. Schäfer, and N. Lörch (2023c): *Fast detection of phase transitions with multi-task learning-by-confusion.* arXiv:2311.09128.

Arnold, J., F. Schäfer, and N. Lörch (2023d): *Code for: Fast detection of phase transitions with multi-task learning-by-confusion*. Online; last accessed on 03/01/2025.

Arora, S. and A. Goyal (2023): *A theory for emergence of complex skills in language models.* arXiv:2307.15936.

Austin, J., A. Odena, M. Nye, M. Bosma, H. Michalewski, D. Dohan, E. Jiang, C. Cai, M. Terry, Q. Le, *et al.* (2021): *Program synthesis with large language models.* arXiv:2108.07732.

Azses, D., R. Haenel, Y. Naveh, R. Raussendorf, E. Sela, and E. G. Dalla Torre (2020): *Identification of symmetry-protected topological states on noisy quantum computers*. Phys. Rev. Lett. **125**, 120502.

Bahamondes, S. (2023): *Study of the possibility of phase transitions in LLMs*. Online; last accessed on 10/04/2024.

Banchi, L., P. Giorda, and P. Zanardi (2014): *Quantum information-geometry of dissipative quantum phase transitions*. Phys. Rev. E **89**, 022102.

Barratt, F., J. Dborin, M. Bal, V. Stojevic, F. Pollmann, and A. G. Green (2021): *Parallel quantum simulation of large systems on small NISQ computers*. Npj Quantum Inf. **7**, 1.

Basterrech, S. and M. Woźniak (2022): *Tracking changes using Kullback-Leibler divergence for the continual learning*. In: *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (IEEE), pages 3279–3285.

Baydin, A. G., B. A. Pearlmutter, A. A. Radul, and J. M. Siskind (2018): *Automatic differentiation in machine learning: A survey*. J. Mach. Learn. Res. **18**, 1.

Beach, M. J. S., A. Golubeva, and R. G. Melko (2018): *Machine learning vortices at the Kosterlitz-Thouless transition*. Phys. Rev. B **97**, 045207.

Beaulieu, C., C. Gallagher, R. Killick, R. Lund, and X. Shi (2024): *Is a recent surge in global warming detectable?.* arXiv:2403.03388.

Beckey, J. L., M. Cerezo, A. Sone, and P. J. Coles (2022): *Variational quantum algorithm for estimating the quantum Fisher information*. Phys. Rev. Res. **4**, 013083.

Belghazi, M. I., A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm (2018): *Mutual information neural estimation*. In: Dy, J. and A. Krause (editors), *Proceedings of the 35th Int. Conf. Mach. Learn. – ICML* (PMLR), volume 80 of *Proceedings of Machine Learning Research*, pages 531–540.

Belkin, M., D. Hsu, S. Ma, and S. Mandal (2019): *Reconciling modern machine-learning practice and the classical bias–variance trade-off*. Proc. Natl. Acad. Sci. U.S.A. **116**, 15849.

Bengio, Y. and O. Delalleau (2011): *On the expressive power of deep architectures*. In: Kivinen, J., C. Szepesvári, E. Ukkonen, and T. Zeugmann (editors), *Algorithmic Learning Theory* (Springer), pages 18–36.

Berisha, V. and A. O. Hero (2014): *Empirical non-parametric estimation of the Fisher information*. IEEE Signal Process. Lett. **22**, 988.

Bernien, H., S. Schwartz, A. Keesling, H. Levine, A. Omran, H. Pichler, S. Choi, A. S. Zibrov, M. Endres, M. Greiner, *et al.* (2017): *Probing many-body dynamics on a 51-atom quantum simulator*. Nature **551**, 579.

Bezanson, J., S. Karpinski, V. B. Shah, and A. Edelman (2012): *Julia: A fast dynamic language for technical computing*. arXiv:1209.5145.

Bickel, P. and K. Doksum (2015): *Mathematical statistics: Basic ideas and selected topics* (Chapman and Hall/CRC), 1st edition.

Bickel, S., M. Brückner, and T. Scheffer (2009): *Discriminative learning under covariate shift*. J. Mach. Learn. Res. **10**.

Biderman, S., H. Schoelkopf, Q. G. Anthony, H. Bradley, K. O'Brien, E. Hallahan, M. A. Khan, S. Purohit, U. S. Prashanth, E. Raff, *et al.* (2023): *Pythia: A suite for analyzing large language models across training and scaling*. In: *Int. Conf. Mach. Learn. – ICML* (PMLR), pages 2397–2430.

Blayo, É., M. Bocquet, E. Cosme, and L. F. Cugliandolo (2014): *Advanced data assimilation for geosciences: Lecture notes of the Les Houches School of Physics: Special Issue, June 2012* (Oxford University Press).

Blei, D. M., A. Y. Ng, and M. I. Jordan (2003): *Latent Dirichlet allocation*. J. Mach. Learn. Res. **3**, 993.

Blücher, S., L. Kades, J. M. Pawlowski, N. Strodthoff, and J. M. Urban (2020): *Towards novel insights in lattice field theory with explainable machine learning*. Phys. Rev. D **101**, 094507.

Blumer, A., A. Ehrenfeucht, D. Haussler, and M. K. Warmuth (1989): *Learnability and the Vapnik-Chervonenkis dimension*. J. ACM **36**, 929–965.

Bluvstein, D., S. J. Evered, A. A. Geim, S. H. Li, H. Zhou, T. Manovitz, S. Ebadi, M. Cain, M. Kalinowski, D. Hangleiter, *et al.* (2024): *Logical quantum processor based on reconfigurable atom arrays.* Nature **626**, 58.

Bohrdt, A., C. S. Chiu, G. Ji, M. Xu, D. Greif, M. Greiner, E. Demler, F. Grusdt, and M. Knap (2019): *Classifying snapshots of the doped Hubbard model with machine learning*. Nat. Phys. **15**, 921.

Bohrdt, A., S. Kim, A. Lukin, M. Rispoli, R. Schittko, M. Knap, M. Greiner, and J. Léonard (2021): *Analyzing nonequilibrium quantum states through snapshots with artificial neural networks*. Phys. Rev. Lett. **127**, 150504.

Braunstein, S. L. and C. M. Caves (1994): *Statistical distance and the geometry of quantum states*. Phys. Rev. Lett. **72**, 3439.

Briegel, H. J. and R. Raussendorf (2001): *Persistent entanglement in arrays of interacting particles*. Phys. Rev. Lett. **86**, 910.

Broecker, P., J. Carrasquilla, R. G. Melko, and S. Trebst (2017): *Machine learning quantum phases of matter beyond the fermion sign problem*. Sci. Rep. **7**, 1.

Brown, T., B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.* (2020): *Language models are few-shot learners*. Adv. Neural Inf. Process. Syst. – NeurIPS **33**, 1877.

Bukov, M., M. Schmitt, and M. Dupont (2021): *Learning the ground state of a non-stoquastic quantum Hamiltonian in a rugged neural network landscape*. SciPost Phys. **10**, 147.

Caballero, E., K. Gupta, I. Rish, and D. Krueger (2022): *Broken neural scaling laws.* arXiv:2210.14891.

Caleca, F., S. Tibaldi, and E. Ercolessi (2024): *3-phases confusion learning.* arXiv:2412.02458.

Campos Venuti, L. and P. Zanardi (2007): *Quantum critical scaling of the geometric tensors*. Phys. Rev. Lett. **99**, 095701.

Carleo, G., I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová (2019): *Machine learning and the physical sciences*. Rev. Mod. Phys. **91**, 045002.

Carleo, G. and M. Troyer (2017): *Solving the quantum many-body problem with artificial neural networks*. Science **355**, 602.

Carrasquilla, J. (2020): *Machine learning for quantum matter*. Adv. Phys.: X **5**, 1797528.

Carrasquilla, J. and R. G. Melko (2017): *Machine learning phases of matter*. Nat. Phys. **13**, 431.

Carrasquilla, J., G. Torlai, R. G. Melko, and L. Aolita (2019): *Reconstructing quantum states with generative models*. Nat. Mach. Intell. **1**, 155.

Caruana, R. (1997): *Multitask learning*. Mach. Learn. **28**, 41.

Casella, G. and R. L. Berger (2002): *Statistical inference* (Duxbury).

Casert, C., T. Vieijra, J. Nys, and J. Ryckebusch (2019): *Interpretable machine learning for inferring the phase boundaries in a nonequilibrium system*. Phys. Rev. E **99**, 023304.

Castelnovo, C. and C. Chamon (2007): *Entanglement and topological entropy of the toric code at finite temperature*. Phys. Rev. B **76**, 184442.

