

Quantum field-theoretic machine learning and the renormalization group

Dimitrios S. Bachtis

Submitted to Swansea University
in fulfilment of the requirements for the degree of

Doctor of Philosophy
in
Mathematics



Swansea University
Prifysgol Abertawe

May 2022

Thesis title: *Quantum field-theoretic machine learning and the renormalization group*

Author: Dimitrios S. Bachtis

Supervisors:

Prof. Gert Aarts

Prof. Biagio Lucini

Date of submission:

06/05/2022

Abstract

Within the past decade, machine learning algorithms have been proposed as a potential solution to a variety of research problems which emerge within physics. As this cross-fertilization matures, one is able to investigate if the efficiency of machine learning algorithms can be increased by interpreting them physically and if there exist fundamental connections that can be established between the two research fields. In this thesis, we pursue research directions intimately related to the above questions.

First, we investigate the practical implications of interpreting machine learning functions as statistical-mechanical observables. Through this perspective, we explore if we can extend the classification capabilities of machine learning algorithms and if we are able to include neural networks within Hamiltonians to induce phase transitions in systems. A related direction concerns the use of machine learning to construct inverse renormalization group transformations to arbitrarily increase the size of a system. These techniques are then utilized to study the infinite volume limit of discrete spin systems and of quantum field theories, in order to investigate if machine learning is a powerful tool to study phase transitions.

In another research direction we explore the derivation of machine learning algorithms from quantum field theories. We investigate if the ϕ^4 scalar field theory satisfies the Hammersley-Clifford theorem and if it can be recast as a Markov random field. We then explore if ϕ^4 neural networks can be derived that generalize a certain class of standard neural network architectures, and we present relevant numerical applications. Finally, we discuss how this research direction opens up the opportunity to investigate machine learning within quantum field theory and how it solidifies a rigorous connection between the research fields of machine learning, probability theory, statistical mechanics, lattice and constructive quantum field theory.

Declarations

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed..... 


Date..... 06/05/2022

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed..... 

Date..... 06/05/2022

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed..... 

Date..... 06/05/2022

The University's ethical procedures have been followed and, where appropriate, that ethical approval has been granted.

Signed..... 

Date..... 06/05/2022

Contents

1	Introduction and motivation	9
2	Statistical physics, lattice field theory, and machine learning	15
2.1	Statistical systems	15
2.1.1	The Ising model	15
2.1.2	Potts models	19
2.1.3	The ϕ^4 scalar field theory	20
2.2	Machine learning	21
2.2.1	Feedforward neural networks	21
2.2.2	Gradient-based methods and loss functions	23
2.2.3	Convolutional neural networks	24
2.2.4	Transposed convolutions	25
3	Interpreting machine learning functions as physical observables	27
3.1	Introduction	27
3.2	The Boltzmann weight	29
3.3	Single histogram reweighting	30
3.4	Phase classification in the Ising model	31
3.4.1	Reweighting machine learning functions	32
3.4.2	Scaling of neural network functions	37
3.5	Discussion	42
4	Discovering phase transitions with machine learning	43
4.1	Introduction	43
4.2	Multiple histogram reweighting	44
4.3	Transfer learning	48
4.3.1	Critical exponents of the ϕ^4 theory	51
4.3.2	Searching for universal structures	52
4.4	Discovery of phase transitions: a summary	54

4.5	Discussion	55
5	Neural networks as Hamiltonian terms	57
5.1	Introduction	57
5.2	Conjugate variables and external fields	58
5.3	Hamiltonian-agnostic reweighting	60
5.4	Neural network-induced phase transitions	61
5.5	The renormalization group	63
5.5.1	Fundamentals and the transformation	63
5.5.2	Flows and the critical fixed point	65
5.5.3	The relevant operators	67
5.6	Discussion	71
6	Inverse renormalization group	73
6.1	Introduction	73
6.2	RG flows in the ϕ^4 theory	74
6.3	Inverting a transformation	75
6.3.1	Inverse flows	77
6.3.2	Extraction of critical exponents	80
6.4	Discussion	83
7	Quantum field-theoretic machine learning	85
7.1	Introduction	85
7.2	Probabilistic graphical models and Markov random fields	86
7.3	The ϕ^4 theory as a Markov field	89
7.4	Machine learning with ϕ^4 Markov random fields	91
7.4.1	Learning without predefined data	91
7.4.2	Learning with predefined data	101
7.5	ϕ^4 neural networks	104
7.5.1	Learning with predefined data	107
7.6	Discussion	109
8	Conclusions	111
A	Architectures and simulation details	115
B	Error Analysis	117

Acknowledgements

As a PhD degree nears its completion, one realizes that there is a certain set of people that have immensely influenced one's own progress. On that front, I would like to express my gratitude to my supervisors, Prof. Gert Aarts and Prof. Biagio Lucini, for providing me with academic freedom to pursue research topics that I found of interest. This turned my PhD into an enjoyable experience that I would gladly repeat. I would additionally like to thank them for their support, the helpful scientific insights provided into my work, the advice that they have offered throughout the past years, and for enabling me to build a better academic profile by providing me with the appropriate opportunities at each step of this process. I am grateful for all the lessons learned.

Being a member of a Marie Skłodowska-Curie network, I would like to thank our director, Prof. Francesco Di Renzo, for his support towards all of the fellows during the COVID-19 pandemic and, personally, for being a gracious host during my secondment at the University of Parma. In addition, I would like to thank all of the other members from the participating universities for creating a welcoming research community, and the academic and industrial partners that provided training courses such as in science communication. I would especially like to thank Ernesto Lozano Tellechea for offering me the opportunity to write an article on my research that was published in the scientific magazine *Investigación y Ciencia*.

While completing almost the entirety of my PhD under a pandemic, there were admittedly not many opportunities to immerse myself, at the expected level, within the academic community of my host university in Swansea. Nevertheless, I would like to express my gratitude to Prof. Jeffrey Herschel Giansiracusa who acted as an internal assessor of my PhD progress and who provided interesting research ideas during our discussions. I would also like to thank Nick for the fruitful discussions at the intersection of statistical physics and topological data analysis.

Finally, I would like to thank my parents, Sotirios and Adamantia, and my brother Michail, for their continuous support throughout my life.

Chapter 1

Introduction and motivation

Lattice field theory is a branch of theoretical physics that concerns the study of quantum field theories [1] which have been discretized on spatial or spacetime lattices. By transitioning to Euclidean space, lattice field theories can be directly expressed within a probabilistic setting and a direct link between lattice field theory and classical statistical mechanics is established. In addition, the introduction of a spacetime lattice opens up the opportunity for a mathematically rigorous treatment of quantum field theory, which is explored in the subfield of constructive field theory [2]. Lattice field theory is amenable to computational treatment and one is therefore able to utilize Markov chain Monte Carlo simulations to study, for instance, the phase transitions that emerge within lattice field theory.

Phase transitions are ubiquitous phenomena which arise in distinct research fields, such as condensed matter physics, quantum field theory, chemistry, and computer science. The study of phase transitions is of tremendous appeal due to the concept of universality. In summary, systems with different microscopic descriptions can manifest identical macroscopic behaviour, and their phase transitions therefore belong in an identical universality class. Universality enables a cross-fertilization between distinct research fields since, for instance, a phase transition in a condensed matter system can be identical to that of a quantum field theory. Consequently, one is able to gain insights into universal behaviour of systems across different research fields by studying the most simple system that undergoes a phase transition within a certain universality class.

Recently, deep learning and machine learning algorithms [3] were extended to different research fields. As an example, a vast amount of machine learning applications have emerged within condensed matter, high energy and statistical physics [4]. One might then mistakenly consider that the cross-fertilization between machine learning

and physics was only recently established. This is far from the truth. In fact, one can trace this cross-fertilization, for instance, to the 1980s with the introduction of the Hopfield network [5], a result that relates spin glasses and machine learning algorithms, or with the application of the replica method to obtain results pertinent to neural networks [6, 7]. In light of the cross-fertilization between machine learning and physics, here we establish further connections that relate machine learning with statistical physics or with quantum field theory.

First, we demonstrate that practical implications emerge by viewing machine learning as a concept that can be interpreted physically. An example, to be investigated in this thesis, is the interpretation of machine learning functions as statistical-mechanical observables. We will explore if this perspective opens up the opportunity to apply the complete spectrum of statistical-mechanical techniques to functions derived from machine learning algorithms. This direction then focuses on establishing statistical-mechanical methods that can, for instance, extend the classification capabilities of machine learning algorithms, or that enable the inclusion of machine learning functions as physical terms within Hamiltonians to induce phase transitions in systems. As a result, we aim to explore the efficiency of machine learning, either generally or in relation to physics, after we interpret it physically and what unique benefits, in relation to computational methods, can be provided to physics by machine learning algorithms.

The second part of research questions introduced in this thesis focuses on exactly the opposite direction: it aims to emphasize fundamental connections that can be established between machine learning and quantum field theory, while the numerical applications remain exploratory. An example concerns the derivation of machine learning algorithms and of neural networks from quantum field theories. Specifically, we explore if an equivalence between lattice field theories and the mathematical framework of Markov random fields can be rigorously established. This research direction therefore focuses on the direct investigation of machine learning within quantum field theory, and aims to solidify a rigorous connection between the research fields of probability theory, statistical mechanics, machine learning, lattice and constructive field theory. Consequently, one might be able to view machine learning as a research field which shares similar mathematical questions with quantum field theory, and it is therefore not as distant from mathematical physics as one might generally expect.

Besides the two distinct research directions described above, there is yet another one that combines exploratory and precision studies: the construction of inverse renormalization group transformations with machine learning. In this thesis we explore how inverse renormalization group transformations can be constructed with the use of machine learning for quantum field theories and for systems with continuous degrees

of freedom. We will then utilize the method to conduct a precision study for the phase transition of a lattice field theory, and we will discuss if the inverse renormalization group can avoid intricate computational problems, such as the critical slowing down effect.

This thesis is structured as follows:

Chapter 2 introduces the statistical-mechanical systems and the quantum field theory that will be studied in this thesis, and briefly describes the machine learning architectures that will be utilized in the thesis.

Chapter 3 introduces the first novel result in this thesis, namely the interpretation of machine learning functions as statistical-mechanical observables. These functions are then utilized to study the thermodynamic limit for the phase transition of the two-dimensional Ising model with the use of reweighting and of convolutional neural networks.

Chapter 4 introduces the concept of transfer learning, namely a method utilized to construct effective order parameters in more complicated systems by relying on a neural network that has been trained exclusively on configurations of a simple system. In addition, the single histogram reweighting method for machine learning functions is extended to the multiple histogram method and the phase transition of the two-dimensional ϕ^4 scalar field theory is studied using neural network functions.

Chapter 5 concerns the inclusion of machine learning functions as physical terms within Hamiltonians. This is achieved by viewing a neural network as a conjugate variable coupled to a fictitious external field. The real-space renormalization group approach is then introduced to extract the critical exponents pertinent to the relevant operators and the critical point of the two-dimensional Ising model using machine learning functions.

Chapter 6 extends the ideas pertinent to the renormalization group to the inverse renormalization group method. Machine learning is introduced to construct inverse renormalization group transformations. The inverse renormalization group method is then used to extract multiple critical exponents of the ϕ^4 scalar field theory.

Chapter 7 investigates connections between quantum field theory and machine learning algorithms. Specifically, we demonstrate that the ϕ^4 lattice field theory is a Markov random field. This equivalence is then utilized to derive ϕ^4 neural networks which are generalizations of standard neural network architectures. In addition, numerical results are presented for both ϕ^4 Markov random fields and ϕ^4 neural networks.

Chapter 8, which is the conclusion, summarizes the previous chapters and the novel results presented in this thesis and highlights potential future research directions.

Appendix A provides the necessary details that enable reproducibility of the results, such as the precise machine learning architectures or sampling algorithms used in this thesis.

Appendix B discusses the error analysis techniques used to obtain results in the thesis.

Where appropriate, each of the chapters includes an introductory section that discusses its scientific aim and presents the relevant literature review.

The scientific results presented in this thesis are based on the following manuscripts:

Journal Articles

- [i] D. Bachtis, G. Aarts, F. Di Renzo, and B. Lucini. Inverse renormalization group in quantum field theory. *Phys. Rev. Lett.*, 128:081603, Feb. 2022.
- [ii] D. Bachtis, G. Aarts, and B. Lucini. Quantum field-theoretic machine learning. *Phys. Rev. D*, 103:074510, Apr. 2021.
- [iii] D. Bachtis, G. Aarts, and B. Lucini. Adding machine learning within Hamiltonians: Renormalization group transformations, symmetry breaking and restoration. *Phys. Rev. Research*, 3:013134, Feb. 2021.
- [iv] D. Bachtis, G. Aarts, and B. Lucini. Mapping distinct phase transitions to a neural network. *Phys. Rev. E*, 102:053306, Nov. 2020.
- [v] D. Bachtis, G. Aarts, and B. Lucini. Extending machine learning classification capabilities with histogram reweighting. *Phys. Rev. E*, 102:033303, Sep. 2020.

Conference Proceedings

- [i] D. Bachtis, G. Aarts, and B. Lucini. Quantum field theories, Markov random fields and machine learning. *Journal of Physics: Conference Series*, 2207(1):012056, Mar. 2022.
- [ii] D. Bachtis, G. Aarts, and B. Lucini. Machine learning with quantum field theories, accepted in *Lattice21*, 2021.
- [iii] G. Aarts, D. Bachtis, and B. Lucini. Interpreting machine learning functions as physical observables, accepted in *Lattice21*, 2021.

Chapter 2

Statistical physics, lattice field theory, and machine learning

2.1 Statistical systems

2.1.1 The Ising model

The Ising model [8] is an important system that had significant impact in statistical physics, quantum field theory [1], computer science and machine learning. It is a simple system described by binary degrees of freedom that can take the values of $+1$ or -1 . Despite its simplicity, the Ising model has a rich structure. This is because of its second-order phase transition from an ordered to a disordered phase. The system is therefore simple but non-trivial. In addition, an analytical solution for the two-dimensional Ising model was obtained by Onsager [9]. Consequently, the system provides an ideal setting to benchmark the efficiency of novel computational techniques against the analytically expected values. Moreover, this can be achieved while having to deal with intricate computational problems that emerge in the context of phase transitions, such as the critical slowing down effect.

We consider the Ising model on a two-dimensional square lattice, see Fig. 2.1, described by the Hamiltonian:

$$E = -J \sum_{\langle ij \rangle} s_i s_j - h \sum_i s_i, \quad (2.1)$$

where $\langle ij \rangle$ denotes two lattice sites i and j which are nearest-neighbor, J is a coupling constant which describes the strength of the interaction between two nearest-neighbors, and h is an external magnetic field. Unless otherwise stated we will con-

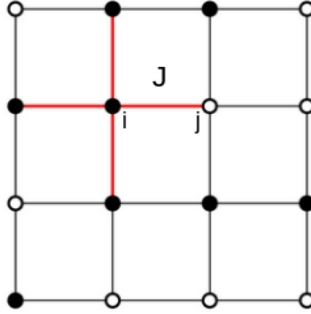


Figure 2.1: The Ising model on a square lattice of size $L = 4$ in each dimension. The interactions J with the nearest neighbours of a lattice site i are depicted by the red lines. The binary degrees of freedom are shown as filled or empty points.

sider that $J = 1$, defining a system in which lattice sites prefer to be aligned towards the same direction in order to minimize the energy of the system. This is called the ferromagnetic Ising model. In addition we consider $h = 0$, hence removing the interaction with an external magnetic field. Finally, we remark that the system is invariant under a reflection symmetry $\{s_i\} \rightarrow \{-s_i\}$ that can be spontaneously broken.

Observables of interest that can be calculated in the Ising model include the absolute value of the normalized magnetization which will be simply called the magnetization:

$$m = \frac{1}{V} \left| \sum_i s_i \right|. \quad (2.2)$$

The statistical fluctuations of the magnetization are equivalent to the magnetic susceptibility χ :

$$\chi = \beta V (\langle m^2 \rangle - \langle m \rangle^2), \quad (2.3)$$

where $V = L \times L$ is the volume of the system and L is the lattice size in each dimension.

The Ising model undergoes a second-order phase transition between an ordered and a disordered phase at a critical value of the coupling denoted β_c . Specifically, at low values of the inverse temperature $\beta \ll \beta_c$ the spins are randomly aligned and no correlations between spins are present. As the inverse temperature of the system increases, but remains below β_c , correlations between adjacent spins begin to emerge, forming clusters of spins which are aligned towards an identical direction. The size of these clusters, measured in terms of lattice units, is called the correlation length ξ and it diverges at the value of the critical inverse temperature β_c . Then as the inverse

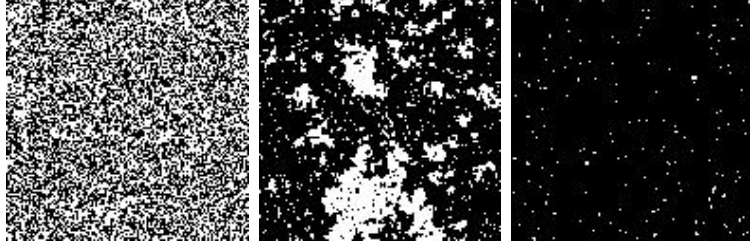


Figure 2.2: Configurations of the Ising model for lattice $L = 128$ in each dimension and $\beta \ll \beta_c$ (left), $\beta \approx \beta_c$ (center) and $\beta \gg \beta_c$ (right).

temperature increases above the critical point β_c , the system spontaneously chooses an ordered state, in which almost all of the spins are aligned towards one direction. In this randomly selected ordered state the system manifests a nonzero magnetization. This behaviour can be observed in Fig. 2.2, and an in-depth treatment of the Ising model's second order phase transition is available, for instance, in Ref. [10].

For the value of the coupling constant $J = 1$ that we will consider in this thesis the two-dimensional Ising model on a square lattice undergoes its second-order phase transition at the critical inverse temperature β_c :

$$\beta_c = \frac{1}{2} \ln(1 + \sqrt{2}) \approx 0.440687. \quad (2.4)$$

We are now interested in defining a quantity which is able to measure the distance of an arbitrary inverse temperature β in relation to the critical inverse temperature β_c . This is achieved via the definition of a reduced coupling constant t :

$$t = \frac{\beta_c - \beta}{\beta_c}. \quad (2.5)$$

Our aim is then to study the second-order phase transition of the Ising model in the thermodynamic limit, namely as $L \rightarrow \infty$ and then $t \rightarrow 0$. When $t \rightarrow 0$, or equivalently $\beta \rightarrow \beta_c$, the system is in the vicinity of the phase transition and its parameter space therefore defines a critical region. Within the critical region we observe critical phenomena due to the increasing correlation length, as discussed above. Specifically, clusters of spins change direction abruptly in the critical region and they therefore give rise to large fluctuations. As an example consider how abruptly the value of the magnetization would change when a large cluster of spins with values $+1$ is replaced by an equally sized cluster with values -1 . Because the correlation length diverges at the critical point β_c the magnitude of the fluctuations that arise in the system will additionally diverge.

We are interested in obtaining insights into how observables diverge in the critical point, and we will achieve this by obtaining a set of critical exponents that govern this divergence related to the correlation length [10]. This is a central aim in this thesis: the critical exponents provide the physics that describes the phase transition of a given systems and their accurate calculation also serves as an indication of the efficiency and general applicability of a novel computational technique. The first quantity of interest is the correlation length, which diverges in the thermodynamic limit according to the relation:

$$\xi \sim |t|^{-\nu}, \quad (2.6)$$

where ν is the correlation length critical exponent. The behaviour of the magnetization m of the system is described by a different critical exponent β_m , when $t < 0$:

$$m \sim |t|^{\beta_m}. \quad (2.7)$$

While the magnetization critical exponent is commonly denoted as β , here we will denote it as β_m to avoid confusion with the inverse temperature β . The fluctuations of the magnetization, namely the magnetic susceptibility χ , diverge in the vicinity of the phase transition according to the critical exponent γ :

$$\chi \sim |t|^{-\gamma}. \quad (2.8)$$

In addition, the specific heat c diverges according to the critical exponent α :

$$c \sim |t|^{-\alpha}. \quad (2.9)$$

Another exponent of interest is the critical exponent δ which governs the divergence of the magnetization m in relation to the external magnetic field:

$$m \sim h^{\frac{1}{\delta}}. \quad (2.10)$$

Equivalently, one can define the critical exponent θ which governs the divergence of the correlation length ξ in terms of the external field h :

$$\xi \sim |h|^{-\theta}. \quad (2.11)$$

In fact, the critical exponents ν and θ are related to the relevant operators of the two-dimensional Ising model's phase transition, and given their knowledge all other exponents can be calculated via scaling relations:

$$\alpha = 2 - \nu d, \quad (2.12)$$

$$\beta_m = \nu \left(d - \frac{1}{\theta} \right), \quad (2.13)$$

$$\gamma = \nu \left(\frac{2}{\theta} - d \right), \quad (2.14)$$

$$\delta = \frac{1}{d\theta - 1}. \quad (2.15)$$

where d is the dimension of a system.

An important concept that emerges in the study of phase transitions is universality, see Ref [11]. Universality implies that the critical exponents and a set of universal measurable quantities remain independent of certain parameters in a given system, such as the topology of the lattice, or the value of the coupling constant J . Different systems can be governed by identical critical exponents and hence belong in a discrete universality class. One important implication of universality is that we are able to study the universal quantities of a complicated system, which can be experimentally relevant, by instead studying a more simple system which belongs in the same universality class. An example of two different systems that belong in the same universality class is the Ising model and the liquid-gas transition at the tri-critical point. The set of critical exponents then define a certain universality class and the two-dimensional Ising universality class is given by the exponents:

$$\nu = 1, \quad \alpha = 0, \quad \beta_m = \frac{1}{8}, \quad (2.16)$$

$$\gamma = \frac{7}{4}, \quad \delta = 15, \quad \theta = \frac{8}{15}. \quad (2.17)$$

2.1.2 Potts models

The q -state Potts models [12] are a generalization of the Ising model for $q > 2$, where the binary degrees of freedom are replaced by discrete values in the range $1, \dots, q$, see Fig. 2.3.

The Potts Hamiltonian is:

$$E_{\text{Potts}} = -J_P \sum_{\langle ij \rangle} \delta(s_i, s_j), \quad (2.18)$$

where $\delta(s_i, s_j)$ is the Kronecker delta. The value of the critical inverse temperature β_c^{Potts} can be obtained analytically for the two-dimensional Potts models and is given by:

$$\beta_c^{\text{Potts}} = \ln(1 + \sqrt{q}), \quad (2.19)$$

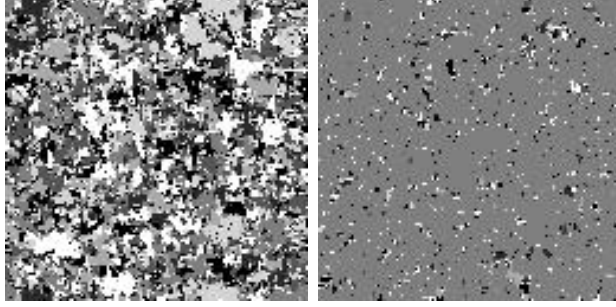


Figure 2.3: A disordered (left) and ordered (right) configuration of the $q = 7$ Potts model.

when $J_P = 1$. The $q = 2$ state Potts model reduces to an Ising model by substituting $s \in \{-1, 1\}$, $J_{\text{Ising}} = J_P/2$. We then obtain $\beta_c^{\text{Ising}} = \beta_c^{\text{Potts}}/2$.

Despite the fact that the Potts models are a generalization of the Ising model, they manifest different critical behaviour for $q \geq 3$. Specifically for the case $q = 3$ and $q = 4$ the systems have second-order phase transitions that belong in distinct universality classes and for $q \geq 5$ the phase transition is of first-order [12].

2.1.3 The ϕ^4 scalar field theory

The final system that will be discussed in this thesis, which is described by continuous degrees of freedom with real values, is the ϕ^4 scalar field theory [13]. To define the system we start from the Euclidean Lagrangian:

$$\mathcal{L}_E = \frac{\kappa}{2}(\nabla\phi)^2 + \frac{\mu_0^2}{2}\phi^2 + \frac{\lambda}{4}\phi^4. \quad (2.20)$$

The system is then discretized on a square lattice with spacing α , obtaining the Euclidean lattice action:

$$S_E = \sum_n \left[\frac{1}{2} \sum_{\nu=1}^{d=2} \kappa_L (\phi_{n+e_\nu} - \phi_n)^2 + \frac{1}{2} \mu_L^2 \phi_n^2 + \frac{1}{4} \lambda_L \phi_n^4 \right], \quad (2.21)$$

where κ_L , μ_L^2 , λ_L are dimensionless parameters, one of which can be absorbed by rescaling the fields. We can expand the terms in the above equation to obtain an expression in relation to lattice sites i and j which are nearest-neighbours $\langle ij \rangle$.

$$S_E = -\kappa_L \sum_{\langle ij \rangle} \phi_i \phi_j + \frac{(\mu_L^2 + 4\kappa_L)}{2} \sum_i \phi_i^2 + \frac{\lambda_L}{4} \sum_i \phi_i^4. \quad (2.22)$$

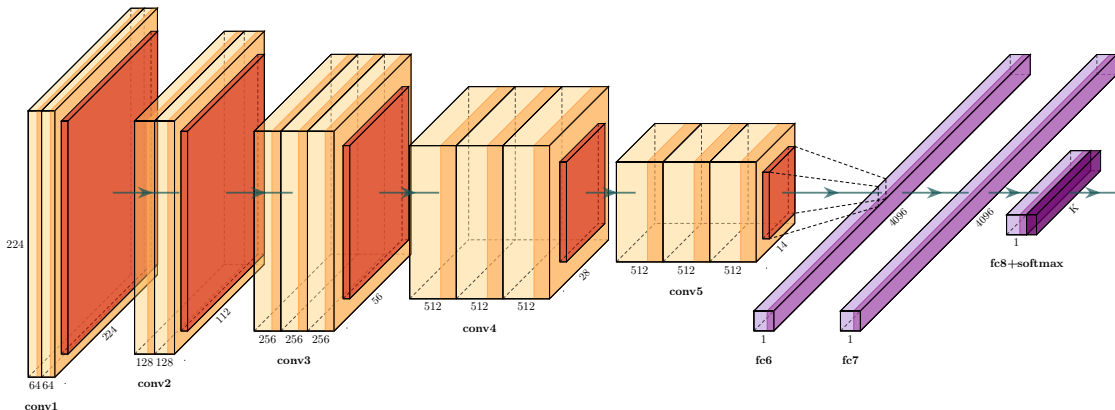


Figure 2.4: A neural network which comprises convolutional and fully-connected layers.

An equivalence between the ϕ^4 scalar field theory and the Ising model can be obtained in the limit κ_L positive and fixed, $\lambda_L \rightarrow \infty$ and $\mu_L^2 \rightarrow -\infty$, where one obtains a binary system. In addition, for κ_L and λ_L positive and fixed one discovers a second-order phase transition for a value of a critical squared mass $\mu_L^2 < 0$ which is conjectured to be in the Ising universality class [13].

2.2 Machine learning

2.2.1 Feedforward neural networks

Deep learning architectures [3] have been extensively used in the past decade to efficiently complete tasks pertinent to supervised machine learning. These architectures comprise a set of iterative layers, where each layer consists of a number of hidden variables. By increasing the number of layers, as well as the number of hidden variables within each layer, one obtains an architecture that can represent intricate functions. Here, we will briefly review relevant machine learning architectures which are utilized throughout the thesis, such as convolutional and fully-connected neural networks, see Fig. 2.4¹. Another architecture that will be investigated in this thesis is the class of neural networks called restricted Boltzmann machines, but complete derivations for these systems are included in Chapter 7.

¹The figure has been produced based on the code from PlotNeuralNet, <https://github.com/HarisIqbal88/PlotNeuralNet>, v.1.0.0, MIT License.

In the initial chapters of this thesis, we are interested in feedforward neural networks. A neural network is called feedforward when the iterative mappings imposed on the input vector \mathbf{x} at each layer do not include a feedback connection with a previous layer. The input is hence iteratively processed by independent transformations until the output layer is reached. In other words, these architectures are used to map an input vector \mathbf{x} to a desired output y via the approximation of an appropriate function f . We remark that in this section a bold symbol denotes a vector. The neural network architecture then defines a mapping $\mathbf{y} = f(\mathbf{x}; \boldsymbol{\theta})$ which depends on a set of variational parameters $\boldsymbol{\theta}$, and \mathbf{y} is the vector of all possible outputs. Our aim is to learn the optimal values of these parameters $\boldsymbol{\theta}$ that provide the optimal solution to the stated machine learning problem.

We remark that the layers of a neural network construct a chain structure of functions that iteratively process the input vector. Explicitly, a number of functions $f^{(i)}$ is associated to each layer i , where $i = 1, 2, \dots, n$. Consequently the output of the neural network function $f(\mathbf{x})$ is obtained as:

$$f(\mathbf{x}) = f^{(n)}(f^{(n-1)}(\dots(f^{(1)}(\mathbf{x}))))). \quad (2.23)$$

The number of layers n then defines the depth of the deep learning architecture. We emphasize that neural networks are approximative methods and, as a result, the output function $f(\mathbf{x})$ that corresponds to the output layer, is an approximation of the desired function that we are interested in constructing. Through the optimization process the intermediate layers learn a set of features that provide the optimal solution at the output. Nevertheless, explicit information about the interpretability of these features is unavailable, thus the intermediate layers are called hidden.

To complete the definition of the aforementioned neural network architectures we must discuss the importance of nonlinearities. If the set of functions in the chain structure is selected as linear, then the machine learning algorithm will only be able to discover a set of linear features. This imposes a constraint on the expressivity of the machine learning algorithm, namely the class of functions that it is able to represent, since linear features are incapable of extracting information pertinent to the interaction of two distinct input variables. Consequently, in relation to applications, one is generally interested in introducing nonlinearities within a neural network architecture to increase the expressivity of the machine learning algorithm.

The inclusion of nonlinearities in a machine learning architecture is achieved via the introduction of nonlinear functions g which transform the output of a certain layer. Specifically, one is interested in modelling a function $y = f(\mathbf{x}; \boldsymbol{\theta}, \mathbf{w}) = g(\mathbf{x}, \boldsymbol{\theta})^\top \mathbf{w}$. To clarify, \mathbf{w} map $g(\mathbf{x}; \boldsymbol{\theta})$ to the output, where g now denotes a hidden layer. Through the inclusion of nonlinearities in a neural network the expressivity of the algorithm

is increased, but certain implications pertinent to the optimization process emerge, which will be discussed in the following subsection.

2.2.2 Gradient-based methods and loss functions

Optimization techniques that are established on a gradient-based approach are common among distinct research fields. Nevertheless, in contrast to the linear case, the optimization of a loss function for a neural network which includes nonlinearities is often nonconvex.

From a practical perspective, nonconvexity implies that the optimization process, which is generally established on a stochastic gradient-descent method, might be unable to reach a global minimum after a certain set of iterations, which are called epochs. The global minimum corresponds to the optimal solution of the problem. In other words, no mathematical proofs can be obtained to guarantee the convergence of the optimization process in the nonconvex case. In addition, the process is substantially influenced by the initial values of the variational parameters in the neural network architecture, specifically the set of weights and biases. Based on empirical observations one is therefore advised to initialize the weights randomly to small positive and negative values, and the biases can be initialized to a value of zero. Nevertheless, each problem might require a different approach and there exists vast literature on the topic, for instance see Ref. [3].

A central concept in the construction of a neural network architecture is the choice of a loss function. The loss function encodes the aim of the machine learning task. For instance one might be interested in learning a loss function which can accurately separate a set of examples \mathbf{x} , which are labeled based on outputs \mathbf{y} . This can be achieved by defining an appropriate conditional probability distribution:

$$p(\mathbf{y}|\mathbf{x}; \boldsymbol{\theta}). \quad (2.24)$$

The training of the machine learning algorithm is then conducted by minimizing a distance function between the training data and the model predictions through the gradient-based approach. For example, in Chapter 3 this can be achieved via the minimization of the cross-entropy:

$$\mathcal{L}(\theta) = -\frac{1}{N} \sum_i^N \left[y_i \log \hat{y}_i(\theta) + (1 - y_i) \log(1 - \hat{y}_i(\theta)) \right], \quad (2.25)$$

where y_i is the correct label, \hat{y}_i is the predicted label which depends on the set of parameters θ , and N is the number of samples. In other cases, the loss function might

be more simple, for instance it could be selected as a mean squared error function between the training data and the machine learning predictions. This is the case in Chapter 6, where the loss function is:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i(\theta))^2, \quad (2.26)$$

where y_i denotes the correct label and \hat{y}_i is the predicted label which depends on a set of parameters θ . The successful minimization of a loss function can be affected by the presence of local minima, the choice of the minimizer, the length of training, and the quality and quantity of the training data set.

2.2.3 Convolutional neural networks

We will briefly review convolutional and fully-connected neural networks. A convolutional neural network comprises an input layer in which we position the data, namely the configurations of a system. The input layer is then followed by potentially multiple hidden layers such as convolutional, pooling, normalization, or fully-connected layers.

A convolutional layer comprises a set of filters which are convolved on the input via a dot product. The aim is to learn the appropriate set of filters that is able to uncover dependencies, such as spatial structures, on the input data.

A pooling layer, which is reminiscent of a real-space transformation, aims to reduce the number of degrees of freedom within the machine learning algorithm. Specifically, the input is separated into blocks of size $b \times b$. For the case of max-pooling used in the thesis all values within each block, except the one with the largest magnitude, are discarded. Max-pooling therefore produces an output with a reduced number of degrees of freedom.

A fully-connected layer then associates all degrees of freedom from the input to a neuron. This is in contrast to convolutional layers, which do not provide full connectivity.

The output layer comprises a loss function which depends on the machine learning task. For instance in this thesis we will consider the output loss function as a softmax function which aims to predict a certain phase of a configuration out of all possible phases. The softmax function is related to the Boltzmann probability distribution and is given by:

$$s(x_i) = \frac{\exp[x_i]}{\sum_{j=1}^n \exp[x_j]}, \quad (2.27)$$

where n is the number of x real values. The outputs of the softmax functions reside in the range $[0, 1]$ and sum to one, thus defining a probability distribution.

Finally, nonlinear functions, such as rectified linear units (ReLUs), which correspond to the function $h(x) = \max(0, x)$, are positioned between the middle layers of the neural network architecture. Detailed explanations of neural networks can be found in Ref. [3].

2.2.4 Transposed convolutions

In this thesis, we additionally utilize transposed convolutions [14], so we will briefly review the mathematical operation of a transposed convolution. Let us consider that we have an input, represented as a 2×2 matrix, which is:

$$\begin{array}{|c|c|} \hline 3 & 1 \\ \hline 2 & 0 \\ \hline \end{array}$$

We consider that this matrix corresponds to the degrees of freedom which are positioned on a specific lattice or graph. Our aim is now to apply a filter of size 2×2 , given by $[w_{11}, w_{12}, w_{21}, w_{22}]$ on this input matrix using a transposed convolution to produce an output of increased size. Let us consider that the weights w_{ij} in the filter are equal to:

$$\begin{array}{|c|c|} \hline 2 & 3 \\ \hline 0 & 1 \\ \hline \end{array}$$

The application of the filter on the input then produces:

$$\begin{array}{|c|c|c|} \hline 6 & 9 & \\ \hline 0 & 3 & \\ \hline & & \\ \hline \end{array} + \begin{array}{|c|c|c|} \hline & 2 & 3 \\ \hline & 0 & 1 \\ \hline & & \\ \hline \end{array} + \begin{array}{|c|c|c|} \hline & & \\ \hline 4 & 6 & \\ \hline 0 & 2 & \\ \hline \end{array} + \begin{array}{|c|c|c|} \hline & & \\ \hline & 0 & 0 \\ \hline & 0 & 0 \\ \hline \end{array}$$

where the empty cells have zero values. The output is then equal to:

$$\begin{array}{|c|c|c|} \hline 6 & 11 & 3 \\ \hline 4 & 9 & 1 \\ \hline 0 & 2 & 0 \\ \hline \end{array}$$

It then becomes clear that by applying a set of transposed convolutions one is able to construct lattices of increased size. In addition, further operations can be

applied to the output, such as nonlinear functions or another set of (transposed) convolutions to further manipulate the size. By learning the proper weights through the minimization of a loss function, one is able to establish an equivalence between a model system of lattice size L in each dimension and a target original system of identical lattice size L which is described by fixed degrees of freedom.

Chapter 3

Interpreting machine learning functions as physical observables

3.1 Introduction

Machine learning applications pertinent to the discovery of phase transitions have recently emerged in physics. A large amount of these applications concerns the construction of a function that is able to accurately separate phases of a system. The motivation behind these applications is that, ultimately, one might be able to construct effective order parameters for phase transitions in systems where conventional order parameters are absent or unknown. To explore this topic, a variety of machine learning architectures, either within a supervised or unsupervised setting, have been employed. Here, we will briefly review a selection of these contributions which is relevant for this work.

The construction of a function to separate phases in systems such as the Ising model, square-ice, and the Ising gauge theory was established in Ref. [15]. Simultaneously, an alternative approach was introduced in systems such as the Kitaev chain, the Ising model, and disordered quantum spin chains, in which the discovery of the phase transition is achieved based on the training of a neural network on data that are deliberately labelled incorrectly [16]. The construction of a novel effective order parameter based on a convolutional neural network was additionally discussed in the context of the two-dimensional Ising model in Ref. [17], where it was utilized to locate the critical temperature.

Further work includes the study of phase transitions with the use of machine learning for the Heisenberg spin-1/2 chain in a random external field [18], quantum many-fermion systems [19] which are affected by the sign problem [20] and the

Kosterlitz-Thouless transition of the two-dimensional XY model [21]. Concerning unsupervised learning, the phase diagrams of interacting boson and fermion models at zero and finite temperatures were obtained, irrespective of the general topology or the number of distinct phases, with the use of machine learning [20], and Boltzmann machines were additionally utilized to generate states in the critical region of the two-dimensional Ising model [22]. Furthermore, an analysis of neural network-based schemes with a single hidden layer was conducted in Ref. [23], and a finite-size scaling calculation of the quantum Hall plateau transition was presented in Ref. [24]. Concerning the study of networks and complex dynamical systems, Ref. [25] explores phase transitions in epidemic spreading dynamics. The use of recurrent neural networks was additionally explored in the context of phase transitions in Ref. [26], and non-equilibrium phase transitions of many-body localized or topological phases were studied in Ref. [27], whereas nonergodic metallic phases of quantum systems were investigated in Ref. [28].

Other machine learning algorithms which are utilized to study phase transitions include Gaussian process regression in the context of quantum systems [29], principal component analysis and variational autoencoders in the context of the two-dimensional Ising model and the three-dimensional XY model [30], a combination of multiple algorithms including autoencoders, random trees, and t -distributed stochastic neighboring ensemble for the Ising and Fermi-Hubbard models [31], and diffusion maps for the two-dimensional XY model and the Ising gauge theory [32]. Support vector machines are another class of machine learning algorithms which provide interpretable results and have been utilized to study the ferromagnetic Ising model [33], the conserved-order-parameter Ising model and the Ising gauge theory [34], nematic order parameters [35], and the multiclassification of distinct phases in systems [36].

In this chapter, we will provide a different perspective on the aforementioned work by exploring the implications of physically interpreting the function learned by a machine learning algorithm as a statistical-mechanical observable [37]. We will provide an explanation of how a neural network function, once applied to a configuration, can be associated with a Boltzmann weight and we will then exploit this perspective to apply traditional statistical-mechanical techniques to functions learned from machine learning algorithms. Consequently, through this perspective, and with the use of histogram reweighting, we will be able to obtain machine learning predictions in different regions of the system's parameter space without requiring new data. As a result, we are able to extend the classification capabilities of machine learning algorithms. We will then conduct a finite-size scaling analysis, based on the neural network function, which acts as an effective order parameter, to calculate multiple critical exponents and the critical inverse temperature for the phase transition of the two-dimensional

Ising model.

3.2 The Boltzmann weight

Consider an arbitrary statistical system which is described by a Hamiltonian E and a Boltzmann probability distribution p . We denote as p_{σ_i} the probability that a configuration σ_i appears in the equilibrium distribution. This probability is given by:

$$p_{\sigma_i} = \frac{\exp[-\beta E_{\sigma_i}]}{\sum_{\sigma} \exp[-\beta E_{\sigma}]}, \quad (3.1)$$

where β is the inverse temperature and the sum is over all possible states σ of the system. An important quantity that appears in the previous expression is the partition function:

$$Z = \sum_{\sigma} \exp[-\beta E_{\sigma}], \quad (3.2)$$

which is a normalization constant. Despite being a simple normalization constant, the partition function Z is of high importance in statistical physics, quantum field theory, and related research fields, since it explicitly encodes all of the information that is required to obtain complete knowledge of a statistical system: quantities of interest for a considered system can be derived in terms of the partition function. Knowledge of the partition function then implies knowledge of the statistical system.

Consider now that we want to obtain through Markov chain Monte Carlo simulations, and based on the probability distribution of Eq. (3.1), a representative subset of samples or configurations σ of the system. We require a subset of these configurations because we are interested in calculating the expectation value $\langle O \rangle$ of an arbitrary observable O which could be, for instance, the magnetization m of the system, or the internal energy E which is equal to the Hamiltonian, or any other quantity of interest. The numerical estimator, which is equivalent to the expectation value $\langle O \rangle$ calculated on this finite subset of configurations that we obtained through Markov chain Monte Carlo simulations, is given by

$$\langle O \rangle = \frac{\sum_{i=1}^N O_{\sigma_i} \tilde{p}_{\sigma_i}^{-1} \exp[-\beta E_{\sigma_i}]}{\sum_{i=1}^N \tilde{p}_{\sigma_i}^{-1} \exp[-\beta E_{\sigma_i}]}, \quad (3.3)$$

where \tilde{p}_{σ_i} are the probabilities that we use to sample each configuration σ_i from the equilibrium distribution and the sum i is over the number N of sampled configurations.

An efficient choice of a sampling probability is made by selecting \tilde{p}_{σ_i} as the probabilities given by Eq. (3.1). This choice leads to the most common and successful way of conducting Markov chain Monte Carlo simulations, which is called importance sampling [10]. With this choice the expectation value of an arbitrary observable in Eq. (3.3) becomes:

$$\langle O \rangle = \frac{1}{N} \sum_{i=1}^N O_{\sigma_i}. \quad (3.4)$$

Indeed, we will directly sample systems at a specific inverse temperature β with importance sampling and calculate expectation values of their observables based on the above equation. However, in a large part of the current work we are interested in a method of obtaining expectation values based on the original dataset, sampled at β , but when the inverse temperature differs $\beta' \neq \beta$: this method is called histogram reweighting [38].

3.3 Single histogram reweighting

The idea of histogram reweighting can be summarized as follows. First, assume that we have conducted a Markov chain Monte Carlo simulation for a specific value of the inverse temperature β and we have obtained a subset of configurations that would correspond to this inverse temperature β . Now instead of using Eq. (3.4) to calculate expectation values of observables for the system at inverse temperature β , we are instead interested on using the configurations sampled at inverse temperature β to calculate, accurately, expectation values that would correspond to a different inverse temperature β' , and we are interested in achieving this without ever sampling configurations at inverse temperature β' .

The method is very simple, and here we will discuss the case called single histogram reweighting. Specifically, we consider the expectation value of an observable O that corresponds to a sufficiently adjacent inverse temperature β' in the system's parameter space:

$$\langle O \rangle = \frac{\sum_{i=1}^N O_{\sigma_i} \tilde{p}_{\sigma_i}^{-1} \exp[-\beta' E_{\sigma_i}]}{\sum_{i=1}^N \tilde{p}_{\sigma_i}^{-1} \exp[-\beta' E_{\sigma_i}]}, \quad (3.5)$$

We are now interested in approximating the above expectation value based on the sampled configurations that we have obtained at inverse temperature β . We will therefore replace \tilde{p}_{σ_i} in the above equation with the probabilities p_{σ_i} that correspond to inverse temperature β . The expectation values of observables at this inverse temperature β' are then calculated, by using configurations obtained at the original

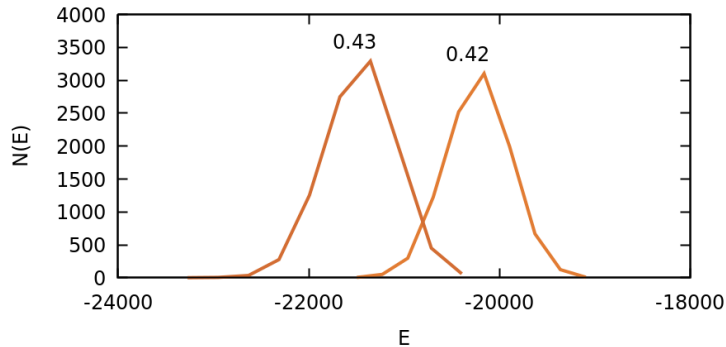


Figure 3.1: Histograms $N(E)$ versus value of the energy E for two cases of inverse temperatures $\beta = 0.42$ and $\beta = 0.43$ of a two-dimensional Ising model with lattice size $L = 128$.

inverse temperature β , via the following expression:

$$\langle O \rangle_{\beta'} = \frac{\sum_{i=1}^N O_{\sigma_i} \exp[-(\beta' - \beta)E_{\sigma_i}]}{\sum_{i=1}^N \exp[-(\beta' - \beta)E_{\sigma_i}]} \quad (3.6)$$

In essence, what the method of reweighting achieves is that it provides a way to predict the histograms of the Hamiltonian at an extrapolated inverse temperature β' based on the histograms of the Hamiltonian at the original inverse temperature β . As a result, there exists a permitted reweighting range for the method to be successful, which is dependent on the overlap of histograms between the ensembles that correspond to the two inverse temperatures. This is illustrated in Fig. 3.1, where a partial overlap of the histograms of the energy can be observed for the inverse temperatures $\beta = 0.42$ and $\beta = 0.43$. By starting from $\beta = 0.42$ one could therefore extrapolate observables with histogram reweighting towards values of the inverse temperature $\beta' \rightarrow 0.43$. Nevertheless, one might not be able to exactly extrapolate from $\beta = 0.42$ to $\beta' = 0.43$ since the overlap between ensembles is not complete. For more details, see Ref. [10].

3.4 Phase classification in the Ising model

Our aim in this chapter is to interpret machine learning functions as statistical-mechanical observables, but to achieve this we must first construct a machine learning function. As discussed in the introductory section, an example can be established

based on a phase classification machine learning task related to the second-order phase transition of the Ising model.

Consider a subset of configurations of the Ising model which have been drawn either from the symmetric (disordered) or the broken-symmetry (ordered) phase. We assign to each configuration σ_i a label $y_{\sigma_i} = 0$ or $y_{\sigma_i} = 1$ depending on the phase that the configuration is associated with. Our aim is to train a machine learning algorithm on the set of the available configurations to successfully learn an optimal function $f(\sigma_i)$, which is able to correctly classify the phase of an unknown configuration σ_i , see Fig. 3.2. By unknown, we mean a configuration σ_i which has not been presented as input to the machine learning algorithm during the training process. In the current chapter, we will consider a convolutional neural network.

Convolutional neural networks are a class of machine learning algorithms which comprise multiple layers and have been extensively used in computer vision tasks and image recognition [3]. Convolutional neural networks can reduce connectivity between the neurons by implementing the mathematical operation of convolution in at least one of their layers and hence enable the study of systems, or images, with large sizes in each dimension. In the current thesis, we are interested in studying the two-dimensional Ising model of lattice size L as $L \rightarrow \infty$ and convolutional neural networks therefore can be utilized to implement machine learning on larger lattice sizes compared to other architectures, such as fully-connected neural networks.

The training process of the convolutional neural network is established based on the minimization of a loss function \mathcal{L} . Specifically, for the discussed example, we consider as a loss function the cross-entropy:

$$\mathcal{L}(\theta) = -\frac{1}{N} \sum_i^N \left[y_i \log \hat{y}_i(\theta) + (1 - y_i) \log(1 - \hat{y}_i(\theta)) \right], \quad (3.7)$$

where y_i is the correct label of the configuration σ_i that acts as a training example, \hat{y}_i is the predicted label which depends on the set of parameters θ , and N is the number of samples. Through the minimization of the loss function the machine learning algorithm is able to learn an optimal function f that is able to accurately separate configurations which belong in distinct phases.

3.4.1 Reweighting machine learning functions

Once the convolutional neural network is successfully trained, we can present as input an unknown configuration σ_i to predict its corresponding phase based on the function $f(\sigma_i)$. The function $f(\sigma_i)$ is bounded between $[0, 1]$ and we interpret it as the probability $P^{(b)}$ that the configuration σ_i belongs in the broken symmetry phase.

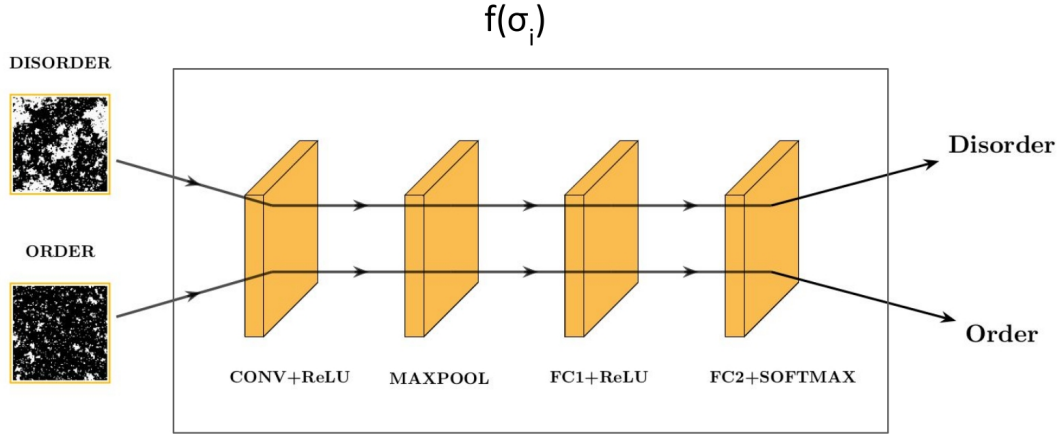


Figure 3.2: The architecture of the convolutional neural network. A set of labeled configurations from each distinct phase are given as input to the neural network in order to learn the optimal function $f(\sigma_i)$ that is able to accurately predict the phase of a configuration σ_i .

We remark that one can obviously obtain the probability $P^{(s)}$ of a configuration being in the symmetric phase via $1 - f(\sigma_i)$.

The interpretation of machine learning functions as statistical-mechanical observables is then an implication of the following observation. The convolutional neural network function $f(\sigma_i)$, which has been calculated on a configuration drawn from an equilibrium distribution via a series of transformations, see Fig. 3.3, is a physically meaningful quantity: it has been learned on a set of importance-sampled configurations of the Ising model and it expresses the probability that the configuration σ_i is associated with the broken-symmetry phase. Furthermore, the unknown configuration σ_i is additionally drawn from an equilibrium distribution $p(\sigma_i; \beta)$ and it is therefore associated with its own corresponding Boltzmann weight of a specific inverse temperature β . As a result $f(\sigma_i)$ is described by the same Boltzmann weight as the configuration σ_i and is a statistical-mechanical observable. Equivalently, the expectation value of the neural network function is:

$$\langle f \rangle = \sum_{\sigma} f_{\sigma} p(\sigma; \beta). \quad (3.8)$$

An important observation is that the expectation value $\langle f \rangle$ of the convolutional neural network function f is expressed as a sum over all possible states σ of the system, weighed by the associated Boltzmann distribution, and as a result the

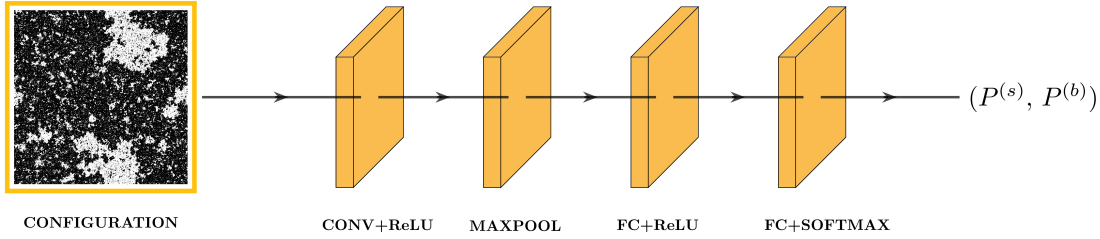


Figure 3.3: A configuration with an unknown phase is presented as input to a trained convolutional neural network to obtain the probability that the configuration belongs in the broken-symmetry phase.

convolutional neural network function f has the proper dependence on the inverse temperature β . One practical implication of this perspective is that we can therefore reweight the neural network function f in parameter space, to predict its value at a different inverse temperature β' . Consequently, we are able to get machine learning predictions for data at a specific inverse temperature β' without having to ever obtain such a dataset. Equivalently, we are able to extend the classification capabilities of machine learning algorithms.

For convenience we rewrite the single histogram reweighting equation for the case of the neural network function f :

$$\langle f \rangle_{\beta'} = \frac{\sum_{i=1}^N f_{\sigma_i} \exp[-(\beta' - \beta)E_{\sigma_i}]}{\sum_{i=1}^N \exp[-(\beta' - \beta)E_{\sigma_i}]}.$$
 (3.9)

We then proceed in the following manner. First, we conduct a Monte Carlo simulation to obtain a set of configurations at a specific inverse temperature β . Second, we present as input to the trained neural network each of the sampled configurations σ_i to obtain the prediction f_{σ_i} that configuration σ_i belongs in the broken-symmetry phase. We are now able to estimate what the neural network prediction would be, specifically the expectation value $\langle f \rangle_{\beta'}$, at a different inverse temperature β' via Eq. (3.9). This is achieved without having to sample configurations at that specific inverse temperature β' . We recall that reweighting is applicable only under a certain range of extrapolated inverse temperatures.

Let us now see a worked example of the above ideas in Fig. 3.4. Specifically, we have conducted a Markov chain Monte Carlo simulation for a system of lattice size $L = 128$ in each dimension to obtain configurations at inverse temperature $\beta_1 = 0.438$ (top) and $\beta_2 = 0.44$ (bottom). In both cases the output of the neural network function

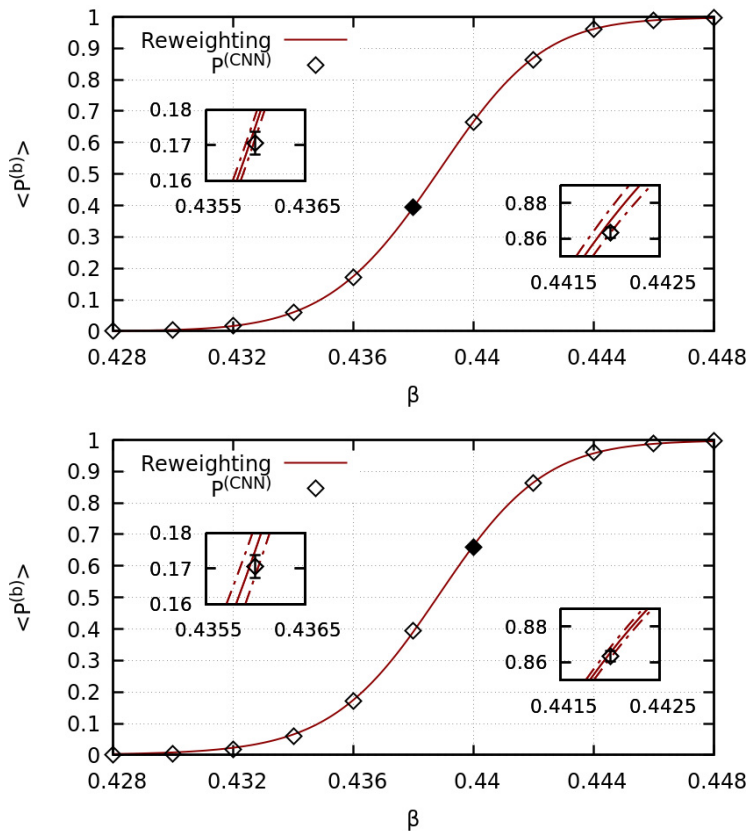


Figure 3.4: Expectation value of the neural network function $f \equiv P^{(b)}$ versus inverse temperature β for a system with lattice size $L = 128$. The Monte Carlo datasets used to conduct reweighting have been obtained at values of the inverse temperature $\beta = 0.438$ (top) and $\beta = 0.44$ (bottom) and their corresponding expectation values are depicted by the filled points. The reweighted extrapolations of the neural network function f are depicted by the red lines. Independent calculations, obtained by presenting as input to the neural network configurations from Monte Carlo simulations at different inverse temperatures are depicted by the empty points.

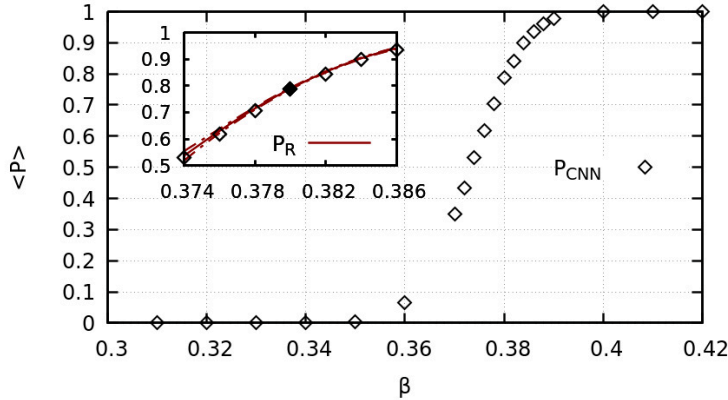


Figure 3.5: Expectation value of a neural network function $f \equiv P$ versus inverse temperature β . The neural network function has been constructed by training on configurations within an identical phase that have a different correlation length. The figure is produced to demonstrate that generic neural network functions can be reweighted in parameter space, as evident in the inset. See text for more details.