Cerezo, M., A. Poremba, L. Cincio, and P. J. Coles (2020): *Variational quantum fidelity estimation*. Quantum **4**, 248.

Cha, P., P. Ginsparg, F. Wu, J. Carrasquilla, P. L. McMahon, and E.-A. Kim (2021): *Attention-based quantum tomography*. Mach. Learn.: Sci. Technol. **3**, 01LT01.

Chaikin, P. M. and T. C. Lubensky (1995): *Principles of condensed matter physics* (Cambridge University Press).

Chartrand, R. (2011): *Numerical differentiation of noisy, nonsmooth data*. Int. Sch. Res. Notices **2011**, 164564.

Chen, A., R. Schwartz-Ziv, K. Cho, M. L. Leavitt, and N. Saphra (2023): *Sudden drops in the loss: Syntax acquisition, phase transitions, and simplicity bias in MLMs*. arXiv:2309.07311.

Chen, R., Z. Song, X. Zhao, and X. Wang (2021): *Variational quantum algorithms for trace distance and fidelity estimation*. Quantum Sci. Technol. **7**, 015019.

Cheng, K. F. and C.-K. Chu (2004): *Semiparametric density estimation under a two-sample density ratio model*. Bernoulli **10**, 583.

Chentsov, N. (1978): *Algebraic foundation of mathematical statistics*. Math. Operationsforsch. statist. **9**, 267.

Ch'ng, K., J. Carrasquilla, R. G. Melko, and E. Khatami (2017): *Machine learning phases of strongly correlated fermions*. Phys. Rev. X **7**, 031038.

Choi, K., M. Liao, and S. Ermon (2021): *Featurized density ratio estimation*. In: de Campos, C. and M. H. Maathuis (editors), *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence* (PMLR), volume 161 of *Proceedings of Machine Learning Research*, pages 172–182.

Chung, S. G. (1999): *Essential finite-size effect in the two-dimensional XY model*. Phys. Rev. B **60**, 11761.

Cohen, M., M. Casebolt, Y. Zhang, K. R. A. Hazzard, and R. Scalettar (2024): *Classical analog of quantum models in synthetic dimensions*. Phys. Rev. A **109**, 013303.

Comparin, T. (2017): *tcompa/BoseHubbardGutzwiller v1.0.2*. Online; last accessed on 06/01/2025.

Cong, I., S. Choi, and M. D. Lukin (2019): *Quantum convolutional neural networks*. Nat. Phys. **15**, 1273.

Conmy, A., A. Mavor-Parker, A. Lynch, S. Heimersheim, and A. Garriga-Alonso (2023): *Towards automated circuit discovery for mechanistic interpretability*. Adv. Neural Inf. Process. Syst. – NeurIPS **36**, 16318.

Corte, I., S. Acevedo, M. Arlego, and C. A. Lamas (2021): *Exploring neural network training strategies to determine phase transitions in frustrated magnetic models*. Comput. Mater. Sci. **198**, 110702.

Cramér, H. (1946): *Mathematical methods of statistics* (Princeton University Press).

Cui, H., F. Behrens, F. Krzakala, and L. Zdeborová (2024): *A phase transition between positional and semantic learning in a solvable model of dot-product attention*. arXiv:2402.03902.

Cybenko, G. (1989): *Approximation by superpositions of a sigmoidal function*. Math. Control Signals Syst. **2**, 303.

Cybiński, K., J. Enouen, A. Georges, and A. Dawid (2024a): *Speak so a physicist can understand you! TetrisCNN for detecting phase transitions and order parameters*. arXiv:2411.02237.

Cybiński, K., M. Płodzień, M. Tomza, M. Lewenstein, A. Dauphin, and A. Dawid (2024b): *Characterizing out-of-distribution generalization of neural networks: Application to the disordered Su-Schrieffer-Heeger model*. arXiv:2406.10012.

Dawid, A., J. Arnold, B. Requena, A. Gresch, M. Płodzień, K. Donatella, K. Nicoli, P. Stornati, R. Koch, M. Büttner, *et al.* (2022): *Modern applications of machine learning in quantum sciences*. arXiv:2204.04198.

Dawid, A., P. Huembeli, M. Tomza, M. Lewenstein, and A. Dauphin (2020): *Phase detection with neural networks: Interpreting the black box*. New J. Phys. **22**, 115001.

Dawid, A., P. Huembeli, M. Tomza, M. Lewenstein, and A. Dauphin (2021): *Hessian-based toolbox for reliable and interpretable machine learning in physics*. Mach. Learn.: Sci. Technol. **3**, 015002.

De Ryck, T., M. De Vos, and A. Bertrand (2021): *Change point detection in time series data using autoencoders with a time-invariant representation*. IEEE Trans. Signal Process. **69**, 3513.

Deldari, S., D. V. Smith, H. Xue, and F. D. Salim (2021): *Time series change point detection with self-supervised contrastive predictive coding*. In: *Proceedings of the Web Conference 2021*. pages 3124–3135.

Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei (2009): *ImageNet: A large-scale hierarchical image database*. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. pages 248–255.

Devroye, L., L. Györfi, and G. Lugosi (1996): *The Bayes error* (Springer), pages 9–20.

Dowty, J. G. (2018): *Chentsov's theorem for exponential families*. Info. Geo. **1**, 117.

Duy, T. T., L. V. Nguyen, V.-D. Nguyen, N. L. Trung, and K. Abed-Meraim (2022): *Fisher information neural estimation*. In: *2022 30th European Signal Processing Conference (EUSIPCO)*. pages 2111–2115.

Dyson, F. J. (1969): *Existence of a phase transition in a one-dimensional Ising ferromagnet*. Commun. Math. Phys. **12**, 91.

Ebadi, S., T. T. Wang, H. Levine, A. Keesling, G. Semeghini, A. Omran, D. Bluvstein, R. Samajdar, H. Pichler, W. W. Ho, *et al.* (2021): *Quantum phases of matter on a 256-atom programmable quantum simulator*. Nature **595**, 227.

Ebrahimzadeh, Z., M. Zheng, S. Karakas, and S. Kleinberg (2019): *Deep learning for multi-scale changepoint detection in multivariate time series.* arXiv:1905.06913.

Fei-Fei, L., R. Fergus, and P. Perona (2004): *Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories*. In: *2004 Conference on Computer Vision and Pattern Recognition Workshop*. pages 178–178.

Fisher, M. P. A., P. B. Weichman, G. Grinstein, and D. S. Fisher (1989): *Boson localization and the superfluid-insulator transition*. Phys. Rev. B **40**, 546.

Fisher, R. A. (1922): *On the mathematical foundations of theoretical statistics*. Philos. Trans. Royal Soc. A **222**, 309.

Fishman, M., S. R. White, and E. M. Stoudenmire (2022): *The ITensor software library for tensor network calculations*. SciPost Phys. Codebases page 4.

Fitzek, D., Y. H. Teoh, H. P. Fung, G. A. Dagnew, E. Merali, M. S. Moss, B. MacLellan, and R. G. Melko (2024): *RydbergGPT.* arXiv:2405.21052.

Flammia, S. T. and R. O'Donnell (2024): *Quantum chi-squared tomography and mutual information testing.* Quantum **8**, 1381.

Franchini, F. (2017): *An introduction to integrable techniques for one-dimensional quantum systems* (Springer).

Friedman, J., T. Hastie, R. Tibshirani, *et al.* (2001): *The elements of statistical learning* (Springer).

Frk, L., P. Baláž, E. Archemashvili, and M. Žonda (2024): *Unsupervised machine learning phase classification for Falicov-Kimball model.* arXiv:2411.07319.

Fujiwara, A. (2022): *Hommage to Chentsov's theorem*. Info. Geo. pages 1–20.

Ganguli, D., D. Hernandez, L. Lovitt, A. Askell, Y. Bai, A. Chen, T. Conerly, N. Dassarma, D. Drain, N. Elhage, *et al.* (2022): *Predictability and surprise in large generative models*. In: *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*. pages 1747–1764.

Garnerone, S., D. Abasto, S. Haas, and P. Zanardi (2009): *Fidelity in topological quantum phases of matter*. Phys. Rev. A **79**, 032302.

Gavreev, M. A., A. S. Mastiukova, E. O. Kiktenko, and A. K. Fedorov (2022): *Learning entanglement breakdown as a phase transition by confusion*. New J. Phys. **24**, 073045.

Ge, J.-C. and M. Tang (2021): *Using machine learning to identify epidemic threshold in complex networks*. In: *2021 2nd International Conference on Artificial Intelligence and Computer Engineering (ICAICE)* (IEEE), pages 333–336.