$f \equiv P^{(b)}$ is depicted with the filled point and the extrapolation of the neural network function f is depicted by the red line. Calculations of the neural network function on independent Monte Carlo datasets sampled at various inverse temperatures, which are depicted by the empty points, are additionally included in order to allow direct comparisons with reweighting, thus enabling the investigation of the accuracy of the method.

We observe that there exists an overlap within statistical errors between the extrapolated red line and the independent Monte Carlo calculations, therefore certifying that the method is accurate. When comparing the two figures in the insets we additionally observe that extrapolations which are conducted further in parameter space in comparison with the initial dataset have increased statistical errors. This behaviour is anticipated since the further we extrapolate in parameter space the smaller the overlap between the histograms of the energy between the two inverse temperatures β and β' is, and therefore the accuracy of the extrapolations is anticipated to diminish, a result that we have verified in the figure.

Now that we have established that the neural network function f can be extrapolated in the system's parameter space we can proceed to further interpret physically the behaviour of this observable. Based on the results depicted in Fig. 3.4, we observe that the neural network function resembles an effective order parameter. Order parameters are quantities used to characterize a phase transition and generally they

are intimately related to the breaking of an underlying symmetry. Specifically, an order parameter is anticipated to be zero in the symmetric phase of the system and manifest a finite value in the broken-symmetry phase. This is exactly the behaviour of the neural network function f as depicted in the above figure. We can therefore investigate if this neural network function can be utilized to accurately study the phase transition of the system. This will be the topic of the next section.

Before proceeding to the next section, we will investigate if it is possible to reweight in parameter space general neural network functions. In the previous example, the neural network function resembles an effective order parameter and one could therefore claim that the neural network has simply learned a quantity which is practically equivalent to the magnetization, thus enabling reweighting in parameter space. Here, we will demonstrate that more general functions learned from machine learning algorithms can be reweighted in parameter space.

As an example we will construct a neural network function f that is able to separate configurations within exclusively one phase. These could be, for instance, configurations with different values of correlation length. We then construct this function on a training dataset which comprises configurations from inverse temperatures $\beta = 0.31$ and $\beta = 0.32$, labeled as zero and configurations $\beta = 0.41$ and $\beta = 0.42$, labeled as one. The results obtained by presenting as input configurations from intermediate inverse temperatures are depicted in Fig. 3.5. We observe in the inset, where the data are compared with independent calculations, that the reweighting of the neural network function is accurate within statistical errors. Consequently, the expectation values of general functions learned with machine learning algorithms can be extrapolated in a system's parameter space. We remark that even though the neural network function resembles an effective order parameter in Fig. 3.5, one does not anticipate that it will manifest the appropriate scaling behaviour since it separates configurations within an identical phase.

We will now focus, in the next section, in studying the scaling behaviour of the neural network function f constructed in Fig. 3.4, where it resembles the behaviour of an effective order parameter. The aim is to explore if multiple critical exponents as well as the critical inverse temperature can be obtained by utilizing exclusively the function derived from the machine learning algorithm.

3.4.2 Scaling of neural network functions

In studies of phase transitions we are generally interested in the behaviour of the system as the lattice size becomes infinite $L \rightarrow \infty$ and as we approach the critical point. Nevertheless, in computational studies of phase transitions we are studying

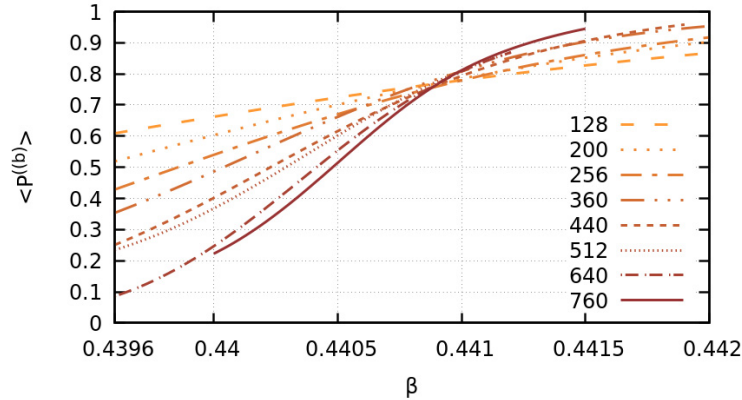


Figure 3.6: Neural network function $f \equiv P^{(b)}$ versus inverse temperature β for lattice sizes $L = 128, \dots, 760$ in each dimension. The reweighted extrapolations have been truncated within accurate ranges, as determined from an overlap of histograms.

finite systems since we have to represent the statistical system on a finite lattice size. As a result, we will not observe the anticipated divergences of the infinite-volume limit directly, but instead we can observe that quantities of interest, such as fluctuations, will manifest maximum values. A common way to study a phase transition is then established based on the study of the order parameter as well as its susceptibility, which is equivalent to the fluctuations of the order parameter.

For the case of the Ising model, the order parameter is the magnetization m , and the susceptibility of the magnetization χ_m is therefore expected to manifest a maximum value in the vicinity of the phase transition. This maximum value of the magnetic susceptibility χ_m^{\max} will appear for a value of a pseudo-critical inverse temperature $\beta_c^{\chi_m^{\max}}(L)$ on a finite system with lattice size L in each dimension. In the thermodynamic limit, the values of the pseudo-critical inverse temperatures will converge to the actual critical point β_c :

$$\lim_{L \rightarrow \infty} \beta_c^{\chi_m^{\max}} = \beta_c. \quad (3.10)$$

An important observation is that the values of the pseudo-critical inverse temperatures differ for different observables O . For example the pseudo-critical inverse temperatures obtained by the susceptibility χ_m of the magnetization will be different from the ones obtained by the susceptibility of the internal energy χ_E . However, both of these will converge to the correct critical point in the infinite-volume limit. As a result the only way to truly verify the correct value of the critical point is via

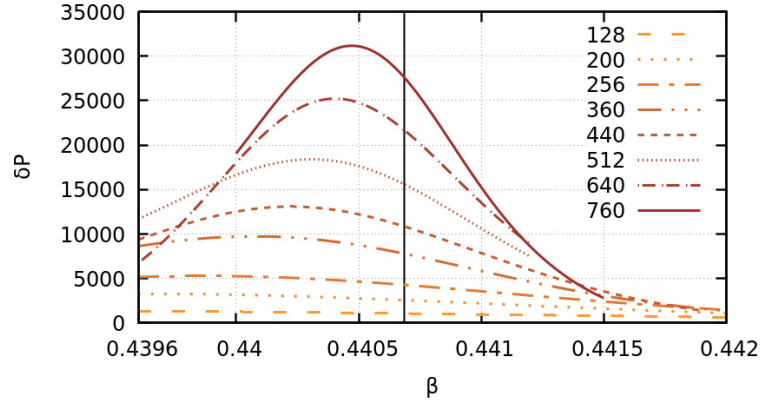


Figure 3.7: Susceptibility of the neural network function $\chi_f \equiv \delta P$ versus the inverse temperature β for systems of lattice size $L = 128, \dots, 760$ in each dimension. We observe a tentative convergence of the maxima of the susceptibility towards the critical point β_c , which is depicted by the vertical line.

extrapolations in the thermodynamic limit.

Based on the above discussion we can now investigate whether the neural network function $f \equiv P^{(b)}$ manifests the anticipated behaviour of the conventional order parameter. In Fig. 3.6 we depict the neural network function f for lattice sizes $L = 128, \dots, 760$. We observe that as the lattice size increases the transition from the value $f = 0$ to $f = 1$ is sharper, which is the expected behavior of the conventional order parameter.

We will now explore if we can obtain the value of the critical point based on the convergence of the pseudo-critical points which are obtained from the susceptibility of the neural network function χ_f . The results are depicted in Fig. 3.7. For each lattice size L we observe that the maximum values of the susceptibility are shifted towards the vertical line which is the value of the critical inverse temperature β_c as obtained from the exact solution of the two-dimensional system. To verify that the pseudo-critical temperatures converge to the critical inverse temperature β_c and not some other value of the inverse temperature we must conduct a calculation in the thermodynamic limit. One widely applicable method to conduct such a study is through a finite size scaling analysis.

To conduct a finite size scaling analysis, we recall that the relation which describes the divergence of the correlation length is $\xi \sim |t|^{-\nu}$, where t is the reduced coupling constant and ν is the correlation length exponent. For a finite system in the vicinity of the phase transition, where the correlation length will have become approximately

L	128	200	256	360
$\beta_c^x(L)$	0.438857(33)	0.439536(24)	0.439889(18)	0.440088(13)
χ_f^{max}	1409(6)	3308(14)	5233(24)	9910(49)
L	440	512	640	760
$\beta_c^x(L)$	0.440261(12)	0.440292(10)	0.440403(10)	0.440465(8)
χ_f^{max}	13138(71)	18912(99)	25215(218)	30841(206)

Table 3.1: Pseudo-critical points $\beta_c^x(L)$ which are obtained from the maximum value of the neural network susceptibility χ_f^{max} on systems of lattice size L .

equal to the system's lattice size $\xi \approx L$, we have:

$$|t| = \left| \frac{\beta_c - \beta_c(L)}{\beta_c} \right| \sim \xi^{-\frac{1}{\nu}} \sim L^{-\frac{1}{\nu}}. \quad (3.11)$$

Based on the above relation we are able to simultaneously calculate the critical inverse temperature β_c and the correlation length exponent ν . In addition, based on the values of the maxima of the susceptibility we are able to calculate the magnetic susceptibility exponent γ/ν , via the equation:

$$\chi_m \equiv \delta P \sim L^{\gamma/\nu}. \quad (3.12)$$

The calculation of the critical inverse temperature β_c and the correlation length exponent γ/ν is then achieved by fitting the pseudo-critical points $\beta_c^x(L)$ which were determined by the maxima of the susceptibility χ_f^{max} of the neural network function, whereas the calculation of the magnetic susceptibility exponent γ/ν is achieved by fitting directly χ_f^{max} . Both calculations are based on the above equations. The values of the pseudo-critical points, as well as the maxima of the susceptibility are given in Table 3.1. Given the data we then conduct the finite-size scaling analysis, which is depicted in Fig. 3.8.

The numerical results from the finite-size scaling analysis are presented in Table 3.2, where they are compared with the exact values from Onsager's analytical solution of the two-dimensional Ising model. We observe that the results obtained from the neural network function f and its susceptibility χ_f are highly accurate and overlap within statistical errors with the exact values. We emphasize that the calculation of statistical errors has been conducted with a bootstrap technique by resampling the dataset 1000 times. In the calculations, no systematic errors are considered from the training of the machine learning algorithms. The results therefore indicate that machine learning can be a powerful tool for precision studies of phase transitions

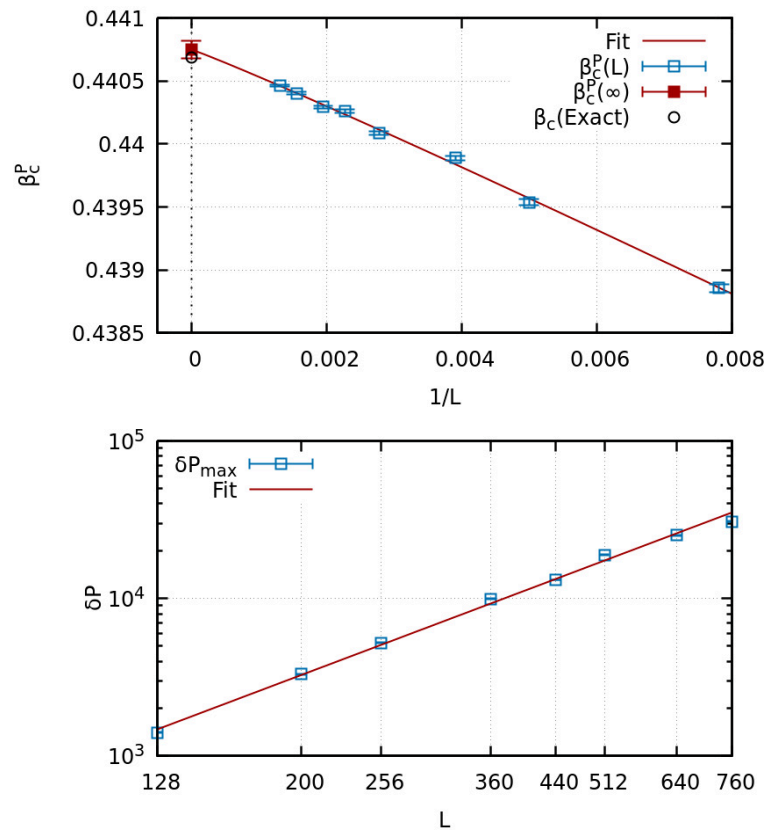


Figure 3.8: The finite size scaling analysis for the pseudo-critical points β_c^P versus the inverse lattice size $1/L$ (top) and the maxima of the neural network susceptibility δP versus the lattice size L (bottom).

	β_c	ν	γ/ν
CNN+Reweighting	0.440749(68)	0.95(9)	1.78(4)
Exact	$\ln(1 + \sqrt{2})/2 \approx 0.440687$	1	$7/4 = 1.75$

Table 3.2: Calculation of the critical exponents $\nu, \gamma/\nu$ and the critical inverse temperature β_c of the Ising model via the finite-size-scaling analysis.

since it enables the extraction of multiple critical exponents and the critical inverse temperature.

3.5 Discussion

In this chapter we demonstrated that functions derived from machine learning algorithms can be interpreted as statistical-mechanical observables by being associated with a corresponding Boltzmann weight. As a result, machine learning functions can be extrapolated in a system’s parameter space with histogram reweighting, and we are therefore able to extend the classification capabilities of neural networks by obtaining machine learning predictions in cases where data are not available. In addition, we showed that a neural network function, trained to separate phases in the two-dimensional Ising model, manifests the behaviour of an effective order parameter. We utilized this neural network function to conduct a precision study of a phase transition via the extraction of multiple critical exponents and the critical inverse temperature in the two-dimensional Ising model.

In summary, via the interpretation of machine learning functions as statistical-mechanical observables the complete spectrum of statistical mechanical techniques can be applied to such functions. As a result, efficient studies of physical systems can be achieved by enhancing machine learning with computational techniques from statistical physics, such as histogram reweighting. Before delving deeper into the interpretability of these functions from the perspective of physics we will first investigate further applications. Specifically, in the next chapter we aim to answer the question: what happens if we present as input to an Ising-trained convolutional neural network configurations of a different system that undergoes a phase transition? Will the neural network accurately separate phases in different systems? And can we discover phase transitions if we do not know that a phase transition exists in a different system?

Chapter 4

Discovering phase transitions with machine learning

4.1 Introduction

In the previous chapter we demonstrated that by training a convolutional neural network on a set of configurations σ of the Ising model we are able to study the system's phase transition. This has been achieved by using the neural network function $f(\sigma_i)$, which provides the probability that a configuration σ_i belongs in the broken-symmetry phase. At this point, one can pose the question: what happens if we present as input to this Ising-trained convolutional neural network a set of configurations σ' which correspond to a different system that might undergo a different type of phase transition? The above question can be explored via the framework of transfer learning [39]. Previous research, which has focused on discrete-spin systems, has utilized the method of transfer learning to determine quantities such as the critical inverse temperature [15, 40, 41].

In this chapter, we will investigate if a machine learning algorithm, trained to separate the phases of the Ising model, can be utilized to predict the phase structure of systems that undergo phase transitions of different order or universality class [42]. This means phase transitions which are described by entirely different critical behaviour. For this reason we must establish a process to guarantee that the obtained results are accurate. In addition we will explore if the Ising-trained machine learning algorithm can provide correct phase diagrams even when applied to systems with discrete but non-binary degrees of freedom or continuous degrees of freedom. The aim of this chapter is to establish if simple systems, such as the Ising model, can be used to obtain the phase diagram of more complicated systems, therefore opening up

the opportunity to discover unknown phase transitions in complicated systems with intricate phase structures by utilizing neural network functions learned on simple ones.

4.2 Multiple histogram reweighting

Before investigating whether transfer learning is possible between distinct phase transitions we will first introduce a different type of reweighting for machine learning functions, called the multiple histogram method [43]. This method allows us to scan large regions of a system's parameter space, hence making it easier to discover a phase transition in a system in which we do not know if a phase transition exists. Multiple histogram reweighting is an extension of the single histogram reweighting approach that was discussed in the previous chapter. Despite the fact that the method is a conceptual generalization of the single histogram technique it is established based on different principles and it should therefore be viewed as a different technique. Here, we follow the derivations and the perspective discussed in Ref. [10].

With the multiple histogram reweighting technique we aim to combine a set of Monte Carlo simulations, conducted at inverse temperatures $\beta_1, \beta_2, \dots, \beta_n$ with $\beta_1 < \beta_2 < \dots < \beta_n$, to accurately calculate expectation values of observables within the entire continuous parameter range between $[\beta_1, \beta_n]$. Consequently, one can interpolate the neural network function f in the entire range defined above to locate the critical region. In addition, since a large number of Monte Carlo simulations are optimally combined to obtain results, the inclusion of an additional Monte Carlo dataset always leads to a reduction of statistical errors in the calculation of expectation values.

To derive the multiple histogram reweighting equations we will first start by defining the probability $p(E)$ of sampling a certain value of the energy E in relation to the density of states $\rho(E)$:

$$p(E) = \rho(E) \frac{\exp[-\beta E]}{Z}. \quad (4.1)$$

All quantities have been defined before, except the density of states $\rho(E)$, which counts the number of configurations of the system that have a specific value of energy E . In addition, the partition function Z can be expressed in terms of the density of states as:

$$Z = \sum_E \rho(E) \exp[-\beta E]. \quad (4.2)$$

When conducting a Markov chain Monte Carlo simulation we are able to estimate

the probability $p(E)$ of an energy E via:

$$p(E) = \frac{N(E)}{n}, \quad (4.3)$$

where n corresponds to the number of statistically independent measurements that we have obtained. We remark that while the histograms of the energy $N(E)$, which are constructed based on a predefined bin size, appear in the derivations we will express the final relations without the use of histograms. We can now substitute Eq. (4.3) to Eq. (4.1) and consider that we have conducted a number of different Markov chain Monte Carlo simulations i, \dots, j for a specific inverse temperature β . We can then obtain an estimate for the density of states from each of these simulations, given by

$$\rho_i(E) = \frac{N_i(E)Z_i}{n_i \exp[-\beta E]}. \quad (4.4)$$

The question is how to optimally combine each of these different estimations ρ_i to estimate the actual density of states $\rho(E)$. We then express the density of states $\rho(E)$ as a weighted average in terms of all the estimations $\rho_i(E)$ as follows:

$$\rho(E) = \sum_i w_i \rho_i(E), \quad (4.5)$$

where w_i are the weights. The weights can be obtained via a minimization of the variance related to the density of states [10], thus arriving at the expression:

$$\rho(E) = \frac{\sum_i N_i(E)}{\sum_j n_j Z_j^{-1} \exp[-\beta_j E]}, \quad (4.6)$$

where the sums are over the number of the obtained Markov chain Monte Carlo simulations. The partition function Z_m , which corresponds to a certain inverse temperature β_m , is then calculated via:

$$Z_m = \sum_E \rho(E) \exp[-\beta_m E]. \quad (4.7)$$

The partition function can then be estimated via an iterative scheme [10], or other appropriate forms of optimization, through the relation:

$$Z_m = \sum_{i,s} \frac{1}{\sum_j n_j Z_j^{-1} \exp[(\beta_m - \beta_j) E_{is}]}, \quad (4.8)$$

where s is a sum over the configurations obtained at a specific simulation i .

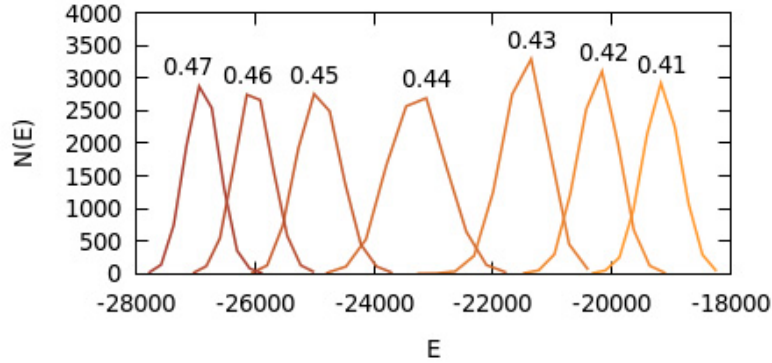


Figure 4.1: Histograms $N(E)$ versus uniquely sampled values of the energy E for a range of inverse temperatures $\beta = 0.41, \dots, 0.47$. The overlap of histograms enables the use of the multiple histogram reweighting technique.

Based on the above we are now able to combine multiple Markov chain Monte Carlo simulations to estimate the partition function for the inverse temperature β_m . For instance, after the optimization approach has converged we are able to calculate the partition function Z_l for a specific inverse temperature β_l which lies anywhere between the interpolated range defined by $[\beta_i, \beta_n]$ of the Markov chain Monte Carlo datasets that we used in the method. As mentioned before, knowledge of the partition function Z implies knowledge of any observable O , and we are therefore able to calculate the expectation value of an arbitrary observable $\langle O \rangle_l$ at a specific interpolated inverse temperature β_l via:

$$\langle O \rangle_l = \frac{1}{Z_l} \sum_{i,s} \frac{O_{is}}{\sum_j n_j Z_j^{-1} \exp[(\beta_l - \beta_j) E_{is}]}. \quad (4.9)$$

The multiple histogram technique is arguably more complicated than the single histogram reweighting technique presented in the previous chapter. As with any other type of reweighting, the multiple histogram method is successful only when there exists an overlap of the histograms of the energies between each inverse temperature of the Markov chain Monte Carlo simulations used to establish the method. To illustrate the concept in the case of the two-dimensional Ising model, an overlap of histograms is depicted in Fig. 4.1. One could implement the multiple histogram method based on the Monte Carlo datasets depicted in the figure to estimate any partition function in the entire region of inverse temperatures defined by $[0.41, 0.47]$ and, consequently, any observable O of interest. In this chapter we are therefore interested in combining multiple Markov chain Monte Carlo simulations to interpolate the neural network

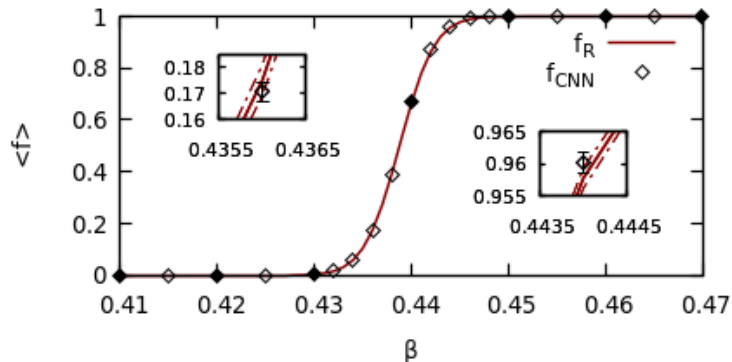


Figure 4.2: Expectation value of the neural network function f versus inverse temperature β . The filled points correspond to the Monte Carlo datasets used to conduct multiple histogram reweighting, thus enabling the interpolation of the neural network function in the entire range $\beta \in [0.41, 0.47]$, shown by the red line. The empty points correspond to independent machine learning predictions, obtained on separate Monte Carlo datasets to allow for a direct comparison with the reweighted result. Excluding the insets, the statistical errors are comparable with the width of the lines.

function f in the parameter space of a statistical system and to obtain the phase diagram via the Ising-trained convolutional neural network. We remark that even when systems with continuous energy spectra are considered, one arrives at identical equations for the multiple histogram method as in the discrete case. In addition, the method straightforwardly extends to systems with Hamiltonians or actions that include multiple terms.

Before proceeding we will verify that multiple histogram reweighting produces the correct result when applied on the interpolation of the expectation value of a neural network function f learned on the Ising model and which is applied on configurations of the Ising model with lattice size $L = 128$ in each dimension. To clarify, for this example we remain in the setting of the previous chapter. The result can be seen in Fig. 4.2, where the set of Monte Carlo simulations that we used to conduct multiple histogram reweighting are depicted by the filled points. The interpolation of the neural network function f in the range $\beta \in [0.41, 0.47]$ is depicted by the red line, and the results are compared with independent calculations, shown as empty points. We observe that the results overlap within statistical errors, demonstrating that multiple histogram reweighting of neural network functions is successful. We are now interested in obtaining analogous results in different systems than the Ising model but by still utilizing a neural network function f that was trained exclusively on configurations

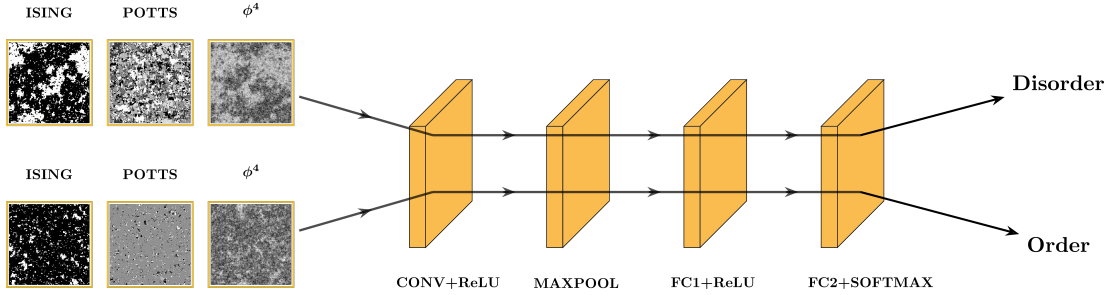


Figure 4.3: Configurations from systems with different degrees of freedom and/or distinct types of phase transitions are presented as input to an Ising-trained neural network to predict their corresponding phase.

of the Ising model.

4.3 Transfer learning

There exist multiple types of transfer learning within the research field of computer science [39]. Here, we will focus on a type of domain adaptation, where the already trained function $f(\sigma_i)$ will be utilized to successfully predict the phase of configurations σ'_i which comprise a different domain, i.e., the phase of configurations of a different system. We remark that the method is anticipated to predict only phases of a similar nature such as in the original system, namely order-disorder phase transitions, since the function remains the same. An important difference with commonly applied methods of transfer learning, is that in this work we will not re-train the machine learning algorithm on a set of different configurations but we will directly apply the function f to predict phases of distinct systems.

To investigate the accuracy of transfer learning from the Ising model, we will predict the phases of q -state Potts models as well as the ϕ^4 scalar field theory. The q -state Potts models possess different critical behaviour from the Ising model when $q \geq 3$. Specifically, when $q = 3$ or $q = 4$ the phase transitions are of second-order and of a different universality class, and when $q \geq 5$ the phase transition is first-order. We recall that another difference of the $q \geq 3$ Potts models from the Ising models is that Potts models have discrete degrees of freedom in the range $1, \dots, q$, hence complicating the transfer learning procedure. Because the Potts Hamiltonian comprises a delta function we can arbitrarily replace the degrees of freedom with unique values in the range $[-1, 1]$, without affecting the physics of the system.

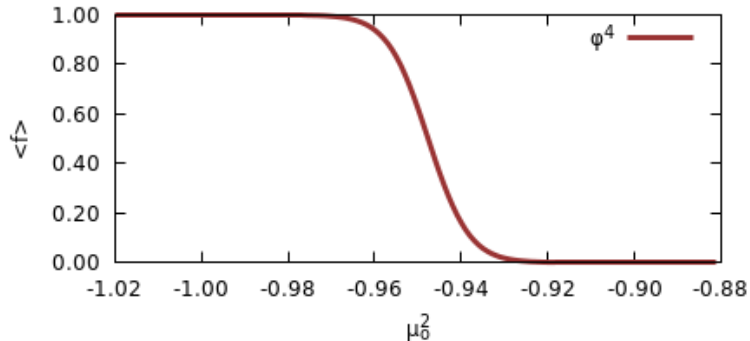


Figure 4.4: Expectation value of the Ising-trained neural network function f versus value of the squared mass $\mu^2 \equiv \mu_o^2$ for the case of the ϕ^4 scalar field theory with $L = 128$. The results have been obtained with the use of multiple-histogram reweighting. The statistical errors are comparable with the width of the lines.

Another system in which we will apply transfer learning from the Ising model is the ϕ^4 scalar field theory. Under an appropriate choice of coupling constants the system undergoes a second-order phase transition which is conjectured to be in the Ising universality class. However the difference of the ϕ^4 theory with the Ising model is that the degrees of freedom are continuous. We are therefore interested in observing if a neural network function f , constructed on a system with binary degrees of freedom, is able to provide accurate results when configurations with continuous degrees of freedom are presented as input. No rescaling to the degrees of freedom of the ϕ^4 theory will be conducted in the following results. We remark that, in this chapter, we express the dimensionless squared mass as μ^2 instead of μ_L^2 to avoid confusion with the pseudo-critical squared masses $\mu_c^2(L)$ which will be discussed below. We will now extend the ideas discussed in the previous chapter to discover the phase transitions of q -state Potts models and the ϕ^4 scalar field theory. This will be achieved based on a neural network that has been trained exclusively on configurations of the Ising model, see Fig. 4.3 for a summary of what we aim to achieve.

We therefore present as input to the Ising-trained neural network a set of configurations from the ϕ^4 theory and implement the multiple histogram reweighting method to interpolate f in parameter space. The results are depicted in Fig. 4.4. We observe that the neural network function f has indicated the location of the crossing of a phase transition, since there are certain regions of parameter space that correspond to a broken-symmetry phase and other regions of parameter space that correspond to a symmetric phase. We remark that, for the chosen values of the coupling constants

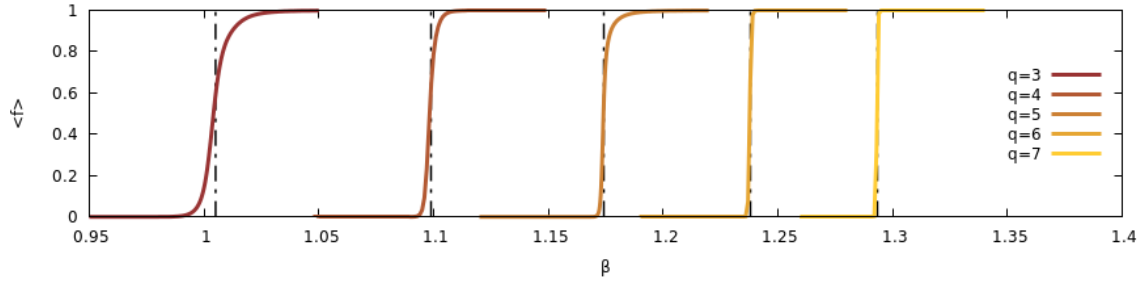


Figure 4.5: Expectation value of the Ising-trained neural network function f versus inverse temperature β for the case of the Potts models with $L = 128$. The results are obtained with the use of multiple-histogram reweighting and the dashed vertical lines correspond to the analytical value of the critical point for each q -state Potts model. The statistical errors are comparable with the width of the lines.

$\mu^2 < 0$, $\lambda = 0.7$, $\kappa = 1$, the critical point of the phase transition is anticipated to be $\mu_c^2 = -0.95151(25)$ [44], $\mu_c^2 = -0.9516(8)$ [45]. Transfer learning is therefore successful in indicating the location of the critical region for the phase transition of the system. This has been achieved despite the fact that the ϕ^4 theory is a system with continuous degrees of freedom and the neural network was trained exclusively on the Ising model, a system with binary degrees of freedom.

To explore if transfer learning can additionally be applied on systems which are described by different critical behaviour, we now extend the method to the case of the q -state Potts models. The results, obtained with the use of multiple histogram reweighting on the Ising-trained neural network function, are depicted in Fig. 4.5. They are additionally compared with the analytical value of the critical inverse temperature β_c which is shown, for each of the q -state Potts models, as a dashed vertical line. We observe that a phase transition is located for all of the cases of the Potts models, even when the universality class or the order of the phase transition is different from the one of the Ising model. The results therefore indicate that transfer learning is a useful tool in locating phase transitions for more complicated systems, for instance Potts models, using simple systems such as the Ising model. As a result, the method could be useful in discovering unknown phase transitions via the reconstruction of effective order parameters in a target system's parameter space.

4.3.1 Critical exponents of the ϕ^4 theory

We remark, that even though the prior results are highly accurate, they should still be treated as approximations in relation to the study of the thermodynamic limit. The reason is that there exists no guarantee that the effective order parameter, constructed from the Ising-trained neural network on a system with different critical behaviour such as the $q = 3$ Potts model, will manifest the proper scaling behavior. In fact, one generally expects that since the underlying critical behaviour is different the reconstructed effective order parameter can only provide qualitative observations. However, this potential problem can be evaded: after identifying the location of the critical region, and therefore becoming aware of which regions of parameter space correspond to the symmetric or the broken-symmetry phases, we can train a new neural network to study the phase transition of the target system. Specifically, we can create a dataset on a set of configurations which would be classified, based on Figs. 4.4 and 4.5, as being in the broken-symmetry or the symmetric phase and then study the thermodynamic limit exactly in the same manner as we did in the previous chapter for the Ising model. Through this approach, we remove any bias or inaccuracy introduced by an Ising-trained neural network function applied on a system described by different critical behaviour.

We will now illustrate how to conduct a finite-size scaling analysis based on the new neural network function f' to extract critical exponents for the ϕ^4 scalar field theory. Having obtained knowledge of the critical region for the ϕ^4 theory via Fig. 4.4 we now train a new neural network for configurations in the range $\mu^2 \leq -1.0$ and $\mu^2 \geq -0.90$. We emphasize that, to avoid misclassification, the training set comprises configurations which are not immediately adjacent to the critical point and the training of the neural network is stopped after observing no evolution in the predicted values of the test set for a large number of epochs. The architecture used is identical to the one implemented for the Ising model. For the case of the ϕ^4 theory the critical coupling constant is the value of the squared mass and the reduced coupling constant is therefore expressed as:

$$\left| \frac{\mu_c^2(L) - \mu_c^2}{\mu_c^2} \right| \sim \xi^{-\frac{1}{\nu}} \sim L^{-\frac{1}{\nu}}. \quad (4.10)$$

Another difference is in the definition of the susceptibility, in which we do not introduce the equivalent of an inverse temperature, i.e.:

$$\chi_f = V(\langle f^2 \rangle - \langle f \rangle^2), \quad (4.11)$$

where $V = L \times L$ is the size of the system.

L	$\mu_c^2(L)$	$\chi_{f'}$
200	-0.94988(4)	8239(50)
256	-0.95037(5)	12915(56)
360	-0.95096(4)	22348(138)
440	-0.95117(3)	34710(211)

Table 4.1: The pseudo-critical points $\mu_c^2(L)$ as determined from the maximum values of the neural network susceptibility $\chi_{f'}$ for lattice size L .

Based on these relations we study the thermodynamic limit of the system following the same procedure as for the Ising model. Explicitly, we associate to the maxima of the susceptibility $\chi_{f'}$ of the new neural network function f' a pseudo-critical squared mass, and obtain the calculations in the infinite volume limit via a finite-size scaling analysis. The calculation of the critical exponents and the critical point is conducted based on the results of Table 4.1 and is shown in Fig. 4.6, where scaling can be evidently observed based on the bottom panel. The values of the critical exponents are depicted in Table 4.2. We observe that the critical exponents agree with the exponents of the Ising universality class, therefore providing evidence that the second-order phase transition of the two-dimensional ϕ^4 scalar field theory is identical to that of the two-dimensional Ising model.

4.3.2 Searching for universal structures

The next step forward is to obtain some insights into the interpretability of the prior results. Since the machine learning algorithm is able to discern between phases of systems that possess, in essence, different critical behaviour, we can investigate if there is some form of universal structure that has emerged on the dependencies that the neural network has learned. Within the research field of computer science, it is a well-established fact that neural networks learn a set of universally applicable features in the first layers [46]. These features are the sets of weights and biases that have been learned at each layer. The subsequent layers of a neural network architecture then comprise specialized features which are tuned to be efficient for the considered

	μ_c^2	ν	γ/ν
CNN+Reweighting	-0.95225(54)	0.99(34)	1.78(7)

Table 4.2: The value of the critical squared mass μ_c^2 and the critical exponents ν , γ/ν of the ϕ^4 scalar field theory.

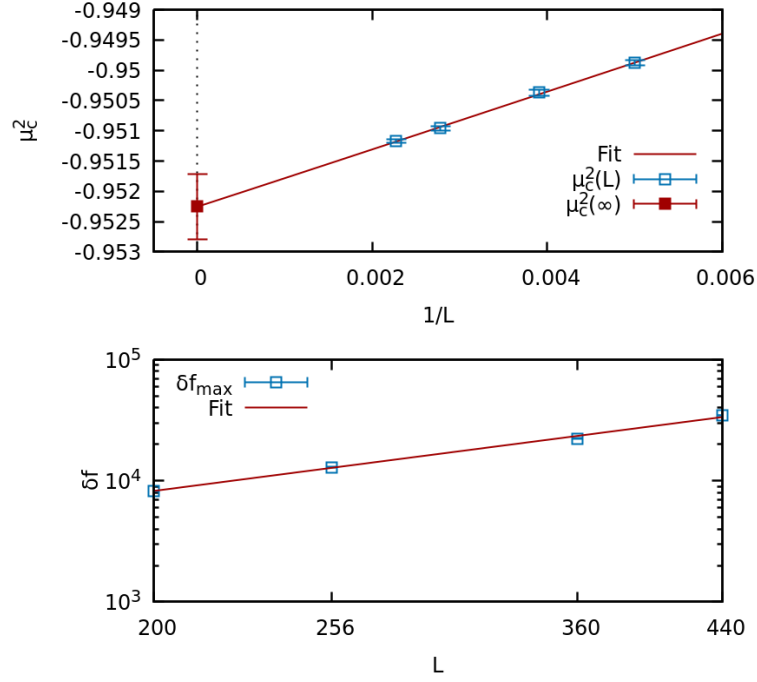


Figure 4.6: Finite size-scaling analysis for the value of the pseudo-critical mass μ_c^2 versus the inverse lattice size $1/L$ (top) and the susceptibility of the Ising-trained neural network function f versus the lattice size L .

machine learning task. Specifically, the machine learning task discussed here is the separation of phases in the two-dimensional Ising model.

In the current chapter we observe that the neural network which has been trained on the Ising model can accurately predict the symmetric and the broken-symmetry phases for all of the considered systems, such as the q -state Potts models and the ϕ^4 theory. So we expect that within the neural network, there exist some form of universal features that should extend further in the deeper layers, since we are able to obtain the correct result for all of the aforementioned systems. To investigate for the existence of universal features, we then monitor the output of intermediate layers when the neural network is presented with configurations of distinct systems. To clarify, consider the neural network architecture depicted in Fig. 4.3. A set of configurations is presented as input to the neural network. At each of the included layers these configurations get iteratively processed by a corresponding function. The output from the function of each layer is then presented as input to the next layer, which gets processed again by the corresponding function of the next layer. Eventually, a

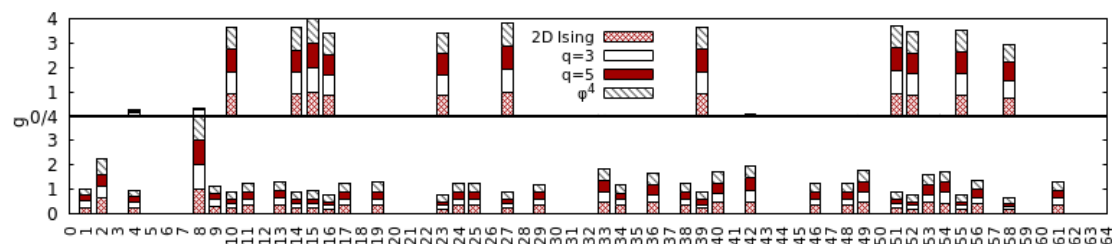


Figure 4.7: Mean activation functions g versus the 64 variables present in the fully connected layer of the Ising-trained convolutional neural network. In the top figure the activations are shown when configurations from the disordered (symmetric) phase of the systems are presented as input and in the bottom figure the activations are shown when configurations from the ordered (broken-symmetry) phase are presented as input. The results have been rescaled to one, and stacked vertically for easier comparison.

set of iterative mappings processes the data until we reach the output layer where we obtain the prediction of the phase of a configuration. Here, we are interested in exploring if any consistency of results can be discovered within one of the intermediate layers of the neural network architecture rather than the final layer.

The results, obtained by monitoring the output of the 64 variables on the first fully connected layer, are depicted in Fig. 4.7. We observe that there is a certain set of variables that get triggered when configurations from the ordered phase are presented as input, and a different set of variables when configurations from the disordered phase are given as input. This observation holds irrespective of the system, therefore indicating that the neural network has learned some form of universal structure for the ordered and disordered phase which remains accurate even when configurations from different systems are given as input. This emergent universal structure throughout the neural network is not generally anticipated since, based on empirical results in computer science, deep neural networks tend to learn universal features only in the initial layers, and generally require retraining in order to be successfully applied to different problems.

4.4 Discovery of phase transitions: a summary

Here, we will summarize through a series of steps how one can employ machine learning techniques to discover a phase transition in a complicated system by relying on a machine learning function that was learned on a simple system.

First, one obtains a set of configurations from Markov chain Monte Carlo simulations that belong in distinct phases of an original system to construct a labeled training dataset. A neural network is then trained on the dataset to learn a neural network function f that is able to accurately separate configurations from each phase of the original system.

Second, the neural network function f of the original system is applied to configurations of a target system to observe if distinct phases are discovered. Large regions of the target system's parameter space can then be scanned with the use of multiple histogram reweighting to locate the critical region.

Third, having obtained the knowledge of the critical region of the target system, the original neural network with function f is not needed anymore and it is therefore discarded. A new neural network is then trained on configurations from the discovered phases of the target system to learn a novel neural network function f' . We remark that one could retrain the original neural network instead of discarding it, but this was avoided in the current work, since there is no guarantee that the converged state of the original neural network, namely the set of learned weights and biases, is an optimal initial state for subsequent training of the neural network.

Finally, by relying on the new neural network function f' and its susceptibility $\chi_{f'}$ one calculates the critical exponents and the critical point of the target system by relying on a finite-size scaling analysis.

By following the above steps one is able to discover a phase transition while avoiding potential inaccuracies that can be introduced by using identical neural network functions on systems which might possess different critical behaviour.

4.5 Discussion

In this chapter we demonstrated that neural network functions learned on simple systems, such as the Ising model, can be utilized to predict the phase structure of more complicated systems, such as the q -state Potts models and the ϕ^4 scalar field theory. This is achieved even when the order or the universality class of the phase transition in the target system differs from the one in the original system. In addition, we introduced the multiple histogram reweighting method to interpolate the neural network function and hence scan large regions of a system's parameter space to discover a phase transition. Finally, given the knowledge of the phase structure of a target system, we calculated multiple critical exponents and the critical point for the ϕ^4 scalar field theory using machine learning functions. A related manuscript that appeared after the current work is Ref. [47].

In summary, the use of transfer learning enables the reconstruction of effective order parameters in target systems and hence opens up the opportunity to discover unknown phase transitions. So far, our discussion has focused on applications based on the neural network function, an observable that has been calculated directly on some configurations. In the next chapter, we will take a further step in interpreting this neural network function by introducing it as a term within the Hamiltonian of the system. Following the analogous approach of introducing an external field in a system, such as the one related to the magnetization, and hence force the system to interact with an external parameter, we will demonstrate that the same can be achieved with a fictitious field coupled to the neural network function. Mathematically, and having expressed the neural network field in relation to the system's partition function, this is equivalent to asking the question of what happens if one allows a statistical-mechanical system to interact with a neural network that has been trained to accurately separate its phases. The implications that will emerge from this perspective is what we will investigate next.

Chapter 5

Neural networks as Hamiltonian terms

5.1 Introduction

In the preceding chapters we explored how the physical interpretation of machine learning functions can lead to certain practical implications. Explicitly, we demonstrated that by interpreting a neural network function as a statistical-mechanical observable we are able to extrapolate it in a system's parameter space with the use of histogram reweighting. Furthermore, we utilized an Ising-trained neural network function f , which acts as an effective order parameter, to predict the phase structure of systems even when the degrees of freedom are non-binary and even when the order or the universality class of the phase transition differs from the one of the Ising model. Here, we will proceed a step further in physically interpreting machine learning functions.

In this chapter, we will introduce a neural network as a term within the Hamiltonian of a system [48, 49]. To the best of knowledge, no other work besides the one discussed in this thesis has ever explored the introduction of machine learning functions as physical terms within Hamiltonians. We aim to investigate the behaviour of a system under the constraint that it interacts with a fictitious field which is coupled to a neural network function. Specifically, we will first explore if the inclusion of the neural network function f in the Hamiltonian can induce an analogous phase transition such as the one induced by the conventional order parameter, which is the magnetization. We remark that we have previously established that the neural network function f acts as an effective order parameter and we have calculated via infinite-volume limit calculations that its susceptibility is governed by the critical ex-

ponent of the magnetic susceptibility. As a result, we are able to investigate if the neural network function and its fictitious field can produce analogous behaviour to that of the magnetization and of the magnetic external field.

The second aim of this chapter is to investigate if we can obtain critical exponents that were previously inaccessible with the use of neural network functions. Consequently, we aim to explore if an exponent could ever be derived for the case of a fictitious neural network field. Here, we will not utilize a finite-size scaling analysis to extract the critical exponents but instead we will rely on the renormalization group [50–55] and, specifically, on its computational aspects [56–65]. The use of the real-space renormalization group provides certain benefits, such as the reduction of finite-size effects in calculations related to the thermodynamic limit. This is due to the fact that only two systems of identical lattice size are required to conduct calculations. The method therefore opens up the possibility to obtain critical exponents on much smaller lattice sizes than what is generally expected.

5.2 Conjugate variables and external fields

To include the neural network function f as a physical term within a Hamiltonian, we turn to the fundamentals of statistical physics. The partition function or, equivalently, the free energy of a system encode all of the statistical information that is required to describe the system in completeness, since every observable of interest can be derived in terms of the partition function. To express a measurable quantity, which is physically interpretable and can be calculated on the configurations of a system as a statistical-mechanical observable we must therefore be able to express it in terms of the system's partition function. What we will discuss below is exactly the mathematical procedure that one would follow to enable a statistical system to interact with an external field, such as the magnetic field, or any other type of constraint that we decide to impose on a statistical system.

We will now focus on exploring how the system is affected if we introduce a neural network function as a term within the Hamiltonian. We remark that within Hamiltonians, only extensive quantities can be introduced. The neural network function f is interpreted as the probability that a configuration is in the broken-symmetry phase and it is therefore bound between $[0, 1]$. As a result the neural network function f is an intensive property since it does not have the proper dependence on the size of the system. To recast f as an extensive property is then as simple as multiplying it by the size V of the system.

To be able to introduce the neural network function f as a term within the Hamiltonian we must additionally couple it to a fictitious external field Y . In statistical

physics, parameters, constraints, or fields that interact with a system have conjugate variables which represent the response of the system to the perturbation of the corresponding parameter. Examples of such conjugate variables are the magnetization or the volume of the system and the associated external fields or constraints are the magnetic external field or the pressure, respectively. By varying the external magnetic field one can then observe the implications in the conjugate variable which is the magnetization. As a result, in our study here we are interested in investigating the behaviour of the neural network function f as a conjugate variable, by varying its associated fictitious field Y .

Having expressed the neural network function as an extensive property Vf , and having coupled it to a fictitious field Y , we are now able to define a modified Hamiltonian for the two-dimensional Ising model which introduces the neural network VfY as a physical term:

$$E_Y = E - VfY. \quad (5.1)$$

If the neural network field is zero $Y = 0$ we obtain the original Hamiltonian of the Ising model. By taking the derivative of the logarithm of the partition function Z_Y in relation to the external field Y we are able to obtain an expression that we recognize as the expectation value $\langle f \rangle$ of the neural network function f :

$$\langle f \rangle = \frac{1}{\beta V} \frac{\partial \ln Z_Y}{\partial Y} = \frac{\sum_{\sigma} f_{\sigma} \exp[-\beta E_{\sigma} + \beta V f_{\sigma} Y]}{\sum_{\sigma} \exp[-\beta E_{\sigma} + \beta V f_{\sigma} Y]}. \quad (5.2)$$

If the neural network field is zero then we obtain the original expression of the expectation value for the neural network function f . A second derivative in terms of the neural network field Y produces the following expression:

$$\chi_f = \frac{\partial \langle f \rangle}{\partial Y} = \beta V (\langle f^2 \rangle - \langle f \rangle^2). \quad (5.3)$$

The quantity χ_f introduced above is the susceptibility of the neural network function which measures the response of the neural network function f to changes in the neural network field Y . We have encountered the susceptibility of the neural network function f in the previous chapter but here we obtained χ_f mathematically in terms of the system's partition function. A natural question that then emerges from the above perspective is what are the effects on the system when the fictitious field has a nonzero value $Y \neq 0$. Equivalently what are the effects when the system is allowed to interact with the introduced neural network term. This is the topic of the current chapter.

5.3 Hamiltonian-agnostic reweighting

Before we proceed to study the effect of a nonzero external field $Y \neq 0$ to the two-dimensional Ising model we will introduce another variation of the histogram reweighting method. So far, we have used histogram reweighting to calculate expectation values of observables by extrapolating to different values of inverse temperatures in the Ising model or to different values of the squared mass in the ϕ^4 theory. Here, we are interested in a different setting. Specifically, starting from configurations of the two-dimensional Ising model that have been sampled at a specific inverse temperature β , we are interested in obtaining expectation values of observables for extrapolations of a nonzero external field while the system remains at the specific inverse temperature β .

We remark that, in principle, one can simultaneously extrapolate on both the inverse temperature β and a nonzero external field Y . However this is generally discouraged, since such extrapolations are established on a trajectory within a two-dimensional parameter space, defined simultaneously by β and Y . Consequently, when relying on extrapolations in a high-dimensional space the results will not be as accurate unless certain precautions are taken in relation to guaranteeing an overlap of histograms. For this reason in the current work we are strictly interested in extrapolating only for nonzero values of Y while the inverse temperature β remains fixed.

To introduce the histogram reweighting approach that allows extrapolations to a nonzero external field Y , we consider the expectation value $\langle O \rangle$ of the arbitrary observable that we aim to sample, in the modified system of Hamiltonian E_Y :

$$\langle O \rangle = \frac{\sum_{i=1}^N O_{\sigma_i} \tilde{p}_{\sigma_i}^{-1} \exp[-\beta E_{\sigma_i} + \beta V f_{\sigma_i} Y]}{\sum_{i=1}^N \tilde{p}_{\sigma_i}^{-1} \exp[-\beta E_{\sigma_i} + \beta V f_{\sigma_i} Y]}. \quad (5.4)$$

We will now choose \tilde{p}_{σ_i} as the probability distribution of the original system which remains at the same inverse temperature β as the modified system, thus obtaining:

$$\langle O \rangle = \frac{\sum_{i=1}^N O_{\sigma_i} \exp[\beta V f_{\sigma_i} Y]}{\sum_{i=1}^N \exp[\beta V f_{\sigma_i} Y]}. \quad (5.5)$$

This form of reweighting enables the calculation of an expectation value $\langle O \rangle$ for a nonzero value of the external field Y , based on some configurations which are obtained via Markov chain Monte Carlo simulations on the original system at a specific inverse temperature β and with zero Y . An important observation about the above reweighting equation is that it is Hamiltonian-agnostic. The value, or the form, of the

Hamiltonian does not appear in this relation. This implies that one can apply this type of reweighting using only a set of configurations, the knowledge of the inverse temperature β , and a trained neural network. As a result, one does not need to know what is the Hamiltonian or the action that produced a certain set of configurations.

5.4 Neural network-induced phase transitions

We will now investigate the behaviour of the two-dimensional Ising model with $L = 64$ for a nonzero neural network field, where for this chapter we consider a function learned on a fully-connected neural network. We remark that, when the neural network field is nonzero, one would generally need to simulate with Markov chain Monte Carlo simulations the system with the modified Hamiltonian to obtain a set of configurations. However, due to the Hamiltonian-agnostic reweighting approach introduced in the previous section, we can evade the previous problem. We can therefore obtain expectation values of observables for the nonzero field case $Y \neq 0$ using only configurations sampled based on the original Hamiltonian of the system at a specific inverse temperature β and without a neural network field $Y = 0$.

We will now apply Hamiltonian-agnostic reweighting to investigate the behaviour of the system when the neural network external field is nonzero. We will conduct this study for three different values of inverse temperature $\beta = 0.43, 0.440687, 0.45$ which define a system below, at, and above the critical point, respectively. The results can be seen in Fig. 5.1. We observe that, irrespective of the phase that the system is positioned in, by varying the neural network field Y the system is able to transition between the symmetric and the broken-symmetry phases. We recall that the neural network function f is the probability that a configuration belongs in the broken-symmetry phase so for values $f \approx 1$ the system resides in the broken-symmetry phase and for values $f \approx 0$ it resides in the symmetric phase. As a result the neural network field is able to induce a phase transition in the system.

We observe that the neural network-induced phase transition, which occurs based on the neural network field Y , differs from the phase transition that is induced by an external magnetic field h . In the case of the external magnetic field h , when $h > 0$ ($h < 0$) the system transitions in the broken-symmetry phase and the spins are positively (negatively) aligned. Equivalently, the external magnetic field always induces explicit symmetry-breaking in the system. As a result, the system is always driven to a broken-symmetry phase, irrespective of the sign of h , and it is therefore unable to transition back to the symmetric phase by restoring the symmetry of the system. However this is not the behaviour that is observed for the case of the neural network field Y . We observe that one is able to transition between both the symmetric

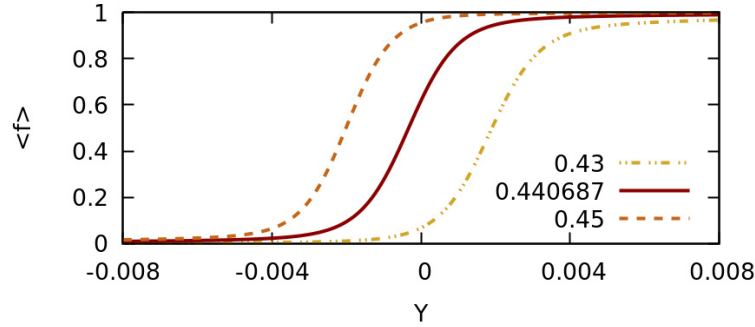


Figure 5.1: Expectation value of the neural network function f versus the neural network field Y for values of inverse temperature $\beta = 0.43, 0.440687, 0.45$ below, exactly at, and above the critical inverse temperature β_c , respectively. The statistical errors are comparable with the width of each line.

and the broken-symmetry phases by varying Y . As a result one is able to both break and restore the symmetry of the system with a neural network field Y .

In fact, the observed behaviour is easy to explain. The neural network function f is the probability that a configuration belongs in the broken-symmetry phase, and it is therefore a quantity that satisfies positivity. This is in contrast with the conventional order parameter, namely the magnetization, which can be both positive and negative. The sign of the introduced neural network term within the Hamiltonian then depends entirely on the sign of the neural network field Y . We recall that the original Hamiltonian of the Ising model includes a term $-\sum_{\langle ij \rangle} \sigma_i \sigma_j$. As a result, and based on the fact that the system favors states with smaller energy, when a positive or negative neural network term is introduced in the system the spins will compensate by aligning towards a ferromagnetic or a disordered state. Consequently, a phase transition between a symmetric and a broken-symmetry phase can be induced by the neural network field Y .

By observing that the neural network field Y is able to induce a phase transition in the two-dimensional Ising model we can then investigate how the susceptibility χ_f of the neural network function is affected. The results are shown in Fig. 5.2 for nonzero values of the neural network field Y . We observe that, irrespective of the initial phase of the system, there exist maxima for the susceptibility, therefore indicating the crossing of a phase transition. We recall that we used similar arguments to study the phase transition of the two-dimensional Ising model and the ϕ^4 scalar field theory in the preceding chapters. These arguments could be extended to the case discussed here, and one could therefore proceed in studying the induced phase transition using

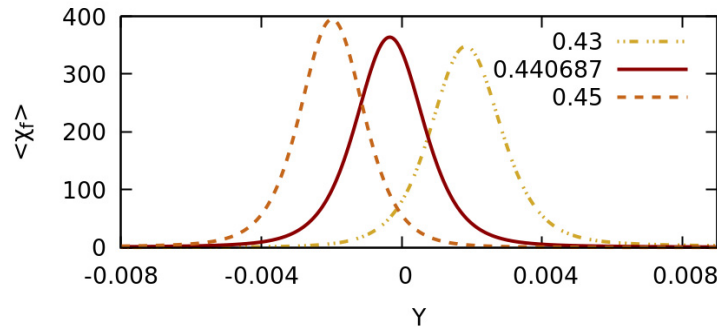


Figure 5.2: Expectation value of the susceptibility χ_f of the neural network function f versus the neural network field Y . The statistical errors are comparable with the width of each line.

a finite size scaling analysis. However, in the subsequent chapters we will focus on a different method to study phase transitions: the real-space renormalization group.

5.5 The renormalization group

5.5.1 Fundamentals and the transformation

In this section we will discuss the real-space renormalization group in the context of phase transitions. The method relies on the application of a transformation that iteratively eliminates degrees of freedom within a system. When applying a renormalization group transformation we must devise a set of rules to produce each rescaled degree of freedom for a system. These rules must respect certain properties.