Gemini Team, G., R. Anil, S. Borgeaud, Y. Wu, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, *et al.* (2023): *Gemini: A family of highly capable multimodal models.* arXiv:2312.11805.

Ghosh, A. and M. Sarkar (2024): *Supervised learning of an interacting two-dimensional hardcore boson model of a weak topological insulator using correlation functions*. Phys. Rev. B **110**, 165134.

Ghosh, S., M. Matty, R. Baumbach, E. D. Bauer, K. A. Modic, A. Shekhter, J. Mydosh, E.-A. Kim, and B. Ramshaw (2020): *One-component order parameter in URu_2Si_2 uncovered by resonant ultrasound spectroscopy and machine learning*. Sci. Adv. **6**, eaaz4074.

Goldenfeld, N. (2018): *Lectures on phase transitions and the renormalization group* (CRC Press).

Goldfeld, Z., D. Patel, S. Sreekumar, and M. M. Wilde (2024): *Quantum neural estimation of entropies*. Phys. Rev. A **109**, 032431.

Gomez, A. M., S. F. Yelin, and K. Najafi (2022): *Reconstructing quantum states using basis-enhanced Born machines.* arXiv:2206.01273.

Gong, M., R. Killick, C. Nemeth, and J. Quinton (2023): *A changepoint approach to modelling non-stationary soil moisture dynamics.* arXiv:2310.17546.

Goodfellow, I. (2016): *NIPS 2016 tutorial: Generative adversarial networks.* arXiv:1701.00160.

Goodfellow, I., Y. Bengio, and A. Courville (2016): *Deep learning* (MIT Press).

Gordon, M. A., K. Duh, and J. Kaplan (2021): *Data and parameter scaling laws for neural machine translation*. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. pages 5915–5922.

Greitemann, J., K. Liu, L. D. C. Jaubert, H. Yan, N. Shannon, and L. Pollet (2019a): *Identification of emergent constraints and hidden order in frustrated magnets using tensorial kernel methods of machine learning*. Phys. Rev. B **100**, 174408.

Greitemann, J., K. Liu, and L. Pollet (2019b): *Probing hidden spin order with interpretable machine learning*. Phys. Rev. B **99**, 060404.

Greplova, E., A. Valenti, G. Boschung, F. Schäfer, N. Lörch, and S. D. Huber (2020): *Unsupervised identification of topological phase transitions using predictive models*. New J. Phys. **22**, 045003.

Griffin, G., A. Holub, and P. Perona (2007): *Caltech-256 object category dataset*. Technical report.

Grundy, T., R. Killick, and G. Mihaylov (2020): *High-dimensional changepoint detection via a geometrically inspired mapping*. Statist. Comput. **30**, 1155.

Gu, S.-J. (2010): *Fidelity approach to quantum phase transitions*. Int. J. Mod. Phys. B **24**, 4371.

Guan, Q. and R. J. Lewis-Swan (2021): *Identifying and harnessing dynamical phase transitions for quantum-enhanced sensing*. Phys. Rev. Res. **3**, 033199.

Guo, W. and L. He (2023): *Learning phase transitions from regression uncertainty: a new regression-based machine learning approach for automated detection of phases of matter*. New J. Phys. **25**, 083037.

Guo, W.-c., B.-q. Ai, and L. He (2023): *Reveal flocking phase transition of self-propelled active particles by machine learning regression uncertainty*. Acta Phys. Sin. **72**, 200701.

Gupta, V. (2015): *Speaker change point detection using deep neural nets*. In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE), pages 4420–4424.

Gur-Ari, G., D. A. Roberts, and E. Dyer (2018): *Gradient descent happens in a tiny subspace*. arXiv:1812.04754.

Gurnee, W. and M. Tegmark (2023): *Language models represent space and time*. arXiv:2310.02207.

Gutmann, M. U. and A. Hyvärinen (2012): *Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics*. J. Mach. Learn. Res. **13**, 307.

Haque, A., S. Chandra, L. Khan, K. Hamlen, and C. Aggarwal (2017): *Efficient multistream classification using direct density ratio estimation*. In: *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*. pages 155–158.

He, K., X. Zhang, S. Ren, and J. Sun (2016): *Deep residual learning for image recognition*. In: *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*

He, Y., K. A. Burghardt, and K. Lerman (2022): *Leveraging change point detection to discover natural experiments in data*. EPJ Data Sci. **11**, 49.

Helstrom, C. W. (1967): *Minimum mean-squared error of estimates in quantum statistics*. Phys. Lett. A **25**, 101.

Hendrycks, D., C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt (2020): *Measuring massive multitask language understanding*. arXiv:2009.03300.

Henighan, T., J. Kaplan, M. Katz, M. Chen, C. Hesse, J. Jackson, H. Jun, T. B. Brown, P. Dhariwal, S. Gray, *et al.* (2020): *Scaling laws for autoregressive generative modeling*. arXiv:2010.14701.

Herrmann, J., S. M. Llima, A. Remm, P. Zapletal, N. A. McMahon, C. Scarato, F. Swiadek, C. K. Andersen, C. Hellings, S. Krinner, *et al.* (2022): *Realizing quantum convolutional neural networks on a superconducting quantum processor to recognize quantum phases*. Nat. Commun. **13**, 4144.

Hestness, J., S. Narang, N. Ardalani, G. Diamos, H. Jun, H. Kianinejad, M. M. A. Patwary, Y. Yang, and Y. Zhou (2017): *Deep learning scaling is predictable, empirically*. arXiv:1712.00409.

Heugel, T. L., M. Biondi, O. Zilberberg, and R. Chitra (2019): *Quantum transducer using a parametric driven-dissipative phase transition*. Phys. Rev. Lett. **123**, 173601.

Hibat-Allah, M., M. Ganahl, L. E. Hayward, R. G. Melko, and J. Carrasquilla (2020): *Recurrent neural network wave functions*. Phys. Rev. Res. **2**, 023358.

Hido, S., T. Idé, H. Kashima, H. Kubo, and H. Matsuzawa (2008): *Unsupervised change analysis using supervised learning*. In: *Advances in Knowledge Discovery and Data Mining: 12th Pacific-Asia Conference, PAKDD 2008 Osaka, Japan, May 20-23, 2008 Proceedings 12* (Springer), pages 148–159.

Ho, C.-T. and D.-W. Wang (2021): *Robust identification of topological phase transition by self-supervised machine learning approach*. New J. Phys. **23**, 083021.

Ho, C.-T. and D.-W. Wang (2023): *Self-supervised ensemble learning: A universal method for phase transition classification of many-body systems*. Phys. Rev. Res. **5**, 043090.

Hoffmann, J., S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. d. L. Casas, L. A. Hendricks, J. Welbl, A. Clark, *et al.* (2022): *Training compute-optimal large language models*. arXiv:2203.15556.

Hornik, K. (1991): *Approximation capabilities of multilayer feedforward networks*. Neural Netw. **4**, 251.

Hsieh, Y.-D., Y.-J. Kao, and A. W. Sandvik (2013): *Finite-size scaling method for the Berezinskii–Kosterlitz–Thouless transition*. J. Stat. Mech. **2013**, P09001.

Hu, X., L. Chu, J. Pei, W. Liu, and J. Bian (2021): *Model complexity of deep learning: A survey*. Knowl. Inf. Syst. **63**, 2585.

Huang, H.-Y., M. Broughton, J. Cotler, S. Chen, J. Li, M. Mohseni, H. Neven, R. Babbush, R. Kueng, J. Preskill, and J. R. McClean (2022a): *Quantum advantage in learning from experiments*. Science **376**, 1182.

Huang, H.-Y., R. Kueng, and J. Preskill (2020): *Predicting many properties of a quantum system from very few measurements*. Nat. Phys. **16**, 1050.

Huang, H.-Y., R. Kueng, G. Torlai, V. V. Albert, and J. Preskill (2022b): *Provably efficient machine learning for quantum many-body problems*. Science **377**, eabk3333.

Huembeli, P., A. Dauphin, and P. Wittek (2018): *Identifying quantum phase transitions with adversarial neural networks*. Phys. Rev. B **97**, 134109.

Huembeli, P., A. Dauphin, P. Wittek, and C. Gogolin (2019): *Automated discovery of characteristic features of phase transitions in many-body localization*. Phys. Rev. B **99**, 104106.

Innes, M. (2018): *Flux: Elegant machine learning with Julia*. J. Open Source Softw. **3**, 602.

Issa, G., O. Bradley, E. Khatami, and R. Scalettar (2025): *Learning by confusion: The phase diagram of the Holstein model*. arXiv:2501.04681.

Itoh, N. and J. Kurths (2010): *Change-point detection of climate time series by non-parametric method*. In: *Proceedings of The World Congress on Engineering and Computer Science*. volume 1, pages 445–448.

Jabari, S., M. Rezaee, F. Fathollahi, and Y. Zhang (2019): *Multispectral change detection using multivariate Kullback-Leibler distance*. ISPRS J. Photogramm. Remote Sens. **147**, 163.

Jacot, A., F. Gabriel, and C. Hongler (2018): *Neural tangent kernel: Convergence and generalization in neural networks*. In: Bengio, S., H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (editors), *Adv. Neural Inf. Process. Syst. – NeurIPS* (Curran Associates, Inc.), volume 31.

Jaksch, D., C. Bruder, J. I. Cirac, C. W. Gardiner, and P. Zoller (1998): *Cold bosonic atoms in optical lattices*. Phys. Rev. Lett. **81**, 3108.

James, G., D. Witten, T. Hastie, and R. Tibshirani (2013): *Statistical learning* (Springer), pages 15–57.

Jarzyna, M. and J. Kołodyński (2020): *Geometric approach to quantum statistical inference*. IEEE J. Sel. Areas Inf. Theory **1**, 367.

Jepsen, P. N., J. Amato-Grill, I. Dimitrova, W. W. Ho, E. Demler, and W. Ketterle (2020): *Spin transport in a tunable Heisenberg model realized with ultracold atoms*. Nature **588**, 403.

Jepsen, P. N., W. W. Ho, J. Amato-Grill, I. Dimitrova, E. Demler, and W. Ketterle (2021): *Transverse spin dynamics in the anisotropic Heisenberg model realized with ultracold atoms*. Phys. Rev. X **11**, 041054.

Jiang, A. Q., A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, *et al.* (2023): *Mistral 7B*. arXiv:2310.06825.

Jozsa, R. (1994): *Fidelity for mixed quantum states*. J. Mod. Opt. **41**, 2315.

Käming, N., A. Dawid, K. Kottmann, M. Lewenstein, K. Sengstock, A. Dauphin, and C. Weitenberg (2021): *Unsupervised machine learning of topological phase transitions from experimental data*. Mach. Learn.: Sci. Technol. **2**, 035037.

Kaplan, J., S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei (2020): *Scaling laws for neural language models*. arXiv:2001.08361.

Kasatkin, V., E. Mozgunov, N. Ezzell, and D. Lidar (2024a): *Detecting quantum and classical phase transitions via unsupervised machine learning of the Fisher information metric*. arXiv:2408.03418.

Kasatkin, V., E. Mozgunov, N. Ezzell, U. Mishra, I. Hen, and D. Lidar (2024b): *ClassiFIM: An unsupervised method to detect phase transitions*. arXiv:2408.03323.

Kashiwa, K., Y. Kikuchi, and A. Tomiya (2019): *Phase transition encoded in neural network*. Prog. Theor. Exp. Phys. **2019**, 083A04.

Kass, M., A. Witkin, and D. Terzopoulos (1988): *Snakes: Active contour models*. Int. J. Comput. Vision **1**, 321.

Katsura, H., D. Schuricht, and M. Takahashi (2015): *Exact ground states and topological order in interacting Kitaev/Majorana chains*. Phys. Rev. B **92**, 115137.

Kawahara, Y. and M. Sugiyama (2009): *Change-point detection in time-series data by direct density-ratio estimation*. In: *Proceedings of the 2009 SIAM international conference on data mining* (SIAM), pages 389–400.

Khan, H., L. Marcuse, and B. Yener (2019): *Deep density ratio estimation for change point detection.* arXiv:1905.09876.

Kharkov, Y. A., V. E. Sotskov, A. A. Karazeev, E. O. Kiktenko, and A. K. Fedorov (2020): *Revealing quantum chaos with machine learning*. Phys. Rev. B **101**, 064406.

Khemani, V., S. P. Lim, D. N. Sheng, and D. A. Huse (2017): *Critical properties of the many-body localization transition*. Phys. Rev. X **7**, 021013.

Kim, H., Y. Zhou, Y. Xu, K. Varma, A. H. Karamlou, I. T. Rosen, J. C. Hoke, C. Wan, J. P. Zhou, W. D. Oliver, *et al.* (2024): *Attention to quantum complexity.* arXiv:2405.11632.

Kim, Y., A. Eddins, S. Anand, K. X. Wei, E. Van Den Berg, S. Rosenblatt, H. Nayfeh, Y. Wu, M. Zaletel, K. Temme, *et al.* (2023): *Evidence for the utility of quantum computing before fault tolerance.* Nature **618**, 500.

Kingma, D. and J. Ba (2014): *Adam: A method for stochastic optimization.* arXiv:1412.6980.

Kitaev, A. Y. (2001): *Unpaired Majorana fermions in quantum wires*. Phys.-Usp. **44**, 131.

Kliesch, M. (2021): *Lecture notes: Quantum characterization, verification, and validation*. Online; last accessed on 21/10/2024.

Kogut, J. B. (1979): *An introduction to lattice gauge theory and spin systems*. Rev. Mod. Phys. **51**, 659.

Kosterlitz, J. (1974): *The critical properties of the two-dimensional XY model*. J. Phys. C: Solid State Phys. **7**, 1046.

Kosterlitz, J. M. and D. J. Thouless (1973): *Ordering, metastability and phase transitions in two-dimensional systems*. J. Phys. C: Solid State Phys. **6**, 1181.

Kottmann, K., P. Corboz, M. Lewenstein, and A. Acín (2021): *Unsupervised mapping of phase diagrams of 2D systems from infinite projected entangled-pair states via deep anomaly detection*. SciPost Phys. **11**, 25.

Kottmann, K., P. Huembeli, M. Lewenstein, and A. Acín (2020): *Unsupervised phase discovery with deep anomaly detection*. Phys. Rev. Lett. **125**, 170603.

Krauth, W., M. Caffarel, and J.-P. Bouchaud (1992): *Gutzwiller wave function for a model of strongly interacting bosons*. Phys. Rev. B **45**, 3137.

Krizhevsky, A. (2009): *Learning multiple layers of features from tiny images*. Master's thesis, University of Toronto.

Krizhevsky, A., I. Sutskever, and G. E. Hinton (2012): *ImageNet classification with deep convolutional neural networks*. In: Pereira, F., C. J. C. Burges, L. Bottou, and K. Q. Weinberger (editors), *Adv. Neural Inf. Process. Syst. – NeurIPS* (Curran Associates, Inc.), volume 25.

Kulkarni, V., R. Al-Rfou, B. Perozzi, and S. Skiena (2015): *Statistically significant detection of linguistic change*. In: *Proceedings of the 24th International Conference on World Wide Web*. pages 625–635.

La Rosa, P. S., A. Nehorai, H. Eswaran, C. L. Lowery, and H. Preissl (2008): *Detection of uterine MMG contractions using a multiple change point estimator and the K-means cluster algorithm*. IEEE Trans. Biomed. Eng. **55**, 453.

Landau, L. D. (1937a): *On the theory of phase transitions. I.* Phys. Z. Sowjet. **11**, 26. Reprinted in Collected Papers of L. D. Landau.

Landau, L. D. (1937b): *On the theory of phase transitions. II.* Phys. Z. Sowjet. **11**, 545. Reprinted in Collected Papers of L. D. Landau.

Lavasani, A., Y. Alavirad, and M. Barkeshli (2021): *Measurement-induced topological entanglement transitions in symmetric random quantum circuits*. Nat. Phys. **17**, 342.

LeCun, Y., F. J. Huang, and L. Bottou (2004): *Learning methods for generic object recognition with invariance to pose and lighting*. In: *2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. volume 2, pages II–104 Vol.2.

LeCun, Y. A., L. Bottou, G. B. Orr, and K.-R. Müller (2012): *Efficient backprop*. In: *Neural Networks: Tricks of the Trade* (Springer), pages 9–48.

Lee, S. S. and B. J. Kim (2019): *Confusion scheme in machine learning detects double phase transitions and quasi-long-range order*. Phys. Rev. E **99**, 043308.

Levi, N., A. Beck, and Y. Bar-Sinai (2023): *Grokking in linear estimators – A solvable model that groks without understanding.* arXiv:2310.16441.

Liese, F. and I. Vajda (2006): *On divergences and informations in statistics and information theory*. IEEE Trans. Inf. Theory **52**, 4394.

Linardatos, P., V. Papastefanopoulos, and S. Kotsiantis (2021): *Explainable AI: A review of machine learning interpretability methods*. Entropy **23**, 18.