In the Ising model the devised rule should produce degrees of freedom which remain binary, and thus construct a rescaled system which resembles an Ising model. Our aim is to devise a set of rules that will best preserve the large-scale information of the system. To choose the rescaled degrees of freedom we use the majority rule, see Fig. 5.3. Specifically, we split the system into blocks of size $b \times b$, and we then choose the rescaled degree of freedom based on the majority of the spins within each block. When the degrees of freedom are equal, we choose the rescaled degree of freedom randomly as $+1$ or -1 . The majority rule is well-established in studies of the phase transition of the Ising model.

We begin by observing that the application of a renormalization group transformation on a system of lattice size L in each dimension will produce a rescaled system

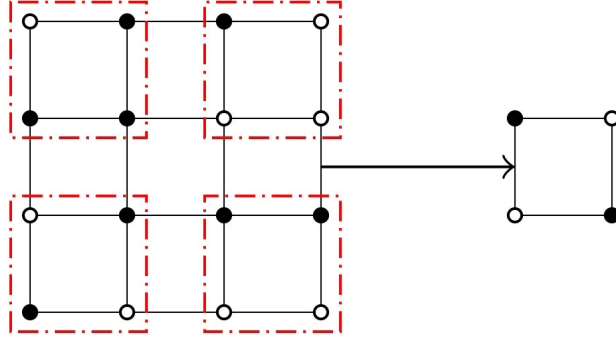


Figure 5.3: A blocking transformation with a rescaling factor of $b = 2$ and the majority rule. When the number of positive $+1$ and negative -1 degrees of freedom within each block is equal the rescaled degree of freedom is chosen randomly.

of lattice size

$$L' = \frac{L}{b}, \quad (5.6)$$

where b is the rescaling factor. We will retain b in the derivations even though in this thesis we always consider $b = 2$.

Now consider that we apply a renormalization group transformation on a configuration σ_i of the two-dimensional Ising model which has been drawn from the equilibrium distribution of an inverse temperature β in the vicinity of the phase transition $\beta \approx \beta_c$. This configuration σ_i therefore encodes a certain correlation length ξ . The renormalization group transformation preserves the large-scale information of the original system and since it reduces the original lattice size by a factor of b in each dimension it should also reduce the original correlation length by the same factor. Consequently, the rescaled correlation length ξ' is given by

$$\xi' = \frac{\xi}{b}. \quad (5.7)$$

The first crucial result follows from an observation that we discussed extensively before, namely that the correlation length ξ is a quantity which depends on the inverse temperature $\xi(\beta)$. The correlation length increases as we approach the critical point $\beta \approx \beta_c$ and it diverges exactly at the critical inverse temperature β_c . Since the original and the rescaled systems have different correlation lengths ξ and ξ' then, by definition, they are associated to different inverse temperatures β and β' . As a result the two systems are additionally described by different observables O and O' , where in this chapter we will work with intensive observables.

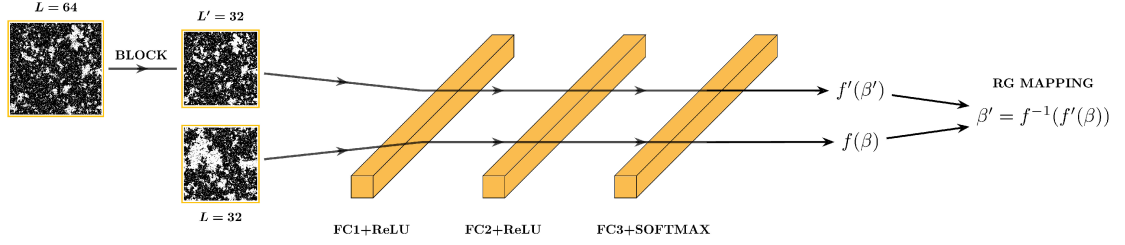


Figure 5.4: The fully-connected neural network architecture. A renormalization group mapping between a rescaled system with lattice size $L' = 32$ and an original system with lattice size $L = 32$ is constructed via the neural network function f .

The second crucial result then follows explicitly from the observation that the correlation length ξ diverges exactly at $\beta = \beta_c$, and thus we are able to obtain a self-consistent approach to locate the critical point. Specifically, the critical point is the point in parameter space where $\beta = \beta' = \beta_c$ and the correlation length becomes infinite (or zero). As a result the intensive observables of the original and the rescaled systems become equal $O(\beta_c) = O'(\beta_c)$. In other words, to discover the critical point one needs only search for an equality between two observables, one in the original system and one in the rescaled system. We remark though that the method is affected by finite-size effects and, consequently, not all observables are expected to intersect. However, as we will discuss below the impact of finite-size effects is still minimal compared to other traditional methods, such as finite-size scaling. Here, we will utilize as observables the neural network function f of the original system and the neural network function f' of the rescaled system. In the critical point we therefore have:

$$f(\beta_c) = f'(\beta_c). \quad (5.8)$$

We will start from the above equation to locate the critical fixed point of the two-dimensional Ising model.

5.5.2 Flows and the critical fixed point

In this section, we will again utilize histogram reweighting but with a major difference: we are now interested in extrapolating observable quantities O' of the rescaled system by relying exclusively on the original system's probability distribution p , and therefore on the original system's Hamiltonian or action E . We recall that to each original

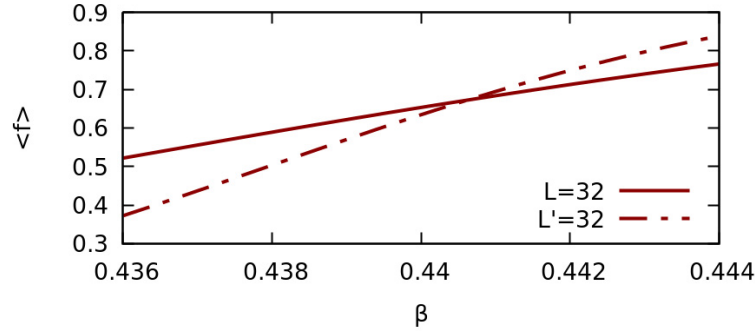


Figure 5.5: Expectation value of the neural network function f and f' for an original and a rescaled system of lattice size $L = L' = 32$ versus the inverse temperature β .

configuration σ_i , which has been drawn from an equilibrium distribution $p(\sigma_i)$, is associated via a renormalization group transformation a rescaled configuration σ'_i with a different probability distribution $p'(\sigma'_i)$. The two probability distributions are different because the rescaled inverse temperature β' and the rescaled lattice size L' have changed during the application of a transformation on an original system. However, there exists a mapping between each σ_i and σ'_i via the renormalization group transformation and, as a result, observables O' of the rescaled system remain, in a probabilistic manner, as observables of the original system. They can therefore be reweighted in parameter space using the original probability distribution $p(\sigma_i)$. Practically, this implies that to reweight a rescaled observable in the original system's parameter space one can simply replace every occurrence of O_{σ_i} with $O'_{\sigma'_i}$ in Eq. (5.5). Specifically:

$$\langle O' \rangle = \frac{\sum_{i=1}^N O'_{\sigma'_i} \exp[\beta V f_{\sigma_i} Y]}{\sum_{i=1}^N \exp[\beta V f_{\sigma_i} Y]}. \quad (5.9)$$

The expectation values of the original and the rescaled neural network functions f and f' at an identical lattice size $L = L' = 32$, as obtained through the use of the histogram reweighting approach, are depicted in Fig. 5.5. We observe that there exists an intersection point in parameter space for the neural network functions of the original and the rescaled system. Based on our previous discussion, which led to the introduction of Eq. (5.8), this point corresponds to the critical inverse temperature β_c of the system, in which we expect a divergence of the correlation lengths ξ and ξ' , and therefore an equivalence between intensive observables of the original and the rescaled system. We emphasize that one generally expects to observe an equivalence of observables only after a larger number of iterative renormalization

group transformations. However, for the case of the Ising model, intersection of multiple observables can be observed even after one step of the renormalization group, see Ref. [10]. We treat this estimation of the critical fixed point as a qualitative result and in the subsequent discussion we will instead obtain a quantitative estimation.

We previously discussed that a renormalization group transformation leads to a reduction of the correlation length as $\xi' = \xi/b$, where $b = 2$, and therefore the rescaled system is described by a different inverse temperature β' . This implies that if we start from an inverse temperature β that is below the critical point $\beta < \beta_c$ then iterative applications of renormalization group transformations will drive the system towards the zero inverse temperature, and therefore to the symmetric phase (complete disorder). Conversely, if we start with configurations above the critical inverse temperature $\beta > \beta_c$, then consecutive applications will drive the system towards the infinite inverse temperature $\beta = \infty$ and therefore to the broken-symmetry phase (complete order). This observation relates to the concept of a renormalization group flow in a system's parameter space, and can be observed in Fig. 5.5. We recall that the neural network function f expresses the probability that the system resides in the broken-symmetry phase. Consequently, we observe that for values below the critical point $f' < f$, indicating that the rescaled system has been driven towards the symmetric phase. Conversely, above the critical point $f' > f$, and the system has been driven towards the broken-symmetry phase.

5.5.3 The relevant operators

Having established the concepts of the renormalization group in a qualitative manner we can now shift focus and instead conduct a quantitative study of the phase transition. Let us first observe that in Fig. 5.5 we are able to select a value of the original neural network function f , which corresponds to a certain inverse temperature β , and then associate to it a rescaled neural network function so that $f = f'$, while $\beta \neq \beta'$. Based on this observation we can generalize Eq. (5.8) as follows:

$$f(\beta') = f'(\beta). \quad (5.10)$$

The equation above hints that one might be able to derive an expression that can directly relate the two inverse temperatures β and β' . It then follows, straightforwardly, that this expression can be obtained by the inverse mapping:

$$\beta' = f^{-1}(f'(\beta)). \quad (5.11)$$

We now recall the definition of the reduced inverse temperature t which measures the distance from the critical point β_c . As we have clarified by now, the original and

the rescaled systems are described by different inverse temperatures β and β' and, as a result, they have different distances t and t' from the critical point β_c . Consequently, the correlation length of the original system diverges based on the relation

$$\xi \sim |t|^{-\nu}, \quad (5.12)$$

while the correlation length of the rescaled system diverges as

$$\xi' \sim |t'|^{-\nu}. \quad (5.13)$$

We remark that the phase transition in both systems is identical, and therefore described by the same set of critical exponents, since both the original and the rescaled systems are two-dimensional Ising models. By dividing the relations for the divergence of the two correlation lengths we obtain:

$$\left(\frac{t}{t'}\right)^{-\nu} = b. \quad (5.14)$$

To conduct Monte Carlo renormalization group calculations, only one final step is required, namely to linearize the renormalization group mapping in the vicinity of the phase transition. We will achieve this, as commonly done in statistical physics, via a Taylor expansion to leading order [66], thus obtaining:

$$\beta_c - \beta' = (\beta_c - \beta) \left. \frac{d\beta'}{d\beta} \right|_{\beta_c}, \quad (5.15)$$

where the notation $|_{\beta_c}$ denotes a calculation in the vicinity of the phase transition. We remark that, due to the linearization, calculations are accurate even when not conducted exactly at the critical point β_c . By substituting the above equation into Eq. (5.14) and taking the natural logarithm we obtain an expression for the calculation of the correlation length exponent ν :

$$\nu = \frac{\ln b}{\ln \left. \frac{d\beta'}{d\beta} \right|_{\beta_c}}. \quad (5.16)$$

We are now able to conduct our first quantitative calculation of a critical exponent and of the critical fixed point with a Monte Carlo renormalization group method. First, we construct the mappings given by Eq. (5.10) which are depicted in Fig. 5.6. The critical point of the renormalization group transformation can then be obtained at the intersection of the two lines, from which we obtain the value $\beta_c = 0.44063(21)$. In addition we calculate the correlation length exponent based on the results depicted

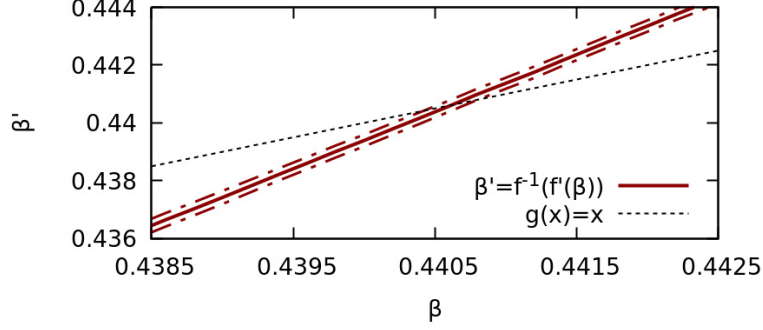


Figure 5.6: Rescaled inverse temperature β' versus inverse temperature β . The dashed lines, parallel to the solid line, indicate the statistical errors. The intersection with the line $g(x) = x$ corresponds to $\beta = \beta' = \beta_c$ and therefore is the obtained estimation of the value of the critical point via the renormalization group.

in the same figure. Specifically via the use of numerical derivatives in the vicinity of the phase transition we obtain the value $\nu = 1.01(2)$. We observe that the value of the correlation length exponent is highly accurate even though the calculation has been conducted on an original and rescaled system of small lattice size $L = L' = 32$. In fact, the accuracy of the results supersedes the calculations conducted with finite-size scaling in the previous chapters, where larger lattice sizes were utilized.

We have now extracted the correlation length exponent ν , which is related to one of the relevant operators of the renormalization group for the two-dimensional Ising model. Obtaining the critical exponents related to the relevant operators is important since all other critical exponents can be calculated directly based on the exponents related to the relevant operators with the use of scaling relations. In the case of the two-dimensional Ising model, discussed here, there exists another relevant operator, namely the critical exponent that governs the divergence of the correlation length for the external magnetic field h , as $h \rightarrow 0$ and $\beta = \beta_c$. We will now investigate if we are able to extract this exponent using the neural network field Y , instead of the external field h . This could be justified from our studies in the previous chapters where we demonstrated that the neural network function f acts as an effective order parameter. So the neural network field might manifest the same scaling behaviour as the external field of the conventional order parameter, which is the magnetization.

We start our investigation by introducing a critical exponent θ_Y that governs the divergence of the correlation length ξ

$$\xi \sim |Y|^{-\theta_Y}. \quad (5.17)$$

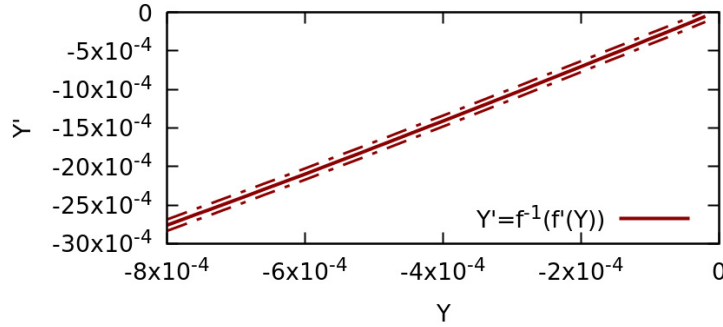


Figure 5.7: Rescaled neural network field Y' versus original neural network field Y . The original simulation was conducted exactly at the critical point $\beta_c = 0.440687$.

It is now possible to pursue exactly the same discussion as we did for the inverse temperature, and construct a mapping that relates the neural network fields Y and Y' of the original and the rescaled systems, respectively, as:

$$Y' = f^{-1}(f'(Y)). \quad (5.18)$$

We can then derive the equation that allows us to calculate numerically the neural network field exponent θ_Y as

$$\theta_Y = \frac{\ln b}{\ln \left. \frac{dY'}{dY} \right|_{Y=0}}. \quad (5.19)$$

The mappings constructed for the case of the neural network fields are depicted in Fig. 5.7. We calculate the critical exponent θ_Y via numerical derivatives on the data depicted in the figure and obtain the value $\theta_Y = 0.534(3)$. We recall that the two-dimensional Ising model is exactly solvable and the analytical values of the two critical exponents related to the relevant operators that govern the divergence of the correlation length are $\nu = 1.0$ for the exponent related to the phase transition induced by the inverse temperature and $\theta = 8/15$ for the exponent related to the phase transition induced by the external magnetic field, when $h \rightarrow 0$ and $\beta = \beta_c$. We observe that the neural network field exponent θ_Y overlaps within statistical errors with the magnetic external field exponent θ , therefore indicating that the two operators are identical.

In summary, by coupling the neural network function f to a fictitious external field Y and introducing it within the Hamiltonian of the two-dimensional Ising model, we were able to obtain the two critical exponents related to the relevant operators of the

renormalization group transformation and the critical point of the two-dimensional Ising model on systems with lattice size as small as $L = L' = 32$. We remark that the renormalization group is a powerful tool to study phase transitions and can, for instance, provide accurate results using systems simulated on small lattice sizes. One of the reasons for this efficiency is that the infinite-volume limit calculation is obtained using measurements conducted on only two systems, hence limiting finite-size effects in comparison to calculations, such as finite-size scaling, which rely on measurements obtained on multiple systems of different lattice sizes.

5.6 Discussion

In this chapter we demonstrated that neural network functions f can be included as physical terms within Hamiltonians by being coupled to a fictitious field Y . We were then able to express quantities related to the machine learning function, such as the expectation value and the susceptibility, as derivatives of the Ising model's partition function in terms of the neural network field Y . Using Hamiltonian-agnostic reweighting, we observed that the neural network field Y can induce a phase transition in the two-dimensional Ising model by breaking or restoring its symmetry. This is in contrast with the phase transition induced by the field associated with the system's conventional order parameter, which breaks the symmetry explicitly. We then introduced a renormalization group approach that enabled the calculation of the two critical exponents related to the relevant operators of the Ising model, as well as its critical point, by utilizing exclusively functions derived from machine learning algorithms. Finally, we discussed the induced renormalization group flows in the system's parameter space.

In summary, via the inclusion of neural network functions as physical terms within Hamiltonians, a new method to induce phase transitions in systems was introduced. Consequently, the opportunity to induce analogous phase transitions via neural network fields in systems where conventional order parameters are absent or unknown is open to explore. Our study related to neural network functions f , which are constructed to separate phases in systems, ends at this point of the thesis. Our focus will now shift to quantum field theories and the use of machine learning to generate states in absence of the critical slowing down effect via the construction of inverse renormalization group transformations.

Chapter 6

Inverse renormalization group

6.1 Introduction

We will now extend the discussion pertinent to the real-space renormalization group by exploring the construction of inverse renormalization group transformations. Only a minimal number of studies have been conducted in relation to the implementation of the inverse renormalization group. These have appeared exclusively within the context of statistical physics and are established on spin systems with discrete degrees of freedom [56, 59, 67–69]. To the best of knowledge, no inverse Monte Carlo renormalization group method has ever been explored in the context of quantum field theory or to a system with continuous degrees of freedom, outside of the current work.

In this chapter, we will implement machine learning algorithms to construct inverse renormalization group transformations [70]. Specifically, we will investigate if a set of inverse transformations can be learned that is able to mimic the inversion of a standard renormalization group transformation. Our aim is to utilize these inverse transformations to iteratively increase the size of a system in absence of the critical slowing down effect, therefore obtaining configurations for systems with larger lattice size without having to simulate them with Markov chain Monte Carlo simulations. We will additionally explore if inverse renormalization group transformations give rise to inverse flows in a system's parameter space, therefore driving it towards its critical point. Finally, we will investigate if the inverse renormalization group which can, in principle, be applied for an arbitrary number of steps to increase the size of a system, can be utilized to extract multiple critical exponents for the phase transition of the ϕ^4 scalar field theory.

6.2 RG flows in the ϕ^4 theory

Before learning a set of transformations that are able to invert a standard renormalization group transformation we must first verify that the renormalization group method, as described in the previous chapter, is accurate when applied to the ϕ^4 scalar field theory. The coupling constant K which induces the phase transition in the ϕ^4 theory is the squared mass $K \equiv \mu^2$ and the reduced coupling constant t then measures the distance from the critical point K_c :

$$t = \frac{K_c - K}{K_c}. \quad (6.1)$$

Following exactly the discussion related to the previous chapter, intensive observable quantities O and O' for an original and a rescaled system will then intersect exactly at the critical point K_c :

$$O(K_c) = O'(K_c), \quad (6.2)$$

thus providing a self-consistent method to locate K_c .

The ϕ^4 scalar field theory is a system with continuous degrees of freedom and, in contrast with the renormalization group study of the Ising model in the previous chapter, we will not rely on a majority rule to define each rescaled degree of freedom. Instead the renormalization group transformation that we will implement works as follows. We will again separate the lattice into blocks of size $b \times b$, and sum the degrees of freedom within each block. If the sum is positive (negative), the rescaled degree of freedom is chosen as the mean of the positive (negative) degrees of freedom within the block.

We will now verify that the renormalization group transformation described above is accurate on the ϕ^4 scalar field theory. We consider configurations of a ϕ^4 scalar field theory of lattice size $L = 32$ that has been sampled in the vicinity of the phase transition, specifically for values $\kappa = 1$, $\mu^2 = -0.9515$, $\lambda = 0.7$ and apply the above transformation to obtain a system of lattice size $L' = 16$. We then implement histogram reweighting to extrapolate the expectation value of the original and the rescaled magnetizations m and m' , for different values of the squared mass μ^2 .

The results are depicted in Fig. 6.1. We observe that the standard renormalization group flows that emerged are analogous to the case of the two-dimensional Ising model. Specifically, below the critical point $\mu^2 < \mu_c^2$ the rescaled system has larger values of the magnetization $m' > m$ since it has been driven towards the broken-symmetry phase. Conversely, above the critical point $\mu^2 > \mu_c^2$ the rescaled system has smaller values of magnetization $m' < m$ since it has been driven towards the

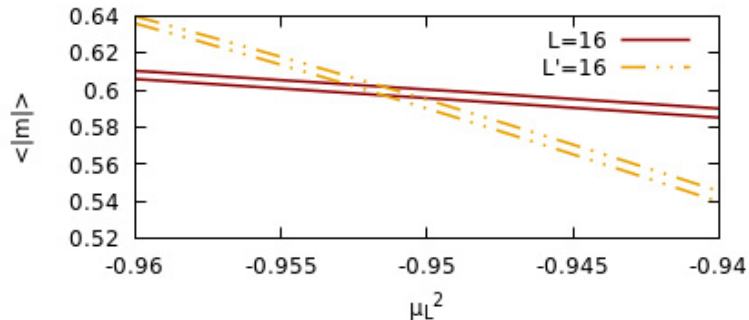


Figure 6.1: Expectation value of the magnetization $|m|$ versus the squared mass μ^2 for an original and a rescaled system of lattice size $L = L' = 16$ in each dimension.

symmetric phase. In addition the original and the rescaled intensive magnetizations intersect, therefore evidencing that a critical fixed point of the renormalization group transformation has emerged.

We have hence verified that the standard renormalization group transformation described above is accurate when applied to the ϕ^4 scalar field theory. Specifically, it produces the anticipated renormalization group flows in the system's parameter space and it has additionally provided a critical fixed point μ_c^2 . By inverting this standard renormalization group transformation successfully with the use of machine learning algorithms, we expect that the inverse transformation will additionally satisfy all of the conditions mentioned above. These are the emergence of a critical fixed point and the associated inverse renormalization group flows in the system's parameter space. We will now focus on the inversion of such a transformation and we will investigate its accuracy via calculations pertinent to the continuum limit.

6.3 Inverting a transformation

We are now interested in constructing an inverse renormalization group transformation. In contrast to the standard renormalization group, which iteratively eliminates degrees of freedom within a system and therefore reduces the system's lattice size, an inverse renormalization group transformation will introduce degrees of freedom and hence produce a rescaled system that is described by an increased lattice size. We recall that in this thesis we always consider that the rescaling factor $b = 2$ and therefore an inverse renormalization group transformation will double the lattice size of an original system in each dimension.

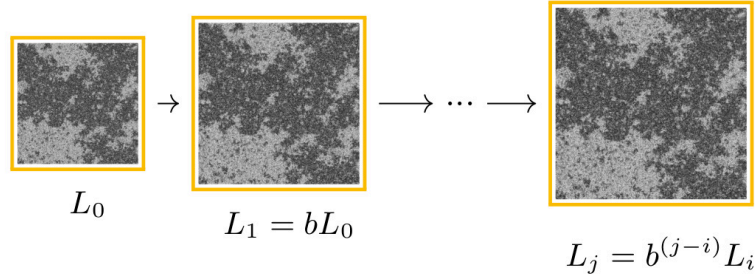


Figure 6.2: Illustration of the inverse renormalization group method. Starting from uncorrelated configurations of a minimally sized lattice L_0 we apply iteratively the inverse transformations to arbitrarily increase the size of the system.

The question is now how to construct such an inverse renormalization group transformation. In fact, one is able to devise any transformation, via the application of a function, that will increase the lattice size of the system by a rescaling factor of b and then one can investigate if the transformation is accurate, for instance on a prototypical system. This research direction is appealing, since it allows for complete interpretability of the obtained results. However it is simultaneously expected to be inefficient, based on the current knowledge of the inverse renormalization group, since there exists no guidance as to what constitutes a successful inverse transformation.

Here, we will follow a different approach: we will treat the construction of the inverse renormalization group transformation as an optimization problem that aims to invert, through the mathematical operation of transposed convolution, the application of a standard renormalization group transformation. Specifically, we start from configurations of an original system that has lattice size $L = 32$ and apply a standard renormalization group transformation to obtain a rescaled system of lattice size $L' = 16$. We then apply a set of transposed convolutions on $L' = 16$, see Chapter 2, to produce configurations for a model system described by lattice size $L_m = 32$. Our aim is now to minimize a loss function, that is able to establish an equivalence between the degrees of freedom of the model system with $L_m = 32$ and the degrees of freedom of the original system with $L = 32$. If we consider a certain degree of freedom this can be achieved via the minimization of a mean squared error function:

$$\text{MSE}(\phi_i, \phi_i^{(m)}, \theta) = (\phi_i - \phi_i^{(m)}(\theta))^2, \quad (6.3)$$

where ϕ_i is an original degree of freedom and $\phi_i^{(m)}(\theta)$ is the corresponding degree of freedom of the model system that has a dependence on the set of variational parameters θ .

The benefit of this approach is that it instantly guarantees that the obtained inverse transformations will be of comparable efficiency and accuracy as the standard renormalization group transformation that was selected for inversion. The reason is that, through this approach, one can certify that they are able to reconstruct the original system from a rescaled one, and therefore one has confirmed that the inverse transformation will satisfy the required conditions that make the standard renormalization group successful, namely the encoded difference of the correlation length between the two systems. Practically, this additionally means that one is able to first verify that a standard renormalization group transformation is successful before the transformation gets inverted.

6.3.1 Inverse flows

For convenience, we briefly recall that an application of a standard renormalization group transformation on an original system of lattice size L in each dimension produces a rescaled system with lattice size L' as:

$$L \rightarrow L' = \frac{L}{b}. \quad (6.4)$$

By devising a set of transformations that are able to mimic the inversion of a standard renormalization group transformation we will be able to reconstruct the original system of lattice size L from the rescaled system of lattice size L' and thus obtain

$$L' \rightarrow L = bL'. \quad (6.5)$$

The important observation then follows from the realization that, once this set of inverse transformations is accurately learned, one is able to apply them consecutively to arbitrarily increase the size of the system as

$$L_j = b^{(j-i)} L_i, \quad (6.6)$$

where $j > i \geq 0$, and $L_0 = L$. Of course, as we have discussed in the previous chapter, if the initial configurations of the original system have been drawn from an equilibrium probability distribution in the vicinity of the system's phase transition then they encode a certain correlation length ξ . As a result the increase in the lattice size corresponds to an equal increase in the correlation length as:

$$\xi_j = b^{(j-i)} \xi_i, \quad (6.7)$$

with $\xi_0 = \xi$.

We will now investigate if the application of an inverse renormalization group transformation produces the anticipated behaviour. We start from a set of configurations of a ϕ^4 scalar field theory with lattice size $L_0 = 32$ that we have sampled in the vicinity of the phase transition. The set of inverse renormalization group transformations is then applied consecutively until we obtain a system of lattice size $L_4 = 512$, and we implement histogram reweighting to extrapolate the value of the magnetization m_j for all of the rescaled systems with lattice sizes L_1, L_2, L_3, L_4 . The results are shown in Fig. 6.3. where we have also included extrapolations of reweighting from original systems of the same lattice size as the rescaled ones, to enable a direct comparison.

We observe that the application of an inverse renormalization group transformation, depicted in Fig. 6.3, has produced the anticipated behaviour. Specifically, for values of the squared mass below the critical point $\mu^2 < \mu_c^2$, the rescaled magnetization has smaller values than the original magnetization $m' < m$ since the rescaled system has flowed towards the critical point, due to the increase of the correlation length. Conversely, when then system resides above the critical point $\mu^2 > \mu_c^2$, the rescaled magnetization has larger values than the original magnetization $m' > m$ as the rescaled system has again flowed towards the critical point because its correlation length has increased. As a result, inverse renormalization group flows have emerged in the system's parameter space.

In addition, we observe that the inverse renormalization group transformations can be utilized to locate the critical fixed point via the intersection of the original and the rescaled magnetization. This is exactly the behaviour that we expect from an inverse renormalization group transformation. We emphasize that simulating the original systems in Fig. 6.3 is not necessary: these results have been introduced to establish that the inverse renormalization group approach is a viable method. In fact, our aim is to avoid simulating the original systems, since a direct simulation of an original system in the vicinity of the phase transition is hindered by the critical slowing down effect.

The critical slowing down effect can be entirely avoided with the inverse renormalization group in calculations pertinent to the study of phase transitions. Starting from uncorrelated configurations the rescaled systems that we obtain with the method are sufficient to calculate multiple critical exponents of a system. Specifically, we achieve this by starting with configurations of a system at a small lattice size L_0 , for instance $L_0 = 32$, for which we are able to obtain uncorrelated measurements in an easy manner. We then apply the inverse transformations on the original system of lattice size L_0 to obtain configurations of rescaled systems with larger lattice sizes, such as

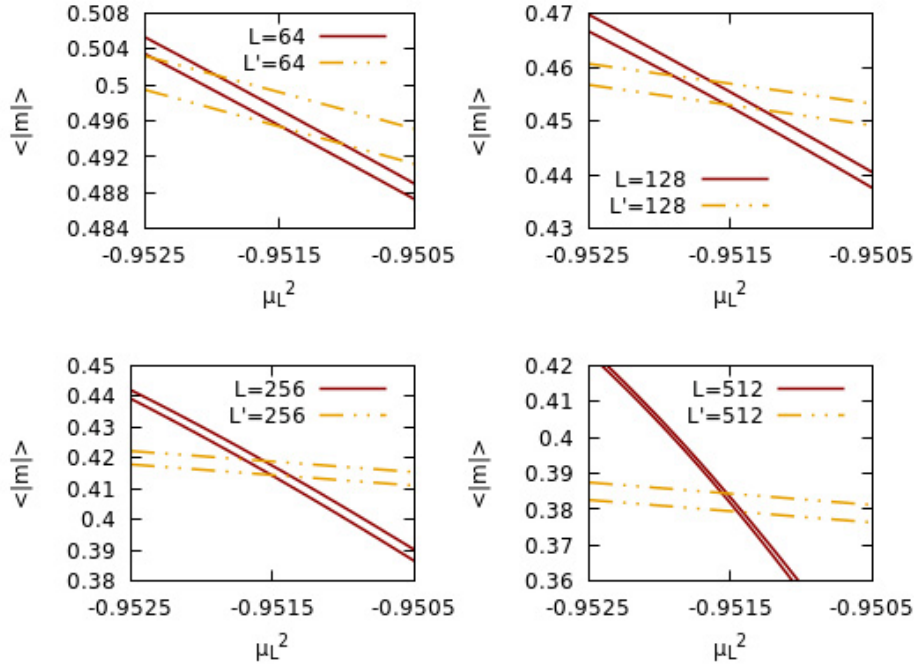


Figure 6.3: Expectation value of the magnetization m versus the value of the squared mass μ^2 .

$L_1 = 64, L_2 = 128, L_3 = 256, L_4 = 512$. Since we never have to simulate the systems at larger lattice sizes, we evade the critical slowing down effect, see also Ref. [59].

In addition, the current inverse renormalization group approach, discussed in this thesis, is further enhanced by the use of histogram reweighting, which enables the extrapolation of expectation values for observables of the rescaled systems. Specifically, starting from the original system of size $L_0 = 32$, we are not only able to obtain configurations of systems with larger lattice sizes $L_j > L_0$ but we are additionally able to obtain expectation values of observables in a large region of parameter space for all of the rescaled systems L_j through the use of histogram reweighting. This provides substantial computational benefits, because otherwise, one would need to conduct a large amount of computationally demanding simulations to obtain the same results of Fig. 6.3. Here through the inverse renormalization group method, combined with reweighting, we were able to obtain all of this information using only one simulation at lattice size $L_0 = 32$, conducted at one point in parameter space $\mu^2 = -0.9515$.

We have thus explored that the inverse renormalization group approach produces the anticipated behaviour, namely that it gives rise to inverse renormalization group

flows in parameter space and that it drives the system towards a critical fixed point. We are now interested in utilizing the inverse renormalization group method to calculate quantities in the infinite-volume limit, specifically to obtain critical exponents. Conceptually, we will follow the relevant discussion pertinent to the standard renormalization group in the previous chapter, but besides working with the inverse renormalization group, we are interested in obtaining a different set of critical exponents, namely those of the magnetization and of the magnetic susceptibility.

6.3.2 Extraction of critical exponents

We recall that during the application of a renormalization group transformation, the original and the rescaled systems have different distances t_i and t_j from the critical point. As a result the critical behaviour of the magnetizations m_i and m_j of the original and the rescaled system is described according to the relations:

$$m_i \sim |t_i|^{\beta_m}, \quad (6.8)$$

$$m_j \sim |t_j|^{\beta_m}. \quad (6.9)$$

The above expressions can be equivalently expressed in terms of the correlation lengths ξ_i, ξ_j as

$$m_i \sim \xi_i^{-\beta_m/\nu}, \quad (6.10)$$

$$m_j \sim \xi_j^{-\beta_m/\nu}. \quad (6.11)$$

We remark that the magnetization critical exponent β_m as well as the correlation length exponent ν are the same in both relations since the original and the rescaled systems are both ϕ^4 scalar field theories. By dividing the two expressions and taking the natural logarithm we obtain:

$$\frac{\beta}{\nu} = -\frac{\ln \frac{m_j}{m_i}}{\ln \frac{\xi_j}{\xi_i}} = -\frac{\ln \frac{m_j}{m_i}}{(j-i) \ln b}. \quad (6.12)$$

We remark that the above expression holds only for an infinite system. However, we are interested in conducting calculations on systems of finite lattices. In line with the Taylor expansion conducted in the previous chapter to establish a linearization, we will now use l'Hôpital's rule, and obtain:

$$\frac{\beta}{\nu} = -\frac{\ln \left. \frac{dm_j}{dm_i} \right|_{K_c}}{\ln \frac{\xi_j}{\xi_i}} = -\frac{\ln \left. \frac{dm_j}{dm_i} \right|_{K_c}}{(j-i) \ln b}. \quad (6.13)$$

Using the expression above we are able to calculate the critical exponent β_m/ν via numerical derivatives in relation to the original and rescaled magnetizations.

We will now derive the expression for the calculation of the magnetic susceptibility exponent γ using the same arguments. Specifically, the divergence of the magnetic susceptibility is given via relations:

$$\chi_i \sim |t_i|^{-\gamma}, \quad (6.14)$$

$$\chi_j \sim |t_j|^{-\gamma}, \quad (6.15)$$

which are equivalently expressed in relation to the correlation lengths as

$$\chi_i \sim \xi_i^{\gamma/\nu}, \quad (6.16)$$

$$\chi_j \sim \xi_j^{\gamma/\nu}. \quad (6.17)$$

The relation for the calculation of the magnetic susceptibility exponent is then:

$$\frac{\gamma}{\nu} = \frac{\ln \left. \frac{d\chi_j}{d\chi_i} \right|_{K_c}}{\ln \frac{\xi_j}{\xi_i}} = \frac{\ln \left. \frac{d\chi_j}{d\chi_i} \right|_{K_c}}{(j-i) \ln b}. \quad (6.18)$$

To calculate the critical exponent γ/ν we require the values of the magnetic susceptibility, which are depicted for the original system $L_0 = 32$ and the rescaled systems $L_1 = 64, L_2 = 128, L_3 = 256, L_4 = 512$ in Fig. 6.4. The results have been obtained with the use of histogram reweighting. We recall that we have already calculated the values of the magnetization for the rescaled systems in Fig. 6.3 and for the original system in Fig. 6.1. Consequently, we are now able to proceed with the calculation of the critical exponents.

We emphasize that one is able to calculate the critical exponents not only by comparing an original and a rescaled system, but by comparing directly two rescaled systems. Specifically, instead of only calculating a critical exponent between the original system of lattice size L_0 and one of the rescaled systems $L_1 = 64, L_2 = 128, L_3 = 256, L_4 = 512$ one can instead use, for instance, L_2 and L_4 to obtain a calculation. This is possible because we have guaranteed that, by learning the inverse of a standard renormalization group transformation, each iteration of the transformation doubles the correlation length. Equivalently, we know by what factor the correlation length differs between any of the aforementioned systems. As a result Eqs. (6.13) and (6.18) are applicable to any combination of the systems described by lattice sizes L_0, L_1, L_2, L_3, L_4 .

Using all possible combinations of the aforementioned systems the calculation of critical exponents is depicted on Table 6.1. The original system with $L_0 = 32$

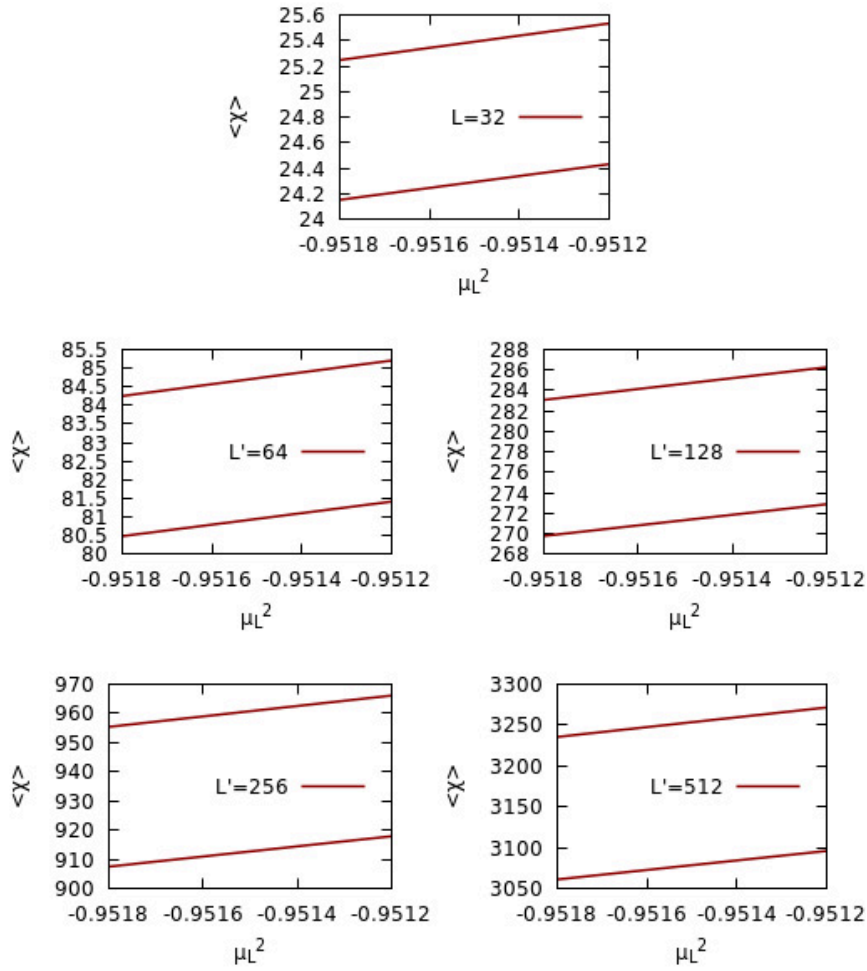


Figure 6.4: Expectation value of the magnetic susceptibility χ versus the squared mass μ^2 . The width of the lines indicates the statistical errors.

L_i/L_j	32/64	32/128	32/256	32/512	64/128
γ/ν	1.735(5)	1.738(5)	1.741(5)	1.742(5)	1.742(5)
β/ν	0.132(2)	0.130(2)	0.128(2)	0.128(2)	0.128(2)
L_i/L_j	64/256	64/512	128/256	128/512	256/512
γ/ν	1.744(5)	1.744(5)	1.745(5)	1.745(5)	1.746(5)
β/ν	0.127(2)	0.127(2)	0.126(2)	0.126(2)	0.126(2)

Table 6.1: The critical exponents γ/ν and β/ν . The original system is of lattice size $L_0 = 32$ in each dimension.

is sampled in the vicinity of the phase transition for values of coupling constants $\mu_L^2 = -0.9515$, $\lambda_L = 0.7$, $\kappa_L = 1$. We observe that there is a convergence of the critical exponents γ/ν and β_m/ν towards the values of the exponents that define the two-dimensional Ising universality class. In addition, this convergence is observed even for any combination of systems, either the original versus a rescaled system or between two rescaled systems.

To cross-verify the results we additionally calculate critical exponents by starting from a system of lattice size $L_0 = 8$, sampled at a different point in parameter space, specifically $\kappa_L = 1$, $\mu_L^2 = -1.2723$ and $\lambda_L = 1$. We remark that the choice of coupling constants defines a system that again resides in the vicinity of the phase transition, see Ref. [44]. We apply to the original system with lattice size $L_0 = 8$ the learned inverse transformations to obtain lattices of size up to $L_6 = 512$. The results are shown in Table 6.2, where again a convergence towards the Ising universality class is observed, irrespective of the choice of the initial system.

We remark that the inverse renormalization group method is anticipated to be applicable only when the original configurations encode a finite correlation length ξ . As a result, inconsistencies are anticipated to emerge in systems of smaller lattice size, a result that we have verified for $L_0 < 8$. This is due to the fact that the correlation length ξ is not properly encoded in systems of small lattice sizes, and the transformations can potentially produce larger systems that will not be representative of ϕ^4 scalar field theories.

6.4 Discussion

In this chapter we introduced the inverse renormalization group approach to quantum field theories and to systems with continuous degrees of freedom. We established an optimization approach to learn a set of transformations that are able to mimic

L_i/L_j	8/16	8/32	8/64	8/128	8/256	8/512	16/32
γ/ν	1.694(6)	1.708(6)	1.717(6)	1.723(6)	1.727(6)	1.730(6)	1.721(6)
β/ν	0.154(2)	0.147(2)	0.142(2)	0.139(2)	0.137(2)	0.135(2)	0.140(2)
L_i/L_j	16/64	16/128	16/256	16/512	32/64	32/128	32/256
γ/ν	1.728(6)	1.732(6)	1.735(6)	1.737(6)	1.735(6)	1.738(6)	1.740(6)
β/ν	0.136(2)	0.134(2)	0.132(2)	0.131(2)	0.133(2)	0.131(2)	0.130(2)
L_i/L_j	32/512	64/128	64/256	64/512	128/256	128/512	256/512
γ/ν	1.740(6)	1.741(6)	1.742(6)	1.742(7)	1.743(6)	1.743(7)	1.743(7)
β/ν	0.129(2)	0.129(2)	0.129(2)	0.128(2)	0.128(2)	0.127(2)	0.127(2)

Table 6.2: The critical exponents γ/ν and β/ν . The original system is of lattice size $L_0 = 8$ in each dimension.

the inversion of a standard renormalization group transformation. These inverse transformations can then be applied iteratively to arbitrarily increase the size of the system, in absence of the critical slowing down effect. We have further demonstrated that the application of an inverse renormalization group transformation gives rise to inverse flows in parameter space that drive a system closer to its critical fixed point, irrespective of the initial phase that the system resides in. In contrast to the standard renormalization group method, which eliminates degrees of freedom within a system and can be applied for a finite number of steps, the inverse renormalization group introduces degrees of freedom within a system and, in principle, can be applied for an arbitrary number of steps. Finally, we have utilized the inverse renormalization group method to calculate accurately multiple critical exponents for the two-dimensional ϕ^4 scalar field theory.

In summary, the current work provides the first implementation of the inverse Monte Carlo renormalization group, a method that is able to evade the critical slowing down effect, on quantum field theories and on systems with continuous degrees of freedom. Further exploration of the method might provide novel insights into the structure of the renormalization group, a method that emerges across diverse research fields such as condensed matter physics, quantum field theory and statistical mechanics. For the remainder of the thesis, we will focus on exploring fundamental connections between machine learning and quantum field theory. In the next chapter we will therefore explore the derivation of machine learning algorithms from quantum field theories.

Chapter 7

Quantum field-theoretic machine learning

7.1 Introduction

Up until this point we have emphasized applications of machine learning but here we will shift focus and investigate instead deeper connections that relate machine learning and physics. On that front, we will investigate probabilistic aspects of quantum field theory. These aspects share connections with the research field of machine learning and are directly accessible through the framework of lattice field theory.

In this chapter, we will derive machine learning algorithms from lattice field theories [71–73]. Specifically, we will establish an equivalence between the ϕ^4 scalar field theory on a square lattice and the framework of Markov random fields. Markov fields are a certain type of machine learning algorithms with applications in research fields such as computer vision or biology [74, 75]. In addition, Markov fields emerge in mathematical physics, specifically in constructive quantum field theory [2], where one utilizes the Markov property on Euclidean fields to construct quantum fields in Minkowski space [76]. Orthogonal work which explores connections between quantum field theory and machine learning relates to the ADS/CFT correspondence [77, 78], or the theory of Gaussian processes [79–81].

Here, we will demonstrate, via the Hammersley-Clifford theorem [82–86], that the ϕ^4 lattice field theory is, by definition, a machine learning algorithm. We will then derive a ϕ^4 neural network architecture that generalizes a certain class of standard neural network architectures, namely restricted Boltzmann machines [87–90]. Finally, we will conduct numerical applications to establish the use of ϕ^4 machine learning algorithms, and we will discuss the opportunity to investigate machine learning within

lattice field theory.

7.2 Probabilistic graphical models and Markov random fields

We will start by presenting the fundamentals related to the framework of probabilistic graphical models. Specifically, we will focus on the case where a system can be represented by a graph, and the degrees of freedom of the system are positioned on the vertices of the graph. The degrees of freedom are then connected through edges. In this thesis, we will investigate exclusively the case where the edges of the graph are undirected, which means that the direction of the edge from a vertex i to a vertex j or, conversely, from j to i is irrelevant. We thus discuss a special case of probabilistic graphical models, namely undirected graphical models. Moreover, we are interested in a special case of undirected graphical models, that is, graphs which satisfy a condition called the Markov property. This type of undirected graph is called a Markov random field. We will hence start by introducing the concept of a Markov random field.

As mentioned above, let us consider a finite set Λ , which we express as a graph $\mathcal{G}(\Lambda, e)$. This graph then describes a physical system where the degrees of freedom $i, j \in \Lambda$ correspond to the vertices of the graph and the edges e which connect i and j are undirected. We associate to each vertex $i \in \Lambda$ a random variable ϕ_i . A configuration comprises the set of random variables and is denoted as ϕ , and the set of all possible configurations will be denoted as Φ . We are generally interested in studying concepts of conditional independence and locality for the random variables within the graph \mathcal{G} , and hence we aim to transition from mathematical expressions in relation to all possible configurations ϕ of a lattice to expressions in relation to all possible values ϕ_i of a lattice site.

We now define the concept of a neighbour of a vertex i as another vertex $j \in \Lambda, j \neq i$ which is connected with i through an edge. We denote as n_i all neighbours of a vertex i . We will ambiguously use the notation ϕ_i when discussing either the random variable or the corresponding vertex i . An important concept in relation to graphs is the notion of a clique. We define a clique as a set which comprises at least two vertices which are neighbours. We then define a maximal clique $c \in C$ as the set to which no additional vertex can be included such that all included vertices are neighbours, i.e. such that the corresponding set remains a clique, see Fig. 7.1.

To clarify the concepts we will briefly describe what constitutes a maximal clique in terms of commonly used graphs and lattices. On the square lattice a maximal clique is defined based on two-nearest neighbours, for instance the vertices that correspond

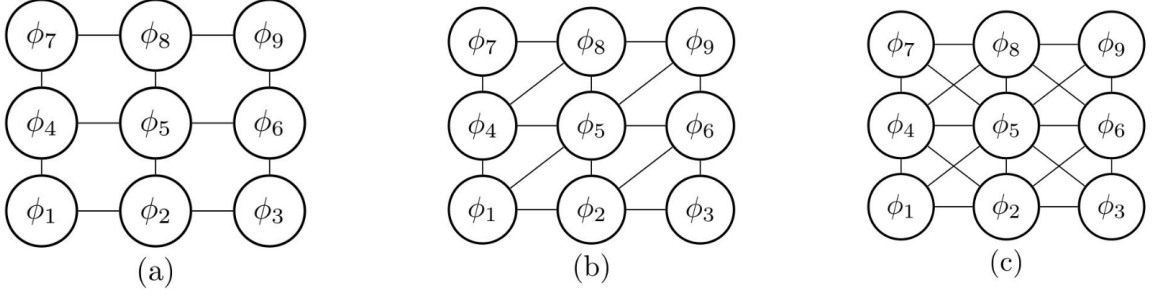


Figure 7.1: (a) A square lattice where a maximal clique is a two-site clique. (b) A triangular lattice where the maximal clique is a triangle, i.e. a three site clique, (c) A square lattice with both diagonals where the maximal clique is a square, i.e. a four-site clique. See text for examples.

to the random variables $\{\phi_5, \phi_6\}$, because no additional vertex can be included that is simultaneously a neighbor with both ϕ_5 and ϕ_6 . The concept becomes clearer in the case of a triangular lattice. For instance, the set $\{\phi_1, \phi_2\}$ in Fig. 7.1b defines a clique because ϕ_1 and ϕ_2 are neighbours, but the clique is not maximal because there exists another vertex, specifically ϕ_5 , that is simultaneously a neighbour with both ϕ_1 and ϕ_2 . As a result on the triangular lattice a maximal clique is $\{\phi_1, \phi_2, \phi_5\}$. Similar arguments can be extended on the square lattice with both diagonals. Here, a set of two neighbours or three neighbours defines a clique but a maximal clique is obtained only when four neighbours are included within the set. For instance a maximal clique is $\{\phi_5, \phi_6, \phi_8, \phi_9\}$. The final graph to be discussed is the case of the bidirected graph, depicted in Fig. 7.2, where the structure of the cliques is analogous to the square lattice, specifically only two-site cliques are maximal.

We remark that, since the vertices of the graph $\mathcal{G}(\Lambda, e)$ correspond to the sites of a physical model, the probability measures on the set of subsets of Λ define a probability distribution p which is the equilibrium distribution of a physical model. We will now introduce the concept of a Markov random field. Specifically, we call a Markov random field a set of random variables, described by an undirected graph $\mathcal{G}(\Lambda, e)$, that satisfy the local Markov property with respect to the graph structure:

$$p(\phi_i | (\phi_j)_{j \in \Lambda - i}) = p(\phi_i | (\phi_j)_{j \in n_i}). \quad (7.1)$$

In simple terms the local Markov property states that what happens in a small region of a lattice is independent with what happens in regions of the lattice that are further away. Formally, it states that a random variable ϕ_i , $i \in \Lambda$ is conditionally independent of all other random variables j in the set Λ given (or excluding) its

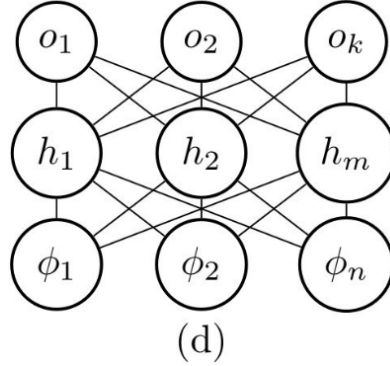


Figure 7.2: A bipartite graph that has identical independence structure such as the square lattice. The maximal cliques are two-site cliques.

neighbours n_i . A probability distribution which satisfies the local Markov property is then associated with the events generated by a Markov random field.

We will prove the local Markov property through the Hammersley-Clifford theorem.

Theorem 1 (Hammersley-Clifford) *Any probability distribution p with a strictly positive mass or density that is represented by an undirected graph \mathcal{G} satisfies the local Markov property if and only if p can be factorized, with respect to the graph structure, in terms of strictly positive potential functions ψ_c over the maximal cliques $c \in C$, i.e.:*

$$p(\phi) = \frac{1}{Z} \prod_{c \in C} \psi_c(\phi), \quad (7.2)$$

where the normalization constant $Z = \int_{\phi} \prod_{c \in C} \psi_c(\phi) d\phi$ is the partition function of the system.

The Hammersley-Clifford theorem establishes an equivalence between the factorization of random variables on a graph and the conditional independence properties that they satisfy. We emphasize that we discuss a simple variation of the theorem, which is actually more generally applicable. For a mathematical treatment of probabilistic graphical models see Ref. [74]. To establish that the ϕ^4 scalar field theory is a Markov random field we will demonstrate that the Hammersley-Clifford theorem holds for the ϕ^4 Boltzmann probability distribution.

7.3 The ϕ^4 theory as a Markov field

We repeat, for convenience, the expression of the Euclidean action of the two-dimensional ϕ^4 scalar field theory which is discretized on a square lattice:

$$S_E = -\kappa_L \sum_{\langle ij \rangle} \phi_i \phi_j + \frac{(\mu_L^2 + 4\kappa_L)}{2} \sum_i \phi_i^2 + \frac{\lambda_L}{4} \sum_i \phi_i^4. \quad (7.3)$$

We will now work with the disordered version of the above action, by substituting $w = \kappa_L$, $a = (\mu_L^2 + 4\kappa_L)/2$, $b = \lambda_L/4$, and considering w, a, b as inhomogeneous. We then arrive to the action:

$$S(\phi; \theta) = - \sum_{\langle ij \rangle} w_{ij} \phi_i \phi_j + \sum_i a_i \phi_i^2 + \sum_i b_i \phi_i^4, \quad (7.4)$$

where $\theta = \{w_{ij}, a_i, b_i\}$ is the set of coupling constants, which we will ambiguously call variational parameters.

The study of disordered systems [91], is motivated by the fact that realistic systems always include some form of impurity or inconsistency in their description. For example a realistic system might be interacting with an inhomogeneous external field instead of a perfectly homogeneous field, and a realistic material could always include some form of impurity via the inclusion of small particles of a different type of material. We emphasize that even simple disordered systems, such as the inhomogeneous case of the Ising model, namely the Ising spin glass, do exist experimentally. All of the conditions that we will prove from now on for the disordered action, also hold for the conventional ϕ^4 theory of Eq. (7.3). Nevertheless we will work with the disordered case since it is mathematically more general, namely for $w_{ij} = w, a_i = a, b_i = b$ it reduces to the traditional action.

The probability distribution $p(\phi; \theta)$ of the ϕ^4 scalar field theory is then given by:

$$p(\phi; \theta) = \frac{\exp[-S(\phi; \theta)]}{\int_{\phi} \exp[-S(\phi; \theta)] d\phi}. \quad (7.5)$$

Because of the theorems related to undirected graphical models which were described above, it is a trivial matter to prove that the ϕ^4 scalar field theory is equivalent to a Markov random field. We observe that a lattice field theory is, by definition, formulated on a graph $\mathcal{G} = (\Lambda, e)$, where each of the vertices or lattice sites belong to the finite set Λ and the edges e correspond to the pairwise interactions. In addition, we observe that we aim to factorize the probability distribution $p(\phi; \theta)$ in terms of strictly positive potential functions ψ_c , so we are able to multiply with strictly

positive functions derived from subsets of the maximal clique c [92]. Practically, this implies that besides functions related to the terms $w_{ij}\phi_i\phi_j$ which emerge from two-site cliques, we are also able to include functions related to terms from one-site cliques, which are subsets of two-site cliques, and hence we can include functions related to the $a_i\phi_i^2$ and $b_i\phi_i^4$ terms. In summary, and by recalling that we have chosen periodic boundary conditions for the system, we arrive at the following choice of a potential function which is able to factorize the probability distribution:

$$\psi_c = \exp \left[-w_{ij}\phi_i\phi_j + \frac{1}{4}(a_i\phi_i^2 + a_j\phi_j^2 + b_i\phi_i^4 + b_j\phi_j^4) \right]. \quad (7.6)$$

In the above expression i and j denote two nearest-neighbours. We then observe that this choice of a potential function leads to a factorization of the probability distribution as

$$p(\phi; \theta) = \frac{\exp \left[\sum_{c \in C} \ln \psi_c(\phi) \right]}{\int_{\phi} \exp \left[\sum_{c \in C} \ln \psi_c(\phi) \right] d\phi} = \frac{1}{Z} \prod_{c \in C} \psi_c(\phi). \quad (7.7)$$

We have therefore proved that the probability distribution of the ϕ^4 scalar field theory can be factorized in terms of potential functions ψ_c per maximal cliques $c \in C$ and therefore the ϕ^4 theory is a Markov random field.

To provide some further insights into the Markov property, and hence Markov fields, we recall that the Markov property is a fundamental concept in the theory of Markov processes, including Markov chain Monte Carlo simulations. The Markov property in a Markov chain can be expressed as the condition that, given a certain configuration ϕ^k , a future configuration ϕ^{k+1} depends only on the current configuration ϕ^k and not on configurations that preceded it, such as ϕ^{k-1} :

$$P(\phi^{k+1} | \phi^k, \dots, \phi^0) = P(\phi^{k+1} | \phi^k). \quad (7.8)$$

The Markov property in a Markov chain is therefore a condition that can be intuitively understood in terms of an evolution related to time. Conversely, in a Markov random field we generalize this condition, expressed in Eq. (7.8), to a condition of (a high-dimensional) space, as expressed in Eq. (7.1). In other words, via Markov random fields we are interested in Markov processes in high dimensions. Markov random fields are widely used as machine learning algorithms, and we will now explore relevant applications based on ϕ^4 Markov random fields.