Liu, J., H.-N. Xiong, F. Song, and X. Wang (2014): *Fidelity susceptibility and quantum Fisher information for density operators with arbitrary ranks*. Physica A **410**, 167.

Liu, K., J. Greitemann, and L. Pollet (2019): *Learning multiple order parameters with interpretable machines*. Phys. Rev. B **99**, 104410.

Liu, L. Z., Y. Wang, J. Kasai, H. Hajishirzi, and N. A. Smith (2021): *Probing across time: What does RoBERTa know and when?* arXiv:2104.07885.

Liu, Q., L. Li, Z. Tang, and D. Zhou (2018): *Breaking the curse of horizon: Infinite-horizon off-policy estimation.* Adv. Neural Inf. Process. Syst. – NeurIPS **31**.

Liu, S., M. Yamada, N. Collier, and M. Sugiyama (2013): *Change-point detection in time-series data by relative density-ratio estimation*. Neural Netw. **43**, 72.

Liu, Y.-H. and E. P. L. van Nieuwenburg (2018): *Discriminative cooperative networks for detecting phase transitions*. Phys. Rev. Lett. **120**, 176401.

Liu, Y.-J., A. Smith, M. Knap, and F. Pollmann (2023): *Model-independent learning of quantum phases of matter with quantum convolutional neural networks*. Phys. Rev. Lett. **130**, 220603.

Liu, Z., O. Kitouni, N. S. Nolte, E. Michaud, M. Tegmark, and M. Williams (2022a): *Towards understanding grokking: An effective theory of representation learning*. In: Koyejo, S., S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (editors), *Adv. Neural Inf. Process. Syst. – NeurIPS* (Curran Associates, Inc.), volume 35, pages 34651–34663.

Liu, Z., E. J. Michaud, and M. Tegmark (2022b): *Omnigrok: Grokking beyond algorithmic data*. In: *The Eleventh International Conference on Learning Representations – ICLR*.

Liu, Z., W. Ping, R. Roy, P. Xu, C. Lee, M. Shoeybi, and B. Catanzaro (2024): *ChatQA: Surpassing GPT-4 on conversational QA and RAG*. arXiv:2401.10225.

Lu, S., S. Huang, K. Li, J. Li, J. Chen, D. Lu, Z. Ji, Y. Shen, D. Zhou, and B. Zeng (2018): *Separability-entanglement classifier via machine learning*. Phys. Rev. A **98**, 012315.

Lu, Z., H. Pu, F. Wang, Z. Hu, and L. Wang (2017): *The expressive power of neural networks: A view from the width*. In: Guyon, I., U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (editors), *Adv. Neural Inf. Process. Syst. – NeurIPS* (Curran Associates, Inc.), volume 30.

Lukin, A., M. Rispoli, R. Schittko, M. E. Tai, A. M. Kaufman, S. Choi, V. Khemani, J. Léonard, and M. Greiner (2019): *Probing entanglement in a many-body–localized system*. Science **364**, 256.

Luo, D., Z. Chen, J. Carrasquilla, and B. K. Clark (2022): *Autoregressive neural network for simulating open quantum systems via a probabilistic formulation*. Phys. Rev. Lett. **128**, 090501.

Macieszczak, K., M. Guţă, I. Lesanovsky, and J. P. Garrahan (2016): *Dynamical phase transitions as a resource for quantum enhanced metrology*. Phys. Rev. A **93**, 022103.

MacLellan, B., P. Roztocki, S. Czischek, and R. G. Melko (2024): *End-to-end variational quantum sensing*. arXiv:2403.02394.

Malladi, R., G. P. Kalamangalam, and B. Aazhang (2013): *Online Bayesian change point detection algorithms for segmentation of epileptic activity*. In: *2013 Asilomar conference on signals, systems and computers* (IEEE), pages 1833–1837.

Martínez-Herrera, J., O. A. Rodríguez-López, and M. Solís (2022): *Critical temperature of one-dimensional Ising model with long-range interaction revisited*. Phys. A: Stat. Mech. Appl. **596**, 127136.

Maskara, N., M. Buchhold, M. Endres, and E. van Nieuwenburg (2022): *Learning algorithm reflecting universal scaling behavior near phase transitions*. Phys. Rev. Res. **4**, L022032.

McClean, J. R., S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven (2018): *Barren plateaus in quantum neural network training landscapes*. Nat. Commun. **9**, 1.

McGrath, T., A. Kapishnikov, N. Tomašev, A. Pearce, M. Wattenberg, D. Hassabis, B. Kim, U. Paquet, and V. Kramnik (2022): *Acquisition of chess knowledge in AlphaZero*. Proc. Natl. Acad. Sci. U.S.A. **119**, e2206625119.

Melko, R. G. and J. Carrasquilla (2024): *Language models for quantum simulation*. Nat. Comput. Sci **4**, 1.

Menon, A. and C. S. Ong (2016): *Linking losses for density ratio and class-probability estimation*. In: Balcan, M. F. and K. Q. Weinberger (editors), *Proceedings of The 33rd International Conference on Machine Learning* (PMLR, New York, New York, USA), volume 48 of *Proceedings of Machine Learning Research*, pages 304–313.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller (1953): *Equation of state calculations by fast computing machines*. J. Chem. Phys. **21**, 1087.

Michaud, E., Z. Liu, U. Girit, and M. Tegmark (2023): *The quantization model of neural scaling*. In: Oh, A., T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (editors), *Adv. Neural Inf. Process. Syst. – NeurIPS* (Curran Associates, Inc.), volume 36, pages 28699–28722.

Miles, C., A. Bohrdt, R. Wu, C. Chiu, M. Xu, G. Ji, M. Greiner, K. Q. Weinberger, E. Demler, and E.-A. Kim (2021): *Correlator convolutional neural networks as an interpretable architecture for image-like quantum matter data*. Nat. Commun. **12**, 1.

Miles, C., R. Samajdar, S. Ebadi, T. T. Wang, H. Pichler, S. Sachdev, M. D. Lukin, M. Greiner, K. Q. Weinberger, and E.-A. Kim (2023): *Machine learning discovery of new phases in programmable quantum simulator snapshots*. Phys. Rev. Res. **5**, 013026.

Millidge, B. (2023): *Basic facts about language models during training*. Online; last accessed on 12/04/2024.

Minnhagen, P. and B. J. Kim (2003): *Direct evidence of the discontinuous character of the Kosterlitz-Thouless jump*. Phys. Rev. B **67**, 172509.

Molnar, C. (2022): *Interpretable machine learning*. 2nd edition.

Mora, T. and W. Bialek (2011): *Are biological systems poised at criticality?* J. Stat. Phys. **144**, 268.

Moustakides, G. V. and K. Basioti (2019): *Training neural networks for likelihood / density ratio estimation*. arXiv:1911.00405.

Muñoz-Gil, G., G. Volpe, M. A. Garcia-March, E. Aghion, A. Argun, C. B. Hong, T. Bland, S. Bo, J. A. Conejero, N. Firbas, *et al.* (2021): *Objective comparison of methods to decode anomalous diffusion*. Nat. Commun. **12**, 6253.

Nakaishi, K., Y. Nishikawa, and K. Hukushima (2024): *Critical phase transition in a large language model*. arXiv:2406.05335.

Nakkiran, P., G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever (2021): *Deep double descent: Where bigger models and more data hurt*. J. Stat. Mech.: Theory Exp **2021**, 124003.

Nanda, N., L. Chan, T. Lieberum, J. Smith, and J. Steinhardt (2023): *Progress measures for grokking via mechanistic interpretability*. arXiv:2301.05217.

Ng, A. and M. Jordan (2001): *On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes*. In: Dietterich, T., S. Becker, and Z. Ghahramani (editors), *Adv. Neural Inf. Process. Syst. – NeurIPS* (MIT Press), volume 14.

Nguyen, H. C., R. Zecchina, and J. Berg (2017): *Inverse statistical problems: From the inverse Ising problem to data science*. Adv. Phys. **66**, 197.

Nguyen, X., M. J. Wainwright, and M. Jordan (2007): *Estimating divergence functionals and the likelihood ratio by penalized convex risk minimization*. In: Platt, J., D. Koller, Y. Singer, and S. Roweis (editors), *Adv. Neural Inf. Process. Syst. – NeurIPS* (Curran Associates, Inc.), volume 20.

Ni, Q., J. Kang, M. Tang, Y. Liu, and Y. Zou (2019a): *Learning epidemic threshold in complex networks by convolutional neural network*. Chaos **29**.

Ni, Q., M. Tang, Y. Liu, and Y.-C. Lai (2019b): *Machine learning dynamical phase transitions in complex networks*. Phys. Rev. E **100**, 052312.