A central concept in the following discussion is the notion of a distance function between two probability distribution. Generally, this will always be our aim in

probabilistic machine learning applications: we are interested in minimizing a distance function between the probability distribution $p(\phi; \theta)$ of the machine learning algorithm and a target probability distribution $q(\phi)$ that we are interested in approximating or learning. After the minimization of the distance function, which is achieved by searching for the optimal values of the variational parameters θ , we have constructed a representation of $q(\phi)$ within $p(\phi; \theta)$. This implies that we are able to utilize the probability distribution $p(\phi; \theta)$ of the machine learning algorithm to draw samples that correspond to the other probability distribution $q(\phi)$, a probability distribution that we might otherwise never be able to sample or whose form might be unknown to us.

To establish an equivalence between the probability distribution $p(\phi; \theta)$ of a machine learning algorithm, and a target probability distribution $q(\phi)$ we will utilize an expression called the Kullback-Leibler divergence:

$$KL(p||q) = \int_{-\infty}^{\infty} p(\phi; \theta) \ln \frac{p(\phi; \theta)}{q(\phi)} d\phi \geq 0. \quad (7.9)$$

We emphasize that the Kullback-Leibler divergence, which is equivalent to a relative entropy, is not a proper distance. The reason is that it does not satisfy the triangle inequality and it is not symmetric. Nevertheless, the Kullback-Leibler divergence satisfies positivity, and it becomes zero only when the two probability distributions $p(\phi; \theta)$ and $q(\phi)$ become equal. For this reason we will call the Kullback-Leibler divergence an asymmetric distance, as it still establishes a measure of the difference between two probability distributions. In fact, we will observe that the asymmetry of the function is actually beneficial in relation to applications since we will use both $KL(p||q)$ as well as $KL(q||p)$ for distinct applications.

7.4 Machine learning with ϕ^4 Markov random fields

7.4.1 Learning without predefined data

We are now interested in utilizing the probability distribution of a ϕ^4 Markov random field with action $S(\phi; \theta)$, given by Eq. (7.4), to approximate a target probability distribution $q(\phi)$ whose form we know. Specifically, we consider that $q(\phi)$ is a Boltzmann probability distribution that describes a statistical system or a quantum field theory with Hamiltonian or action \mathcal{A} and is given by

$$q(\phi) = \frac{\exp[-\mathcal{A}(\phi)]}{Z_{\mathcal{A}}}. \quad (7.10)$$

We now expand the Kullback-Leibler divergence to obtain:

$$\langle \ln p(\phi; \theta) \rangle_{p(\phi; \theta)} - \langle \ln q(\phi) \rangle_{p(\phi; \theta)} \geq 0, \quad (7.11)$$

where we recall that the notation $\langle \rangle_{p(\phi; \theta)}$ denotes the calculation of an expectation value under the probability distribution $p(\phi; \theta)$. We now substitute the two probability distributions $p(\phi; \theta)$ and $q(\phi)$ in the expression to obtain:

$$-\langle \ln Z_{\mathcal{A}} \rangle_{p(\phi; \theta)} \leq \langle \mathcal{A} - S \rangle_{p(\phi; \theta)} - \langle \ln Z \rangle_{p(\phi; \theta)}, \quad (7.12)$$

where we observe that expectation values in relation to partition functions are constant, hence the above equation is equal to:

$$-\ln Z_{\mathcal{A}} \leq \langle \mathcal{A} - S \rangle_{p(\phi; \theta)} - \ln Z. \quad (7.13)$$

In the current chapter we will consider that any parameter, such as the inverse temperature β , is absorbed within the action. We can then consider that $F_{\mathcal{A}} = -\ln Z_{\mathcal{A}}$ and $F = -\ln Z$, where F denotes the free energy and we obtain:

$$F_{\mathcal{A}} \leq \langle \mathcal{A} - S \rangle_{p(\phi; \theta)} + F \equiv \mathcal{F}. \quad (7.14)$$

We can now establish two important observations for the above equation. The first observation is that it sets a rigorous bound to the calculation of the free energy $F_{\mathcal{A}}$ of the target system with action \mathcal{A} . The second observation is that this bound is entirely dependent on calculations conducted only under the probability distribution $p(\phi; \theta)$ of the ϕ^4 Markov random field. Practically, this means that we can use exclusively samples drawn from $p(\phi; \theta)$ with action $S(\phi; \theta)$ to approximate another system with probability distribution $q(\phi)$ and action \mathcal{A} . To achieve this, we have to minimize the quantity defined above, specifically the variational free energy \mathcal{F} .

The minimization of the variational free energy \mathcal{F} will be achieved with a gradient descent approach. To establish a gradient-based approach we calculate the derivatives of \mathcal{F} in relation to one of the variational parameters θ and obtain:

$$\frac{\partial \mathcal{F}}{\partial \theta_i} = \frac{\partial \langle \mathcal{A} \rangle_{p(\phi; \theta)}}{\partial \theta_i} - \frac{\partial \langle S \rangle_{p(\phi; \theta)}}{\partial \theta_i} - \frac{\partial (-\ln Z)}{\partial \theta_i}, \quad (7.15)$$

where each term is calculated as:

$$\frac{\partial \langle \mathcal{A} \rangle_{p(\phi; \theta)}}{\partial \theta_i} = \frac{\partial}{\partial \theta_i} \left[\frac{\int_{\phi} \mathcal{A}(\phi) \exp[-S(\phi; \theta)] d\phi}{\int_{\phi} \exp[-S(\phi; \theta)] d\phi} \right] \quad (7.16)$$

$$= -\left\langle \mathcal{A} \frac{\partial S}{\partial \theta_i} \right\rangle_{p(\phi; \theta)} + \langle \mathcal{A} \rangle_{p(\phi; \theta)} \left\langle \frac{\partial S}{\partial \theta_i} \right\rangle_{p(\phi; \theta)}, \quad (7.17)$$

$$\frac{\partial \langle S \rangle_{p(\phi; \theta)}}{\partial \theta_i} = \frac{\partial}{\partial \theta_i} \left[\frac{\int_{\phi} S(\phi; \theta) \exp[-S(\phi; \theta)] d\phi}{\int_{\phi} \exp[-S(\phi; \theta)] d\phi} \right] \quad (7.18)$$

$$= \left\langle \frac{\partial S}{\partial \theta_i} \right\rangle_{p(\phi; \theta)} - \left\langle S \frac{\partial S}{\partial \theta_i} \right\rangle_{p(\phi; \theta)} + \langle S \rangle_{p(\phi; \theta)} \left\langle \frac{\partial S}{\partial \theta_i} \right\rangle_{p(\phi; \theta)}, \quad (7.19)$$

$$\frac{\partial(-\ln Z)}{\partial \theta_i} = - \frac{\int_{\phi} \frac{\partial}{\partial \theta_i} (-S(\phi; \theta)) \exp[-S(\phi; \theta)] d\phi}{\int_{\phi} \exp[-S(\phi; \theta)] d\phi} = \left\langle \frac{\partial S}{\partial \theta_i} \right\rangle_{p(\phi; \theta)}. \quad (7.20)$$

By substituting the above expressions to Eq. (7.15) we obtain:

$$\frac{\partial \mathcal{F}}{\partial \theta_i} = - \left\langle \mathcal{A} \frac{\partial S}{\partial \theta_i} \right\rangle + \langle \mathcal{A} \rangle \left\langle \frac{\partial S}{\partial \theta_i} \right\rangle - \left\langle \frac{\partial S}{\partial \theta_i} \right\rangle + \left\langle S \frac{\partial S}{\partial \theta_i} \right\rangle - \langle S \rangle \left\langle \frac{\partial S}{\partial \theta_i} \right\rangle + \left\langle \frac{\partial S}{\partial \theta_i} \right\rangle. \quad (7.21)$$

The derivative of the variational free energy \mathcal{F} in terms of a variational parameter θ_i is then:

$$\frac{\partial \mathcal{F}}{\partial \theta_i} = \langle \mathcal{A} \rangle \left\langle \frac{\partial S}{\partial \theta_i} \right\rangle - \left\langle \mathcal{A} \frac{\partial S}{\partial \theta_i} \right\rangle + \left\langle S \frac{\partial S}{\partial \theta_i} \right\rangle - \langle S \rangle \left\langle \frac{\partial S}{\partial \theta_i} \right\rangle. \quad (7.22)$$

We will now update the parameters θ_i at each step of the optimization process until the variational free energy is minimized and the two probability distributions have therefore become equal. This is achieved via the following update rule:

$$\theta^{(t+1)} = \theta^{(t)} - \eta * \mathcal{L}, \quad (7.23)$$

where the quantity η denotes the learning rate and the loss function \mathcal{L} is replaced by the derivative of each of the variational parameters $\partial \mathcal{F} / \partial \theta^{(t)}$. Here, t denotes each step or, equivalently, epoch of the optimization process.

We will explore applications of the previously discussed approach by introducing a general ϕ^4 action that we will utilize to investigate different applications between various probability distributions:

$$\mathcal{A} = \sum_{k=1}^5 g_k \mathcal{A}^{(k)} = g_1 \sum_{\langle ij \rangle_{nn}} \phi_i \phi_j + g_2 \sum_i \phi_i^2 \quad (7.24)$$

$$+ g_3 \sum_i \phi_i^4 + g_4 \sum_{\langle ij \rangle_{nnn}} \phi_i \phi_j + ig_5 \sum_i \phi_i^2. \quad (7.25)$$

We observe that this action \mathcal{A} includes a term with a next-nearest-neighbor interaction nnn and another term with an imaginary coupling constant ig_5 , where i

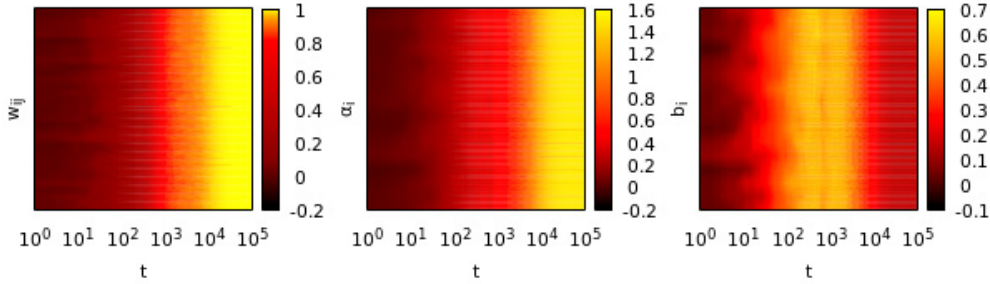


Figure 7.3: The evolution of the variational parameters θ in relation to the number of epochs t . The x-axis is logarithmic.

outside the sums denotes the imaginary unit. We have chosen the couplings constants as $g_1 = g_4 = -1$, $g_2 = 1.52425$, $g_3 = 0.175$, $g_5 = 0.15$, where the choice of g_1 , g_2 , g_3 defines a ϕ^4 theory in the vicinity of the second-order phase transition when $g_4 = g_5 = 0$.

Our first application is a proof-of-principle demonstration to explore if the minimization of the variational free energy via a gradient-descent method is successful. For this reason we consider the following target action:

$$\mathcal{A}_{\{3\}} = \sum_{k=1}^3 g_k \mathcal{A}^{(k)} = g_1 \sum_{\langle ij \rangle_{nn}} \phi_i \phi_j + g_2 \sum_i \phi_i^2 + g_3 \sum_i \phi_i^4. \quad (7.26)$$

Our aim is to approximate the probability distribution $q(\phi)$ which is defined by the above action $\mathcal{A}_{\{3\}}(\phi)$ using instead the probability distribution $p(\phi; \theta)$ of the ϕ^4 Markov random field with action $S(\phi; \theta)$. The discussed problem is easy to solve. The inhomogeneous coupling constants of the action in Eq. (7.4), which are randomly drawn from a Gaussian distribution, must converge to their homogeneous values $g_1 = -1$, $g_2 = 1.52425$, $g_3 = 0.175$ after the minimization of the variational free energy is achieved. The evolution of the variational parameters θ in terms of the epochs is depicted in Fig. 7.3. We observe that, given sufficient training time, the inhomogeneous variational parameters converge towards the anticipated values $g_1 = -1$, $g_2 = 1.52425$, $g_3 = 0.175$. In fact, after 10^5 epochs the precision with which the inhomogeneous coupling constants approximate the target values is of order of magnitude 10^{-8} , therefore verifying that the minimization of the Kullback-Leibler divergence, or equivalently the variational free energy, via a gradient-based method is successful.

We will now focus on a more intricate example by defining as a target probability distribution $q(\phi)$ one that is described by an action $\mathcal{A}_{\{4\}}$ which includes longer-range interactions:

$$\mathcal{A}_{\{4\}} = \sum_{k=1}^4 g_k \mathcal{A}^{(k)} = g_1 \sum_{\langle ij \rangle_{nn}} \phi_i \phi_j + g_2 \sum_i \phi_i^2 + g_3 \sum_i \phi_i^4 + g_4 \sum_{\langle ij \rangle_{nnn}} \phi_i \phi_j. \quad (7.27)$$

Our aim here is again to utilize the probability distribution $p(\phi; \theta)$ of the inhomogeneous action $S(\phi; \theta)$ to approximate the probability distribution $q(\phi)$ of the above action $\mathcal{A}_{\{4\}}$. However, the current problem is more challenging since the action $S(\phi; \theta)$ does not include a term that is able to learn the next-nearest-neighbour interactions that exist within the action $\mathcal{A}_{\{4\}}$. Nevertheless, the action $S(\phi; \theta)$ is inhomogeneous, and we will explore if this inhomogeneity in the coupling constants enables the representation of systems with longer-range interactions. In other words, we aim to explore if the inhomogeneity in the coupling constants of the system, is able to increase the representational capacity of the probability distribution of a ϕ^4 theory, in terms of classes of probability distributions that the inhomogeneous system can model.

For this example we will estimate the KL divergence between the probability distribution $p(\phi; \theta)$ and the probability distribution $q(\phi)$ to observe if training is successful. To allow for a comparison, we will additionally estimate the KL divergence between the probability distribution of the action $\mathcal{A}_{\{3\}}$ and the probability distribution $q(\phi)$. The actions $S(\phi; \theta)$ and $\mathcal{A}_{\{3\}}$ have identical terms but the former has inhomogeneous coupling constants. We are therefore interested in investigating if the inhomogeneous action can lead to smaller values of the Kullback-Leibler divergence and, as a result, if $p(\phi; \theta)$ can approximate $q(\phi)$ better.

The values of the Kullback-Leibler divergence are depicted in Fig. 7.4. We approximate the results for the two probability distributions and their corresponding actions $S(\phi; \theta)$ and $\mathcal{A}_{\{3\}}$ based on a finite sample of configurations, which is of the same sample size, to allow for a direct comparison of the statistical fluctuations. We observe that as the training time increases the Kullback-Leibler divergence converges towards a zero value for the probability distribution of the inhomogeneous action $S(\phi; \theta)$, therefore indicating that the two probability distributions $p(\phi; \theta)$ and $q(\phi)$ become approximately equal. Since inhomogeneous actions are able to absorb terms with longer-range interactions we anticipate that they are capable of representing intricate target actions. This is further verified by the observation in Fig. 7.4 that the probability distribution of the action $S(\phi; \theta)$ approximates the target probability distribution of action $\mathcal{A}_{\{4\}}$ better than the probability distribution of action $\mathcal{A}_{\{3\}}$.

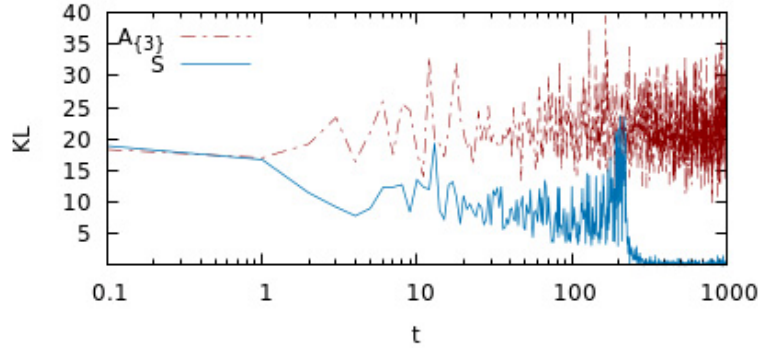


Figure 7.4: Estimation of the Kullback-Leibler divergence versus epoch t for the probability distribution of the ϕ^4 Markov random field with the inhomogeneous action $S(\phi; \theta)$. Results are additionally included for the probability distribution of action $\mathcal{A}_{\{3\}}$, see text for details.

Since inhomogeneous actions have increased representational capacity in approximating target probability distributions, compared to homogeneous actions, we will investigate if they can be utilized to reweight in regions of parameter space that are otherwise inaccessible, for instance due to an insufficient overlap of statistical ensembles. In addition, reweighting will now emerge as a different tool: specifically as a means to correct approximating probability distributions. To clarify, consider that after the minimization of the Kullback-Leibler divergence we might obtain a value of $KL(p||q) \approx 0$. This means that the two probability distributions $p(\phi; \theta)$ and $q(\phi)$ are approximately equal. Through a reweighting step we might be able to make them exactly equal and hence guarantee that $KL(p||q) = 0$.

We will now discuss the above statements of approximation and exactness from a different perspective. Specifically, consider the expectation value $\langle O \rangle$ of an arbitrary observable as calculated under the probability distribution that corresponds to action $\mathcal{A}_{\{4\}}$:

$$\langle O \rangle_{q(\phi)} = \frac{\sum_{l=1}^N \tilde{p}_l^{-1} O_l \exp[-\sum_{k=1}^4 g_k \mathcal{A}_l^{(k)}]}{\sum_{l=1}^N \tilde{p}_l^{-1} \exp[-\sum_{k=1}^4 g_k \mathcal{A}_l^{(k)}]}. \quad (7.28)$$

There are two different ways to calculate the expectation value of an arbitrary observable based on the above equation, and both of them depend on the choice of the sampling probability distribution \tilde{p} .

The first choice is to draw samples from the probability distribution $p(\phi; \theta)$, and conjecture that the Kullback-Leibler divergence has become exactly zero, which im-

plies that $p(\phi; \theta) = q(\phi)$. However, the problem with this direction is that a systematic error will be introduced into the calculation of the expectation value, because $KL(p||q) \approx 0$, and hence the two probability distribution $p(\phi; \theta)$ and $q(\phi)$ are not exactly equal. We will not pursue this research direction.

The second direction, which involves the use of reweighting, is to again draw samples from $p(\phi; \theta)$ but this time consider that the samples have been drawn from the actual probability distribution $p(\phi; \theta)$ of the inhomogeneous action $S(\phi; \theta)$. This automatically acts as a reweighting step to correct the difference between the two probability distributions. The remaining question is whether this reweighting step is anticipated to be successful or, equivalently, if there exists a sufficient overlap of statistical ensembles between the two probability distributions $p(\phi; \theta)$ and $q(\phi)$. The answer is that we anticipate the reweighting step to be successful because of the training procedure: we have minimized the Kullback-Leibler divergence $KL(p||q) \approx 0$, so the two probability distributions are almost equal, and hence we anticipate that a sufficient overlap of statistical ensembles exists.

We will now combine the reweighting process, established in Eq. (7.28), with the reweighted extrapolations that we discussed in the preceding chapters. Our aim is to benchmark the accuracy with which the probability distribution $p(\phi; \theta)$ of the inhomogeneous action $S(\phi; \theta)$ has approximated the target distribution $q(\phi)$ with action $\mathcal{A}_{\{4\}}$. For this reason, instead of only introducing reweighting as a correction step between the probability distributions $p(\phi; \theta)$ and $q(\phi)$, we will also investigate if it is possible to extrapolate expectation values in the parameter space of $\mathcal{A}_{\{4\}}$ along the trajectory of a selected coupling constant, using $S(\phi; \theta)$. Specifically, we are now interested in the following reweighting relation:

$$\langle O \rangle = \frac{\sum_{l=1}^N O_l \exp[S_l - g'_j \mathcal{A}_l^{(j)} - \sum_{k=1, k \neq j}^5 g_k \mathcal{A}_l^{(k)}]}{\sum_{l=1}^N \exp[S_l - g'_j \mathcal{A}_l^{(j)} - \sum_{k=1, k \neq j}^5 g_k \mathcal{A}_l^{(k)}]}. \quad (7.29)$$

Based on our prior discussion, this reweighting expression includes the correction step between the probability distributions $p(\phi; \theta)$ and $q(\phi)$, an extrapolation along the parameter space of a coupling constant g'_j , and another extrapolation via the inclusion of an imaginary term. We recall that the action $\mathcal{A}_{\{5\}} \equiv \mathcal{A}$ is complex-valued, see Eq. (7.24). For the following results we consider $j = 4$ and hence extrapolate in parameter space along the trajectory of the coupling constant g'_4 , which is related to the longer-range interaction term. Specifically, we conduct this extrapolation for values $g'_4 \in [-0.85, -1.15]$ while simultaneously including the imaginary valued term. We recall that $p(\phi; \theta)$ was trained to approximate the action $\mathcal{A}_{\{4\}}$ with coupling constant $g_4 = -1$.

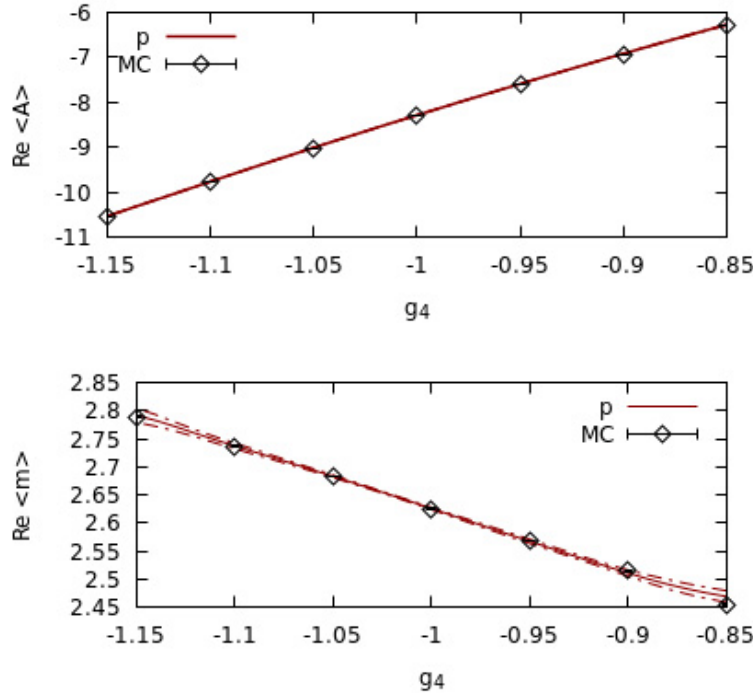


Figure 7.5: Real part of the action \mathcal{A} (top) and real part of the magnetization m (bottom) versus the coupling constant g_4 .

The results for the expectation values of the real part of the action \mathcal{A} and the magnetization m , which are obtained through reweighting, are depicted in Fig. 7.5. We observe that the results agree within statistical errors with calculations obtained through the use of reweighting from the phase-quenched theory to the complex action, depicted as empty points. Consequently, the probability distribution $p(\phi; \theta)$ has approximated $q(\phi)$ with sufficiently high accuracy that it is possible not just to obtain expectation values for the specific coupling constants in the target action \mathcal{A} , but in addition to obtain accurate expectation values even under extrapolations to other coupling constants of \mathcal{A} .

We are now interested in answering the question of how to define the range in which we are able to conduct histogram reweighting extrapolations. For this reason we consider as an observable in Eq. (7.29) the value of the inhomogeneous action $S(\phi; \theta)$. We then construct histograms for the inhomogeneous action $S(\phi; \theta)$ based on unique values of the action within the Markov chain Monte Carlo dataset that we have obtained. We then reexpress the expectation value in terms of weight functions

$\mathcal{W}(S)$ and uniquely sampled values of the action $S(\phi; \theta)$ as:

$$\langle S \rangle = \sum_S S \mathcal{W}(S). \quad (7.30)$$

The weight functions for each uniquely sampled value of the action S are then given by:

$$\mathcal{W}(S) = \frac{\sum_{\Re[\mathcal{A}'], \Im[\mathcal{A}']} h(S, \Re[\mathcal{A}'], \Im[\mathcal{A}']) \exp[S - \Re[\mathcal{A}'] - i\Im[\mathcal{A}']]}{\sum_{S, \Re[\mathcal{A}'], \Im[\mathcal{A}']} h(S, \Re[\mathcal{A}'], \Im[\mathcal{A}']) \exp[S - \Re[\mathcal{A}'] - i\Im[\mathcal{A}']]}, \quad (7.31)$$

where $h(S, \Re[\mathcal{A}'], \Im[\mathcal{A}'])$ is a multi-dimensional histogram, constructed based on the value of the action S and each term that we are interested in extrapolating to. The quantity \mathcal{A}' corresponds to:

$$\mathcal{A}' = g'_j \mathcal{A}^{(j)} + \sum_{k=1, k \neq j}^5 g_k \mathcal{A}^{(k)}. \quad (7.32)$$

We recall that in the preceding chapters, where histogram reweighting was first discussed in relation to inverse temperatures β and β' , we emphasized that, for reweighting to be successful, a sufficient overlap of the histograms of the energies related to the inverse temperatures β and β' was required. In the current example, which is admittedly conceptually more complicated, the essence remains the same: we will investigate the permitted reweighting range via the weight functions, and hence the histograms of the action $S(\phi; \theta)$, by observing at what part of parameter space inconsistencies will emerge. When we locate these inconsistencies we can guarantee that we can't extrapolate to coupling constants that reside beyond that range.

To demonstrate the above ideas numerically, we calculate the weight functions $\mathcal{W}(S)$ for different values of coupling constants g'_4 and depict the results in Fig. 7.6. We additionally include the quantity $\mathcal{W}'(S)$, which corresponds to the weight function proportional to the histograms without any extrapolation in parameter space. We recall that the action $S(\phi; \theta)$ was trained to approximate an action $\mathcal{A}_{\{4\}}$ with $g_4 = -1$. We then observe that when $g'_4 = -0.95$ or $g'_4 = -1.05$ the weight functions have been successfully predicted. In contrast, when $g'_4 = -0.8$ a noticeable inconsistency has emerged. We have therefore demonstrated that we anticipate histogram reweighting to be inaccurate at $g'_4 = -0.8$, as well as for $g'_4 < -0.8$. This behaviour is additionally evident from the figure of the magnetization, see Fig. 7.5, where we observe that the statistical errors begin to increase near the range $g_4 = -0.85$.

We can now evidence more directly what has been achieved, in relation to the representational capacity of probability distributions described by inhomogeneous

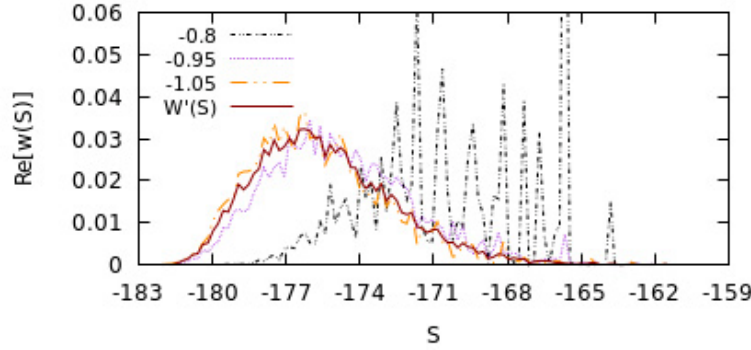


Figure 7.6: Real part of the weight function $W(S)$ versus the action S of the ϕ^4 Markov random field.

actions. Assume that we have already obtained samples from the probability distribution of the following action

$$\mathcal{A}_{\{3\}} = \sum_{k=1}^3 g_k \mathcal{A}^{(k)} = g_1 \sum_{\langle ij \rangle_{nn}} \phi_i \phi_j + g_2 \sum_i \phi_i^2 + g_3 \sum_i \phi_i^4, \quad (7.33)$$

which is locally defined on the graph, i.e. it includes interactions of lattice sites with adjacent neighbours, and we are now interested in reweighting from $\mathcal{A}_{\{3\}}$ to the action $\mathcal{A}_{\{4\}}$ with $g_4 = -1$ which includes longer-range interactions. We will demonstrate that this is not possible, due to an insufficient overlap of statistical ensembles. However, reweighting from the probability distribution of the inhomogeneous action $S(\phi; \theta)$ to $\mathcal{A}_{\{4\}}$ was possible, even though $S(\phi; \theta)$ is additionally locally defined on the graph. The difference between $S(\phi; \theta)$ and $\mathcal{A}_{\{3\}}$ is the inhomogeneity in the coupling constants.