Nielsen, M. A. and I. L. Chuang (2010): *Quantum computation and quantum information: 10th anniversary edition* (Cambridge University Press).

Noel, C., P. Niroula, D. Zhu, A. Risinger, L. Egan, D. Biswas, M. Cetina, A. V. Gorshkov, M. J. Gullans, D. A. Huse, *et al.* (2022): *Measurement-induced quantum phases realized in a trapped-ion quantum computer*. Nat. Phys. pages 1–5.

Nowozin, S., B. Cseke, and R. Tomioka (2016): *f-GAN: Training generative neural samplers using variational divergence minimization*. Adv. Neural Inf. Process. Syst. – NeurIPS **29**.

Ohtsuki, T. and T. Ohtsuki (2017): *Deep learning the quantum phase transitions in random electron systems: Applications to three dimensions*. J. Phys. Soc. Jpn. **86**, 044708.

Olah, C. (2022): *Mechanistic interpretability, variables, and the importance of interpretable bases*. Online; last accessed on 15/04/2024.

Olsson, C., N. Elhage, N. Nanda, N. Joseph, N. DasSarma, T. Henighan, B. Mann, A. Askell, Y. Bai, A. Chen, *et al.* (2022): *In-context learning and induction heads*. arXiv:2209.11895.

Onsager, L. (1944): *Crystal statistics. I. A two-dimensional model with an order-disorder transition*. Phys. Rev. **65**, 117.

Pal, A. and D. A. Huse (2010): *Many-body localization phase transition*. Phys. Rev. B **82**, 174411.

Pan, A., K. Bhatia, and J. Steinhardt (2022): *The effects of reward misspecification: Mapping and mitigating misaligned models*. arXiv:2201.03544.

Paszke, A., S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, *et al.* (2019): *PyTorch: An imperative style, high-performance deep learning library*. In: Wallach, H., H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (editors), *Adv. Neural Inf. Process. Syst. – NeurIPS* (Curran Associates, Inc.), volume 32.

Patel, Z., E. Merali, and S. J. Wetzel (2022): *Unsupervised learning of Rydberg atom array phase diagram with Siamese neural networks*. New J. Phys. **24**, 113021.

Pepelyshev, A. and A. S. Polunchenko (2015): *Real-time financial surveillance via quickest change-point detection methods*. arXiv:1509.01570.

Pilati, S. and P. Pieri (2019): *Supervised machine learning of ultracold atoms with speckle disorder*. Sci. Rep. **9**, 1.

Ponte, P. and R. G. Melko (2017): *Kernel methods for interpretable machine learning of order parameters*. Phys. Rev. B **96**, 205146.

Poole, B., S. Lahiri, M. Raghu, J. Sohl-Dickstein, and S. Ganguli (2016): *Exponential expressivity in deep neural networks through transient chaos*. Adv. Neural Inf. Process. Syst. – NeurIPS **29**.

Power, A., Y. Burda, H. Edwards, I. Babuschkin, and V. Misra (2022): *Grokking: Generalization beyond overfitting on small algorithmic datasets*. arXiv:2201.02177.

Prokopenko, M., J. T. Lizier, O. Obst, and X. R. Wang (2011): *Relating Fisher information to order parameters*. Phys. Rev. E **84**, 041116.

Qin, J. (1998): *Inferences for case-control and semiparametric two-sample density ratio models*. Biometrika **85**, 619.

Radford, A., J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever (2021): *Learning transferable visual models from natural language supervision*. In: Meila, M. and T. Zhang (editors), *Proceedings of the 38th International Conference on Machine Learning* (PMLR), volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763.

Radford, A., J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.* (2019): *Language models are unsupervised multitask learners*. OpenAI blog **1**, 9.

Rae, J. W., S. Borgeaud, T. Cai, K. Millican, J. Hoffmann, F. Song, J. Aslanides, S. Henderson, R. Ring, S. Young, *et al.* (2021): *Scaling language models: Methods, analysis & insights from training gopher*. arXiv:2112.11446.

Raghu, M., B. Poole, J. Kleinberg, S. Ganguli, and J. Sohl-Dickstein (2017): *On the expressive power of deep neural networks*. In: Precup, D. and Y. W. Teh (editors), *Proceedings of the 34th International Conference on Machine Learning* (PMLR), volume 70 of *PMLR*, pages 2847–2854.

Rao, C. R. (1945): *Information and the accuracy attainable in the estimation of statistical parameters*. Bull. Calcutta Math. Soc. **37**, 81.

Räuker, T., A. Ho, S. Casper, and D. Hadfield-Menell (2023): *Toward transparent AI: A survey on interpreting the inner structures of deep neural networks*. In: *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)* (IEEE), pages 464–483.

Raventós, A., M. Paul, F. Chen, and S. Ganguli (2024): *Pretraining task diversity and the emergence of non-Bayesian in-context learning for regression*. Adv. Neural Inf. Process. Syst. – NeurIPS **36**.

Razeghi, Y., R. L. Logan IV, M. Gardner, and S. Singh (2022): *Impact of pretraining term frequencies on few-shot reasoning.* arXiv:2202.07206.

Reeves, J., J. Chen, X. L. Wang, R. Lund, and Q. Q. Lu (2007): *A review and comparison of changepoint detection techniques for climate data*. J. Appl. Meteorol. Climatol. **46**, 900.

Reh, M., M. Schmitt, and M. Gärttner (2021): *Time-dependent variational principle for open quantum systems with artificial neural networks*. Phys. Rev. Lett. **127**, 230501.

Rem, B. S., N. Käming, M. Tarnowski, L. Asteria, N. Fläschner, C. Becker, K. Sengstock, and C. Weitenberg (2019): *Identifying quantum phase transitions using artificial neural networks on experimental data*. Nat. Phys. **15**, 917.

Rende, R., S. Goldt, F. Becca, and L. L. Viteritti (2024): *Fine-tuning neural network quantum states*. Phys. Rev. Res. **6**, 043280.

Rhodes, B., K. Xu, and M. U. Gutmann (2020): *Telescoping density-ratio estimation*. In: Larochelle, H., M. Ranzato, R. Hadsell, M. Balcan, and H. Lin (editors), *Adv. Neural Inf. Process. Syst. – NeurIPS* (Curran Associates, Inc.), volume 33, pages 4905–4916.

Richter-Laskowska, M., M. Kurpas, and M. M. Maśka (2023): *Learning by confusion approach to identification of discontinuous phase transitions*. Phys. Rev. E **108**, 024113.

Rispoli, M., A. Lukin, R. Schittko, S. Kim, M. E. Tai, J. Léonard, and M. Greiner (2019): *Quantum critical behaviour at the many-body localization transition*. Nature **573**, 385.

Rodriguez-Nieva, J. F. and M. S. Scheurer (2019): *Identifying topological order through unsupervised machine learning*. Nat. Phys. **15**, 790.

Rombach, R., A. Blattmann, D. Lorenz, P. Esser, and B. Ommer (2021): *High-resolution image synthesis with latent diffusion models.* arXiv:2112.10752.

Rosenfeld, J. S., A. Rosenfeld, Y. Belinkov, and N. Shavit (2019): *A constructive prediction of the generalization error across scales.* arXiv:1909.12673.

Rota, R., F. Storme, N. Bartolo, R. Fazio, and C. Ciuti (2017): *Critical behavior of dissipative two-dimensional spin lattices*. Phys. Rev. B **95**, 134431.

Rubin, N., I. Seroussi, and Z. Ringel (2023): *Droplets of good representations: Grokking as a first order phase transition in two layer networks.* arXiv:2310.03789.

Ruder, S. (2017): *An overview of multi-task learning in deep neural networks.* arXiv:1706.05098.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986): *Learning representations by back-propagating errors*. Nature **323**, 533.

Russakovsky, O., J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.* (2015): *ImageNet large scale visual recognition challenge*. Int. J. Comput. Vis. **115**, 211.

Rybach, D., C. Gollan, R. Schluter, and H. Ney (2009): *Audio segmentation for speech recognition using segment features*. In: *2009 IEEE International Conference on Acoustics, Speech and Signal Processing* (IEEE), pages 4197–4200.

Sachdev, S. (2011): *Quantum phase transitions* (Cambridge University Press).

Sadoune, N., G. Giudici, K. Liu, and L. Pollet (2023): *Unsupervised interpretable learning of phases from many-qubit systems*. Phys. Rev. Res. **5**, 013082.

Saitta, L., A. Giordana, and A. Cornuejols (2011): *Phase transitions in machine learning* (Cambridge University Press).

Salton, G. and C. Buckley (1988): *Term-weighting approaches in automatic text retrieval*. Inf. Process. Manag. **24**, 513.

Satzinger, K., Y.-J. Liu, A. Smith, C. Knapp, M. Newman, C. Jones, Z. Chen, C. Quintana, X. Mi, A. Dunsworth, *et al.* (2021): *Realizing topologically ordered states on a quantum processor*. Science **374**, 1237.

Savitzky, A. and M. J. Golay (1964): *Smoothing and differentiation of data by simplified least squares procedures.* Anal. Chem. **36**, 1627.

Schaeffer, R., B. Miranda, and S. Koyejo (2023): *Are emergent abilities of large language models a mirage?* In: Oh, A., T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (editors), *Adv. Neural Inf. Process. Syst. – NeurIPS* (Curran Associates, Inc.), volume 36, pages 55565–55581.

Schäfer, F. and N. Lörch (2019): *Vector field divergence of predictive model output as indication of phase transitions*. Phys. Rev. E **99**, 062107.

Scheurer, M. S. and R.-J. Slager (2020): *Unsupervised machine learning and band topology*. Phys. Rev. Lett. **124**, 226401.

Schindler, F., N. Regnault, and T. Neupert (2017): *Probing many-body localization with neural networks*. Phys. Rev. B **95**, 245134.

Schlömer, H. and A. Bohrdt (2023): *Fluctuation based interpretable analysis scheme for quantum many-body snapshots*. SciPost Phys. **15**, 099.

Scholl, P., M. Schuler, H. J. Williams, A. A. Eberharter, D. Barredo, K.-N. Schymik, V. Lienhard, L.-P. Henry, T. C. Lang, T. Lahaye, *et al.* (2021): *Quantum simulation of 2D antiferromagnets with hundreds of Rydberg atoms*. Nature **595**, 233.

Schollwöck, U., J. Richter, D. J. Farnell, and R. F. Bishop (2008): *Quantum magnetism*, volume 645 (Springer).

Schollwöck, U. (2011): *The density-matrix renormalization group in the age of matrix product states*. Ann. Phys. **326**, 96.

Schuhmann, C., R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, *et al.* (2022): *LAION-5B: An open large-scale dataset for training next generation image-text models*. Adv. Neural Inf. Process. Syst. – NeurIPS **35**, 25278.

Seif, A., M. Hafezi, and C. Jarzynski (2021): *Machine learning the thermodynamic arrow of time*. Nat. Phys. **17**, 105.

Semeghini, G., H. Levine, A. Keesling, S. Ebadi, T. T. Wang, D. Bluvstein, R. Verresen, H. Pichler, M. Kalinowski, R. Samajdar, *et al.* (2021): *Probing topological spin liquids on a programmable quantum simulator*. Science **374**, 1242.

Sethna, J. P. (2023): *Statistical mechanics: Entropy, order parameters, and complexity* (Oxford University Press).

Sharir, O., Y. Levine, N. Wies, G. Carleo, and A. Shashua (2020): *Deep autoregressive models for the efficient variational simulation of many-body quantum systems*. Phys. Rev. Lett. **124**, 020503.

Shi, X., C. Gallagher, R. Lund, and R. Killick (2022): *A comparison of single and multiple changepoint techniques for time series data*. Comput. Stat. Data Anal. **170**, 107433.

Shin, M., J. Lee, and K. Jeong (2024): *Estimating quantum mutual information through a quantum neural network*. Quantum Inf. Process. **23**, 57.

Shwartz-Ziv, R. and N. Tishby (2017): *Opening the black box of deep neural networks via information*. arXiv:1703.00810.

Siegler, M. A., U. Jain, B. Raj, and R. M. Stern (1997): *Automatic segmentation, classification and clustering of broadcast news audio*. In: *Proc. DARPA speech recognition workshop*. volume 1997.

Silver, D., T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, *et al.* (2018): *A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play*. Science **362**, 1140.

Simon, J., W. S. Bakr, R. Ma, M. E. Tai, P. M. Preiss, and M. Greiner (2011): *Quantum simulation of antiferromagnetic spin chains in an optical lattice*. Nature **472**, 307.

Simon, J. B., M. Knutins, L. Ziyin, D. Geisz, A. J. Fetterman, and J. Albrecht (2023): *On the stepwise nature of self-supervised learning*. In: Krause, A., E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett (editors), *Proceedings of the 40th International Conference on Machine Learning* (PMLR), volume 202 of *Proceedings of Machine Learning Research*, pages 31852–31876.

Singh, J., M. Scheurer, and V. Arora (2021): *Conditional generative models for sampling and phase transition indication in spin systems*. SciPost Phys. **11**, 043.

Smacchia, P., L. Amico, P. Facchi, R. Fazio, G. Florio, S. Pascazio, and V. Vedral (2011): *Statistical mechanics of the cluster Ising model*. Phys. Rev. A **84**, 022304.

Smith, A., B. Jobst, A. G. Green, and F. Pollmann (2022): *Crossing a topological phase transition with a quantum computer*. Phys. Rev. Res. **4**, L022020.

Smith, A., M. Kim, F. Pollmann, and J. Knolle (2019): *Simulating quantum many-body dynamics on a current digital quantum computer*. Npj Quantum Inf. **5**, 1.

Spanhol, F. A., L. S. Oliveira, C. Petitjean, and L. Heutte (2016): *A Dataset for breast cancer histopathological image classification*. IEEE. Trans. Biomed. **63**, 1455.

Srivastava, A., A. Rastogi, A. Rao, A. A. M. Shoeb, A. Abid, A. Fisch, A. R. Brown, A. Santoro, A. Gupta, A. Garriga-Alonso, *et al.* (2022): *Beyond the imitation game: Quantifying and extrapolating the capabilities of language models*. arXiv:2206.04615.

Stephens, G. J., T. Mora, G. c. v. Tkačik, and W. Bialek (2013): *Statistical thermodynamics of natural images*. Phys. Rev. Lett. **110**, 018701.

Suchsland, P. and S. Wessel (2018): *Parameter diagnostics of phases and phase transition learning by neural networks*. Phys. Rev. B **97**, 174435.

Sugiyama, M., T. Suzuki, and T. Kanamori (2012): *Density ratio estimation in machine learning* (Cambridge University Press).

Sugiyama, M., T. Suzuki, S. Nakajima, H. Kashima, P. Von Bünau, and M. Kawanabe (2008): *Direct importance estimation for covariate shift adaptation*. Ann. Inst. Stat. Math. **60**, 699.

Sun, X., H. Yang, N. Wu, T. C. Scott, J. Zhang, and W. Zhang (2023): *Snake net with a neural network for detecting multiple phases in the phase diagram*. Phys. Rev. E **107**, 065303.

Szołdra, T., P. Sierant, K. Kottmann, M. Lewenstein, and J. Zakrzewski (2021): *Detecting ergodic bubbles at the crossover to many-body localization using neural networks*. Phys. Rev. B **104**, L140202.

Taillefer, L. (2010): *Scattering and pairing in cuprate superconductors*. Annu. Rev. Condens. Matter Phys. **1**, 51.

Tamai, K., T. Okubo, T. V. T. Duy, N. Natori, and S. Todo (2023): *Absorbing phase transitions in artificial deep neural networks.* arXiv:2307.02284.

Tan, K. C. and T. Volkoff (2021): *Variational quantum algorithms to estimate rank, quantum entropies, fidelity, and Fisher information via purity minimization*. Phys. Rev. Res. **3**, 033251.

Tao, P., C. Du, Y. Xiao, and C. Zeng (2023): *Data-driven detection of critical points of phase transitions in complex systems*. Commun. Phys. **6**, 311.

Tevissen, Y., J. Boudy, G. Chollet, and F. Petitpont (2023): *Zero-shot speaker change point detection using large language models*. In: *Journée des doctorants Paris Saclay*.

Thilak, V., E. Littwin, S. Zhai, O. Saremi, R. Paiss, and J. Susskind (2022): *The slingshot mechanism: An empirical study of adaptive optimizers and the grokking phenomenon.* arXiv:2206.04817.

Tibaldi, S., G. Magnifico, D. Vodola, and E. Ercolessi (2023): *Unsupervised and supervised learning of interacting topological phases from single-particle correlation functions*. SciPost Phys. **14**, 005.

Torlai, G., B. Timar, E. P. L. van Nieuwenburg, H. Levine, A. Omran, A. Keesling, H. Bernien, M. Greiner, V. Vuletić, M. D. Lukin, R. G. Melko, and M. Endres (2019): *Integrating neural networks with a quantum simulator for state reconstruction*. Phys. Rev. Lett. **123**, 230504.

Truong, C., L. Oudre, and N. Vayatis (2020): *Selective review of offline change point detection methods*. Signal Process. **167**, 107299.

Valenti, A., E. Greplova, N. H. Lindner, and S. D. Huber (2022): *Correlation-enhanced neural networks as interpretable variational quantum states*. Phys. Rev. Res. **4**, L012010.

Van den Burg, G. J. and C. K. Williams (2020): *An evaluation of change point detection algorithms.* arXiv:2003.06222.

Van Den Oord, A., N. Kalchbrenner, and K. Kavukcuoglu (2016): *Pixel recurrent neural networks.* In: *Int. Conf. Mach. Learn. – ICML* (PMLR), pages 1747–1756.

Van Himbergen, J. E. and S. Chakravarty (1981): *Helicity modulus and specific heat of classical* XY *model in two dimensions*. Phys. Rev. B **23**, 359.

van Nieuwenburg, E., E. Bairey, and G. Refael (2018): *Learning phase transitions from dynamics*. Phys. Rev. B **98**, 060301.

Van Nieuwenburg, E. P., Y.-H. Liu, and S. D. Huber (2017): *Learning phase transitions by confusion*. Nat. Phys. **13**, 435.

Vapnik, V. N. (1998): *Statistical learning theory* (Wiley-Interscience).

Vapnik, V. N. (1999): *The nature of statistical learning theory* (Springer), 2nd edition.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin (2017): *Attention is all you need*. In: Guyon, I., U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (editors), *Adv. Neural Inf. Process. Syst. – NeurIPS* (Curran Associates, Inc.), volume 30.

Venderley, J., V. Khemani, and E.-A. Kim (2018): *Machine learning out-of-equilibrium phases of matter*. Phys. Rev. Lett. **120**, 257204.

Verresen, R., R. Moessner, and F. Pollmann (2017): *One-dimensional symmetry protected topological phases and their transitions*. Phys. Rev. B **96**, 165124.

Vicsek, T., A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet (1995): *Novel type of phase transition in a system of self-driven particles*. Phys. Rev. Lett. **75**, 1226.

Vieijra, T., C. Casert, J. Nys, W. De Neve, J. Haegeman, J. Ryckebusch, and F. Verstraete (2020): *Restricted Boltzmann machines for quantum states with non-abelian or anyonic symmetries*. Phys. Rev. Lett. **124**, 097201.

Wang, B., M. Feng, and Z.-Q. Chen (2010): *Berezinskii-Kosterlitz-Thouless transition uncovered by the fidelity susceptibility in the XXZ model*. Phys. Rev. A **81**, 064301.

Wang, H., M. Weber, J. Izaac, and C. Y.-Y. Lin (2022): *Predicting properties of quantum systems with conditional generative models.* arXiv:2211.16943.

Wang, L. (2016): *Discovering phase transitions with unsupervised learning*. Phys. Rev. B **94**, 195105.

Wang, X., R. A. Borsoi, C. Richard, and J. Chen (2023): *Change point detection with neural online density-ratio estimator*. In: *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE), pages 1–5.

Wegner, F. J. (1971): *Duality in generalized Ising models and phase transitions without local order parameters*. J. Math. Phys. **12**, 2259.

Wei, J., Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, *et al.* (2022): *Emergent abilities of large language models.* arXiv:2206.07682.

Weinberg, P. and M. Bukov (2017): *QuSpin: a Python package for dynamics and exact diagonalisation of quantum many body systems part I: Spin chains*. SciPost Phys. **2**, 003.

Weinberg, P. and M. Bukov (2019): *QuSpin: a Python package for dynamics and exact diagonalisation of quantum many body systems. Part II: Bosons, fermions and higher spins*. SciPost Phys. **7**, 20.

Wen, X.-G. (1990): *Topological orders in rigid states*. Int. J. Mod. Phys. B **4**, 239.

Wetzel, S. J. (2017): *Unsupervised learning of phase transitions: From principal component analysis to variational autoencoders*. Phys. Rev. E **96**, 022140.

Wetzel, S. J. and M. Scherzer (2017): *Machine learning of explicit order parameters: From the Ising model to SU(2) lattice gauge theory*. Phys. Rev. B **96**, 184410.

Wilczek, F. (2009): *Majorana returns*. Nat. Phys. **5**, 614.

Wu, D., L. Wang, and P. Zhang (2019): *Solving statistical mechanics using variational autoregressive networks*. Phys. Rev. Lett. **122**, 080602.

Yang, M.-F. (2007): *Ground-state fidelity in one-dimensional gapless models*. Phys. Rev. B **76**, 180403.

Yang, S., S.-J. Gu, C.-P. Sun, and H.-Q. Lin (2008): *Fidelity susceptibility and long-range correlation in the Kitaev honeycomb model*. Phys. Rev. A **78**, 012304.

You, W.-L., Y.-W. Li, and S.-J. Gu (2007): *Fidelity, dynamic structure factor, and susceptibility in critical phenomena*. Phys. Rev. E **76**, 022101.

Yu, Y., L.-W. Yu, W. Zhang, H. Zhang, X. Ouyang, Y. Liu, D.-L. Deng, and L.-M. Duan (2022): *Experimental unsupervised learning of non-Hermitian knotted phases with solid-state spins*. Npj Quantum Inf. **8**, 116.

Zanardi, P., P. Giorda, and M. Cozzini (2007): *Information-theoretic differential geometry of quantum phase transitions*. Phys. Rev. Lett. **99**, 100603.

Zapletal, P., N. A. McMahon, and M. J. Hartmann (2024): *Error-tolerant quantum convolutional neural networks for symmetry-protected topological phases*. Phys. Rev. Res. **6**, 033111.

Zhai, X., A. Kolesnikov, N. Houlsby, and L. Beyer (2022): *Scaling vision transformers*. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pages 12104–12113.

Zhang, D., F. Schäfer, and J. Arnold (2024a): *Machine learning the Ising transition transitions: A comparison between discriminative and generative approaches*. arXiv:2411.19370.

Zhang, K., S. Feng, Y. D. Lensky, N. Trivedi, and E.-A. Kim (2024b): *Machine learning reveals features of spinon Fermi surface*. Commun. Phys. **7**, 54.

Zhang, P., H. Shen, and H. Zhai (2018): *Machine learning topological invariants with neural networks*. Phys. Rev. Lett. **120**, 066401.

Zhang, W., L. Wang, and Z. Wang (2019a): *Interpretable machine learning study of the many-body localization transition in disordered quantum Ising spin chains*. Phys. Rev. B **99**, 054208.

Zhang, Y., P. Ginsparg, and E.-A. Kim (2020): *Interpreting machine learning of topological quantum phase transitions*. Phys. Rev. Res. **2**, 023283.

Zhang, Y. and E.-A. Kim (2017): *Quantum loop topography for machine learning*. Phys. Rev. Lett. **118**, 216401.

Zhang, Y., A. Mesaros, K. Fujita, S. Edkins, M. Hamidian, K. Ch'ng, H. Eisaki, S. Uchida, J. S. Davis, E. Khatami, *et al.* (2019b): *Machine learning in electronic-quantum-matter imaging experiments*. Nature **570**, 484.

Zhao, W., S. Guo, K. Lerman, and Y.-Y. Ahn (2024): *Discovering collective narratives shifts in online discussions*. In: *Proceedings of the International AAAI Conference on Web and Social Media*. volume 18, pages 1804–1817.

Zhong, Z., Z. Liu, M. Tegmark, and J. Andreas (2023): *The clock and the pizza: Two stories in mechanistic explanation of neural networks*. In: Oh, A., T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (editors), *Adv. Neural Inf. Process. Syst. – NeurIPS* (Curran Associates, Inc.), volume 36, pages 27223–27250.

Zhou, D.-X. (2020): *Universality of deep convolutional neural networks*. Appl. Comput. Harmon. Anal. **48**, 787.

Zhou, L., J. Kong, Z. Lan, and W. Zhang (2023): *Dynamical quantum phase transitions in a spinor Bose-Einstein condensate and criticality enhanced quantum sensing*. Phys. Rev. Res. **5**, 013087.

Ziyin, L. and M. Ueda (2022): *Exact phase transitions in deep learning*. arXiv:2205.12510.

Zvyagintseva, D., H. Sigurdsson, V. Kozin, I. Iorsh, I. Shelykh, V. Ulyantsev, and O. Kyriienko (2022): *Machine learning of phase transitions in nonlinear polariton lattices*. Commun. Phys. **5**, 8.

Zwerger, W. (2003): *Mott–Hubbard transition of cold atoms in optical lattices*. J. Opt. B: Quantum Semiclass. Opt. **5**, S9.