The weight functions, constructed from the probability distribution of action $\mathcal{A}_{\{3\}}$, are depicted in Fig. 7.7. In comparison with Fig. 7.6 we observe that the values of the action lie at an entirely different scale, a first indication that reweighting is impossible. The second observation is that inconsistencies begin to emerge for the value of $g'_4 = -0.2$, indicating that we cannot extrapolate further in parameter space for $g'_4 < -0.2$. However, to extrapolate from action $\mathcal{A}_{\{3\}}$ to the action $\mathcal{A}_{\{4\}}$ we would require a permitted reweighting range that would include the value of $g'_4 = -1.0$. We hence conclude that reweighting from the locally defined action $\mathcal{A}_{\{3\}}$ to the longer range action $\mathcal{A}_{\{4\}}$ is impossible, even though we were able to achieve this with a locally defined inhomogeneous action $S(\phi; \theta)$. Consequently, we have provided evidence to establish that the probability distributions of inhomogeneous actions possess increased

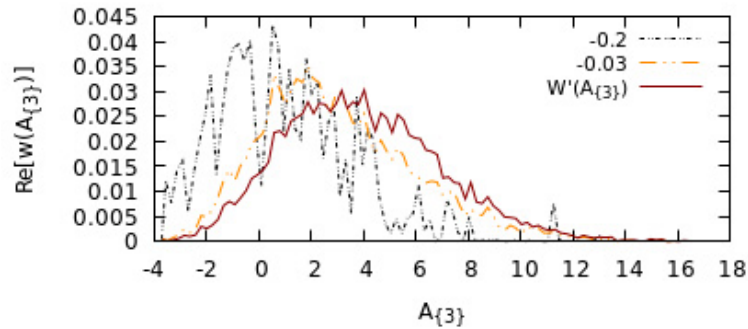


Figure 7.7: Real part of the weight function $W(\mathcal{A}_3)$ versus the action \mathcal{A}_3 .

representational capacity compared to the probability distributions of homogeneous actions which include identical terms, and they can therefore be utilized to model a richer class of target probability distributions.

7.4.2 Learning with predefined data

Previously, we demonstrated that the probability distribution $p(\phi; \theta)$ of a ϕ^4 Markov random field can be utilized to approximate a target probability distribution $q(\phi)$ whose form was known: it was a Boltzmann probability distribution and we had knowledge of its corresponding action \mathcal{A} . Here, we will shift focus on a different type of machine learning. Specifically we will consider that we have already obtained a set of samples from an unknown probability distribution $q(\phi)$, but we do not know what the form of the probability distribution is. Our aim is then to approximate this unknown probability distribution $q(\phi)$ by utilizing again the probability distribution $p(\phi; \theta)$ of the ϕ^4 Markov random field.

This type of machine learning problem, which establishes an equivalence between an empirical probability distribution $q(\phi)$ and a model probability distribution $p(\phi; \theta)$, is general and appears frequently. In fact, most relevant machine learning applications in computer science, outside of the research field of physics, deal with this problem. For instance, consider that one has available experimental data that correspond to an unknown empirical probability distribution $q(\phi)$ and one is interested in learning a probability distribution $p(\phi; \theta)$ that is able to represent these data and reproduce them. Another example is, as simple as, the inclusion of some images within a dataset which then construct an empirical probability distribution $q(\phi)$ to be learned by a machine learning algorithm. Any type of available data which encode an empirical

probability distribution can then be mapped to a model probability distribution.

To introduce this type of machine learning, we will again utilize the Kullback-Leibler divergence, but this time we will work with the opposite divergence:

$$KL(q||p) = \int_{-\infty}^{\infty} q(\phi) \ln \frac{q(\phi)}{p(\phi; \theta)} d\phi. \quad (7.34)$$

In the above expression of the Kullback-Leibler divergence (compare with Eq. 7.9), the probability distribution $q(\phi)$ is unknown. We can still expand the expression to obtain:

$$KL(q||p) = \langle \ln q(\phi) \rangle_{q(\phi)} - \langle \ln p(\phi; \theta) \rangle_{q(\phi)}. \quad (7.35)$$

An important observation is that the term $\langle \ln q(\phi) \rangle_{q(\phi)}$ is constant, as it has no dependence on the variational parameters θ , and the minimization of the Kullback-Leibler divergence is then equivalent to maximizing the second right-hand term, under the training data:

$$\langle \ln p(\phi; \theta) \rangle_{q(\phi)} = \frac{1}{N} \sum_x \ln p(\phi^{(x)}; \theta), \quad (7.36)$$

where N denotes the number of training data x . The quantity $\ln p(\phi; \theta)$ can be recognized as a log-likelihood and its derivative in terms of a variational parameter θ is then equal to:

$$\frac{\partial \ln p(\phi; \theta)}{\partial \theta} = \frac{\partial}{\partial \theta} \left[\ln \frac{\exp[-S(\phi; \theta)]}{\int_{\phi} \exp[-S(\phi; \theta)] d\phi} \right] \quad (7.37)$$

$$= \frac{\partial}{\partial \theta} \left[\ln \exp[-S(\phi; \theta)] - \ln \int_{\phi} \exp[-S(\phi; \theta)] d\phi \right] \quad (7.38)$$

$$= \frac{\partial}{\partial \theta} (-S(\phi; \theta)) - \frac{\int_{\phi} \frac{\partial}{\partial \theta} (-S(\phi; \theta)) \exp[-S(\phi; \theta)] d\phi}{\int_{\phi} \exp[-S(\phi; \theta)] d\phi} \quad (7.39)$$

$$= \frac{\partial}{\partial \theta} (-S(\phi; \theta)) - \int_{\phi} p(\phi; \theta) \frac{\partial (-S(\phi; \theta))}{\partial \theta} d\phi \quad (7.40)$$

$$= \frac{\partial}{\partial \theta} (-S(\phi; \theta)) - \left\langle \frac{\partial}{\partial \theta} (-S(\phi; \theta)) \right\rangle_{p(\phi; \theta)}. \quad (7.41)$$

To solve this optimization problem pertinent to the log-likelihood or, equivalently, the minimization of the Kullback-Leibler divergence of Eq. (7.34), we will utilize again

a gradient-based approach, see Eq. (7.23), where now the loss function \mathcal{L} is replaced by:

$$\mathcal{L} = -\frac{\partial \ln p(\phi; \theta^{(t)})}{\partial \theta^{(t)}}. \quad (7.42)$$

To verify that the minimization of the Kullback-Leibler divergence enables the learning of an empirical probability distribution $q(\phi)$ by utilizing the probability distribution $p(\phi; \theta)$ we will conduct a proof-of-principle demonstration by considering as $q(\phi)$ a Gaussian distribution with $\mu = -0.5$ and $\sigma = 0.05$. Specifically, we first sample data from a Gaussian distribution to construct $q(\phi)$ and our aim is now to reproduce this data based on $p(\phi; \theta)$, without introducing any information about $q(\phi)$ within the expressions.

Practically, the method works as follows: we position the obtained data on the lattice, by replacing the degrees of freedom with the data, and calculate the quantity $\frac{\partial}{\partial \theta}(-S(\phi; \theta))$. We then randomly initialize the degrees of freedom of the system and conduct a Markov chain Monte Carlo simulation to obtain a set of samples from which we calculate the second term $\langle \frac{\partial}{\partial \theta}(-S(\phi; \theta)) \rangle_{p(\phi; \theta)}$. We then subtract the terms to update the parameters θ . Eventually we learn the optimal values of the parameters θ in the ϕ^4 action that are able to reproduce the data as configurations in the equilibrium distribution. We remark that one does not need to initiate a Markov chain at each step of the training process but one can retain the last state from the previous Markov chain as the initial state of the next step, and then arrive at equilibrium on a small number of sampling steps.

We recall that the action of the ϕ^4 scalar field theory is Z_2 invariant so we anticipate that the empirical data, which have a negative mean $\mu = -0.5$, are equiprobable in being reproduced with the equivalent data of a positive mean $\mu = +0.5$. The results, as obtained from the probability distribution of the ϕ^4 Markov random field after the training is completed, are depicted in Fig. 7.8. We observe the anticipated behaviour, namely that the symmetric data can be reproduced. One way to remove this feature is via the introduction of a term $\sum_i r_i \phi_i$ which breaks the symmetry of the system explicitly. By including this term in an action

$$S_b = S + \sum_i r_i \phi_i, \quad (7.43)$$

the system favors states that are either positive or negative, and the machine learning algorithm is therefore always able to reproduce the correct probability distribution with a negative mean. This is additionally depicted in Fig. 7.8.

We remark that Markov random fields are machine learning algorithms that have been used extensively in image analysis, image segmentation and computer vision.

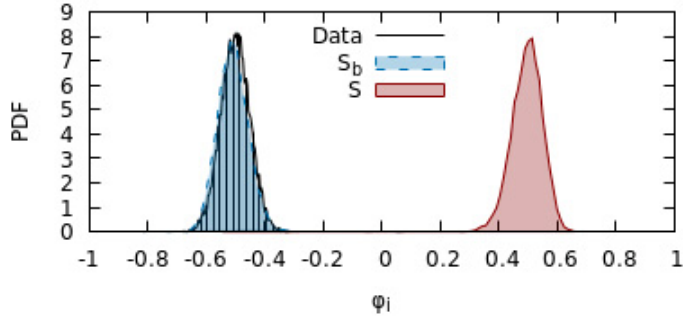


Figure 7.8: Probability density function versus the lattice value ϕ_i for the action $S(\phi; \theta)$ and an action $S_b(\phi; \theta)$ that includes a symmetry-breaking term.

Here, we will investigate if we can map an image to a ϕ^4 action that will emerge as a configuration in the equilibrium distribution $p(\phi; \theta)$ of the ϕ^4 Markov random field.

We consider an image from the CIFAR-10 dataset [93], and we aim to learn the optimal values of the coupling constants θ in the inhomogeneous action of the ϕ^4 theory that are able to reproduce the image in the equilibrium distribution. Following exactly the same procedure as in the case of the Gaussian distribution, the original image and the equilibration of the trained Markov random field are depicted in Fig. 7.9. We observe that the image emerges as a configuration in the equilibrium distribution. Consequently, it is now possible to explore conventional applications of Markov random fields in the research field of computer science using the ϕ^4 scalar field theory.

7.5 ϕ^4 neural networks

An important factor which contributes to the success of machine learning is the conception of deep architectures. Generally, one is interested in constructing an architecture that iteratively maps input data into consecutive layers which comprise a set of hidden variables. Here, we will demonstrate that neural network architectures with hidden variables can be derived from the ϕ^4 scalar field theory.

Our aim is to establish a connection between the ϕ^4 machine learning algorithms and neural network architectures that have been extensively used in computer science. We now consider that part of the degrees of freedom of a ϕ^4 scalar field theory correspond to visible variables ϕ_i and the remaining degrees of freedom correspond to hidden variables h_j . We then utilize a bipartite graph, see Fig. 7.10, and remove

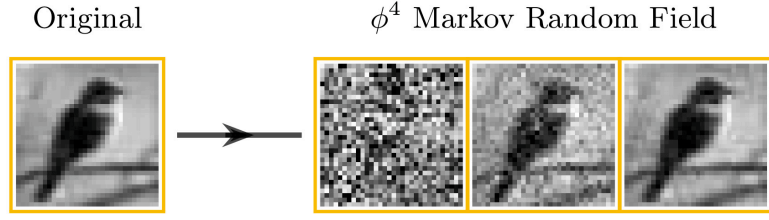


Figure 7.9: Original image (left) and equilibration of the trained ϕ^4 Markov random field (right).

intralayer interactions, both between the visible variables as well as between the hidden variables. The remaining interactions are then exclusively between the visible ϕ and the hidden h variables. We have therefore arrived at a machine learning architecture that it is able to model continuous-valued data and which can be recognized as a variant of a restricted Boltzmann machine. The system is described by a joint probability distribution $p(\phi, h; \theta)$ and the resulting action $S(\phi, h; \theta)$ is:

$$S(\phi, h; \theta) = - \sum_{i,j} w_{ij} \phi_i h_j + \sum_i r_i \phi_i + \sum_i a_i \phi_i^2 \quad (7.44)$$

$$+ \sum_i b_i \phi_i^4 + \sum_j s_j h_j + \sum_j m_j h_j^2 + \sum_j n_j h_j^4. \quad (7.45)$$

We remark that the ϕ^4 neural network which is defined by the action $S(\phi, h; \theta)$ is not only a variant of a restricted Boltzmann machine but it can additionally be viewed as a generalization of standard restricted Boltzmann machine architectures [94, 87, 95, 96]. If we select the parameters $b_i = n_j = 0$ we obtain a Gaussian-Gaussian restricted Boltzmann machine. Another architecture, specifically the Gaussian-Bernoulli restricted Boltzmann machine, is obtained if $b_i = n_j = m_j = 0$ and $h_j \in \{-1, 1\}$. Finally, the Bernoulli-Bernoulli restricted Boltzmann machine is obtained by setting $a_i = b_i = m_j = n_j = 0$ and $\phi_i, h_j \in \{-1, 1\}$. However, in this thesis we will focus solely on the ϕ^4 neural network of action $S(\phi, h; \theta)$.

We will now highlight certain properties that the ϕ^4 neural network satisfies in relation to its joint probability distribution which is given by

$$p(\phi, h; \theta) = \frac{\exp[-S(\phi, h; \theta)]}{\int_{\phi, \mathbf{h}} \exp[-S(\phi, \mathbf{h}; \theta)] d\phi d\mathbf{h}}. \quad (7.46)$$

The joint probability distribution $p(\phi, h; \theta)$ can be marginalized over either the

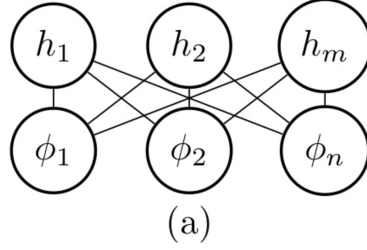


Figure 7.10: A bipartite graph used to represent a ϕ^4 neural network, where ϕ and h are the visible and hidden variables, respectively.

hidden h or the visible ϕ variables to obtain two marginal probability distributions $p(\phi; \theta)$ and $p(h; \theta)$, respectively, as:

$$p(\phi; \theta) = \int_{\mathbf{h}} p(\phi, \mathbf{h}; \theta) d\mathbf{h} = \frac{\int_{\mathbf{h}} \exp[-S(\phi, \mathbf{h}; \theta)] d\mathbf{h}}{\int_{\phi, \mathbf{h}} \exp[-S(\phi, \mathbf{h}; \theta)] d\phi d\mathbf{h}}, \quad (7.47)$$

$$p(h; \theta) = \int_{\phi} p(\phi, h; \theta) d\phi = \frac{\int_{\phi} \exp[-S(\phi, h; \theta)] d\phi}{\int_{\phi, \mathbf{h}} \exp[-S(\phi, \mathbf{h}; \theta)] d\phi d\mathbf{h}}. \quad (7.48)$$

In addition we can directly define conditional probability distributions, for instance the conditional probability distribution of the visible variables ϕ given the hidden variables h

$$p(\phi|h; \theta) = \frac{p(\phi, h; \theta)}{p(h; \theta)} = \frac{\exp[-S(\phi, h; \theta)] dh}{\int_{\phi} \exp[-S(\phi, h; \theta)] d\phi} \quad (7.49)$$

$$= \frac{\exp[\sum_{i,j} w_{ij} \phi_i h_j - \sum_i r_i \phi_i - \sum_i a_i \phi_i^2 - \sum_i b_i \phi_i^4 - \sum_j s_j h_j - \sum_j m_j h_j^2 - \sum_j n_j h_j^4]}{\int_{\phi} \exp[\sum_{i,j} w_{ij} \phi_i h_j - \sum_i r_i \phi_i - \sum_i a_i \phi_i^2 - \sum_i b_i \phi_i^4 - \sum_j s_j h_j - \sum_j m_j h_j^2 - \sum_j n_j h_j^4] d\phi} \quad (7.50)$$

$$= \frac{\prod_i \exp[\phi_i \sum_j w_{ij} h_j - r_i \phi_i - a_i \phi_i^2 - b_i \phi_i^4]}{\int_{\phi} \prod_i \exp[\phi_i \sum_j w_{ij} h_j - r_i \phi_i - a_i \phi_i^2 - b_i \phi_i^4] d\phi} \quad (7.51)$$

$$= \prod_i \frac{\exp[\phi_i \sum_j w_{ij} h_j - r_i \phi_i - a_i \phi_i^2 - b_i \phi_i^4]}{\int_{\phi_i} \exp[\phi_i \sum_j w_{ij} h_j - r_i \phi_i - a_i \phi_i^2 - b_i \phi_i^4] d\phi_i} \quad (7.52)$$

$$= \prod_i p(\phi_i|h; \theta). \quad (7.53)$$

Equivalently, one obtains the conditional probability distribution of the hidden variables as:

$$p(h|\phi; \theta) = \frac{p(\phi, h; \theta)}{p(\phi; \theta)} = \frac{\exp[-S(\phi, h; \theta)]}{\int_{\mathbf{h}} \exp[-S(\phi, \mathbf{h}; \theta)] d\mathbf{h}} = \prod_j p(h_j|\phi; \theta). \quad (7.54)$$

Having obtained the above expressions we can now discuss how the ϕ^4 neural network can be utilized to complete machine learning tasks and how the implementations differ in comparison with the ϕ^4 Markov random field.

7.5.1 Learning with predefined data

The setting that we are interested in is analogous to the setting that we explored for the ϕ^4 Markov random field in relation to learning with predefined data. We consider again that we have a set of data which correspond to an empirical probability distribution $q(\phi)$ and define again the analogous Kullback-Leibler divergence which is repeated here for convenience:

$$KL(q||p) = \int_{-\infty}^{\infty} q(\phi) \ln \frac{q(\phi)}{p(\phi; \theta)} d\phi. \quad (7.55)$$

Nevertheless, when working with ϕ^4 neural networks that are described by a joint probability distribution $p(\phi, h; \theta)$ there exists a major difference in comparison to the ϕ^4 Markov random field. The difference is, that now, we are interested in minimizing the Kullback-Leibler divergence between the empirical probability distribution $q(\phi)$ and the marginal probability distribution of the ϕ^4 neural network $p(\phi; \theta)$, as given by Eq. (7.47). The reason is that we are interested in mapping the input data to the degrees of freedom within the visible layer of the ϕ^4 neural network and, additionally, in obtaining the data in the sampling process exclusively from the visible layer of the ϕ^4 neural network. As a result, the hidden layer will then extract dependencies on the input data during the training process.

What we aim to achieve, from a practical perspective, is conceptually analogous to what we discussed before in the case of the ϕ^4 Markov random field, and we will once again rely on a gradient-based approach to optimize the variational parameters in relation to the log-likelihood. Since the marginal probability distribution $p(\phi; \theta)$ of the ϕ^4 neural network differs from the probability distribution of the ϕ^4 Markov random field we arrive at a different expression for the case of the ϕ^4 neural network:

$$\frac{\partial \ln p(\phi; \theta)}{\partial \theta} = \frac{\partial}{\partial \theta} \left[\ln \frac{\int_{\mathbf{h}} \exp[-S(\phi, \mathbf{h}; \theta)] d\mathbf{h}}{\int_{\phi, \mathbf{h}} \exp[-S(\phi, \mathbf{h}; \theta)] d\phi d\mathbf{h}} \right] \quad (7.56)$$

$$= \frac{\partial}{\partial \theta} \left[\ln \int_{\mathbf{h}} \exp[-S(\phi, \mathbf{h}; \theta)] d\mathbf{h} - \ln \int_{\phi, \mathbf{h}} \exp[-S(\phi, \mathbf{h}; \theta)] d\phi d\mathbf{h} \right] \quad (7.57)$$

$$= \frac{\int_{\mathbf{h}} \frac{\partial}{\partial \theta} (-S(\phi, \mathbf{h}; \theta)) \exp[-S(\phi, \mathbf{h}; \theta)] d\mathbf{h}}{\int_{\mathbf{h}} \exp[-S(\phi, \mathbf{h}; \theta)] d\mathbf{h}} - \frac{\int_{\phi, \mathbf{h}} \frac{\partial}{\partial \theta} (-S(\phi, \mathbf{h}; \theta)) \exp[-S(\phi, \mathbf{h}; \theta)] d\phi d\mathbf{h}}{\int_{\phi, \mathbf{h}} \exp[-S(\phi, \mathbf{h}; \theta)] d\phi d\mathbf{h}} \quad (7.58)$$

$$= \int_{\mathbf{h}} p(\mathbf{h}|\phi; \theta) \frac{\partial}{\partial \theta} (-S(\phi, \mathbf{h}; \theta)) d\mathbf{h} - \int_{\phi, \mathbf{h}} p(\phi, \mathbf{h}; \theta) \frac{\partial}{\partial \theta} (-S(\phi, \mathbf{h}; \theta)) d\phi d\mathbf{h} \quad (7.59)$$

$$= \left\langle \frac{\partial}{\partial \theta} (-S(\phi, h; \theta)) \right\rangle_{p(h|\phi; \theta)} - \left\langle \frac{\partial}{\partial \theta} (-S(\phi, h; \theta)) \right\rangle_{p(\phi, h; \theta)}. \quad (7.60)$$

For completeness, we clarify that in standard implementations of restricted Boltzmann machines approximations are introduced to calculate the final expression. Specifically, a commonly used approximation is that of contrastive divergence [97, 98]. The visible units are set equal to the value of a training example $\phi^{(x)}$ and then hidden units $h^{(x)}$ are sampled based on $p(h|\phi^{(x)})$. Based on the values of the hidden units one then would sample $\phi^{(x+1)}$ and repeat the process for k iterations:

$$CD_k = \left\langle \frac{\partial}{\partial \theta} (-S(\phi^{(0)}, h; \theta)) \right\rangle_{p(h|\phi^{(0)}; \theta)} - \left\langle \frac{\partial}{\partial \theta} (-S(\phi^{(k)}, h; \theta)) \right\rangle_{p(h|\phi^{(k)}; \theta)}. \quad (7.61)$$

We remark that even though contrastive divergence is an approximation it yields accurate results even when the number of steps is taken equal to $k = 1$.

We will now conduct a proof-of-principle demonstration to verify that the ϕ^4 neural network with action $S(\phi, h; \theta)$, given by Eq. (7.44), is able to accurately learn data. Specifically, we consider the first forty examples from the Olivetti faces dataset¹ that we present as input to the visible layer of the ϕ^4 neural network. We then train the machine learning algorithm by optimizing the variational parameters based on the derivatives of the log-likelihood in Eq. (7.56).

Our aim is to now discover if the neural network has learned some form of meaningful features in the hidden layer. One way to achieve this is by observing the values of the weights w_{ij} for a fixed j , which connect a hidden variable with each of the

¹This data set contains a set of face images taken between April 1992 and April 1994 at AT&T Laboratories Cambridge.



Figure 7.11: Example weights w_{ij} for a fixed j . See text for more details.

visible variables, and which are depicted in Fig. 7.11. We observe that the machine learning algorithm has learned features which resemble abstract face shapes and characteristics, therefore demonstrating that the ϕ^4 neural network is able to accurately extract meaningful dependencies on a set of input data.

7.6 Discussion

In this chapter we derived machine learning algorithms from quantum field theories. Specifically, we established, via the Hammersley-Clifford theorem, that the ϕ^4 scalar field theory on a square lattice is equivalent to a Markov random field. Markov random fields are a special case of probabilistic graphical models, specifically undirected graphical models that satisfy the local Markov property. Based on this equivalence, we then derived ϕ^4 neural networks that generalize a certain class of neural network architectures, namely restricted Boltzmann machines. By utilizing the ϕ^4 machine learning algorithms, we then presented applications related to physics and computer science for two cases of learning, specifically with or without a set of predefined data.

In summary, the derivation of machine learning algorithms from quantum field theories opens up the opportunity to investigate machine learning directly within lattice field theory. This equivalence is established via the Markov property, a mathematical condition that additionally emerges within constructive quantum field theory, specifically in relation to the construction of quantum fields in Minkowski space based on Markov fields in Euclidean space. As a result, the current work solidifies a rigorous connection between the research fields of machine learning, statistical physics, probability theory, lattice and constructive quantum field theory, and opens up the opportunity to directly investigate machine learning within physics.

Chapter 8

Conclusions

In this thesis we investigated the practical implications of interpreting physically functions derived from neural networks and we additionally established a connection between the research fields of machine learning and of quantum field theory. We will briefly review the contributions before concluding by highlighting potential future research directions.

In Chapter 3 we demonstrated that neural network functions can be interpreted as statistical-mechanical observables by being associated to a Boltzmann weight. We utilized this perspective to obtain machine learning predictions in extended regions of a system's parameter space. This was achieved with the use of single histogram reweighting and therefore without requiring additional data. Furthermore, we demonstrated that neural network functions act as effective order parameters, and we extracted multiple critical exponents as well as the critical inverse temperature of the two-dimensional Ising model by relying exclusively on quantities derived from the neural network implementation.

In Chapter 4 we established, based on the use of transfer learning, that functions learned from machine learning algorithms on simple systems, such as the Ising model, can be utilized to predict the phase diagram of more complicated systems, such as the Potts models or the ϕ^4 scalar field theory. This is achieved even under a change of the universality class or the order of the phase transition, as well as when the degrees of freedom are non-binary or continuous. In addition we extended single histogram reweighting for neural network functions to the multiple histogram method, thus enabling the scanning of a larger region of parameter space in order to discover unknown phase transitions. We then utilized a neural network to calculate critical exponents and the critical squared mass of the two-dimensional ϕ^4 scalar field theory.

In Chapter 5 we introduced machine learning algorithms as physical terms within

Hamiltonians, by coupling them to a fictitious field and expressing them in relation to the system's partition function. We observed that the neural network field is able to induce an order-disorder phase transition in the Ising model, in contrast to the magnetic field of the conventional order parameter which always induces explicit symmetry breaking. Furthermore we were able to establish a Hamiltonian-agnostic reweighting approach, via the inclusion of neural network functions as terms within Hamiltonians, that we utilized to study the system's phase transition. Specifically, we implemented the real-space renormalization group to extract the two critical exponents related to the relevant operators and the critical point for the phase transition of the two-dimensional Ising model.

In Chapter 6 we utilized machine learning to construct inverse renormalization group transformations that can be applied iteratively to arbitrarily increase the size of the system. Starting from lattice sizes as small as 8^2 for the case of the two-dimensional ϕ^4 scalar field theory, we applied the transformations to produce systems with lattice size up to 512^2 in absence of the critical slowing down effect. We showed that the inverse transformations induce inverse renormalization group flows in the parameter space of the ϕ^4 scalar field theory that drive the system closer to its critical point. In addition, we utilized the inverse transformations to extract the critical exponents of the magnetization and of the magnetic susceptibility for the two-dimensional ϕ^4 scalar field theory.

In Chapter 7 we derived machine learning algorithms and neural networks from quantum field theories. Specifically, we demonstrated via the Hammersley-Clifford theorem that the ϕ^4 scalar field theory is mathematically equivalent to a Markov random field. We then derived ϕ^4 neural networks that generalize a certain class of neural network architectures, namely restricted Boltzmann machines. Finally, we explored proof-of-principle numerical applications pertinent to physics or computer science by utilizing the ϕ^4 Markov random fields and ϕ^4 neural networks.

Potential future research directions, related to the physical interpretation of machine learning algorithms, can explore the construction of effective order parameters with the use of machine learning in systems where conventional order parameters are absent or unknown. Examples of such systems include topological superconductivity or the finite-temperature phase transition at finite quark mass in quantum chromodynamics. The inclusion of neural network functions within Hamiltonians additionally enables a certain control over a system by breaking or restoring its symmetry and inducing a phase transition. As a result, further exploration of this research direction might alter our understanding of how machine learning algorithms can affect systems when they are allowed to interact with them.

On the contributions pertinent to the inverse renormalization group, one can in-

stantly explore the inversion of standard renormalization group transformations with the use of machine learning in physically relevant quantum field theories. As a result, one might be able to study quantum field theories, which are computationally demanding to simulate, in absence of the critical slowing down effect. In addition, it will be of interest to explore the construction of inverse renormalization group transformations that do not utilize machine learning. Such implementations, which would be completely interpretable, could provide insights pertinent to the structure of the inverse renormalization group.

Through the derivation of machine learning algorithms from quantum field theories one is able to map the solution of a machine learning problem to the action of a quantum field theory. This instantly suggests that insights into machine learning can be obtained by using exclusively tools available within theoretical physics. In addition, the proof of the Markov property for the ϕ^4 scalar field theory solidifies a rigorous connection between the research fields of machine learning, probability theory, statistical mechanics, lattice and constructive quantum field theory. As a result, cross-fertilization between these research fields can be envisaged. For instance, the theorems pertinent to Markov random fields are now directly extendable to the ϕ^4 scalar field theory. Finally, one can investigate quantum field-theoretic machine learning algorithms within the theory of disordered systems, where substantial advances towards understanding machine learning have been established in the recent years.

In conclusion, machine learning implementations can undeniably offer tremendous benefits to enhance our insights into physical theories that we utilize to further our understanding of the world that we live in. Nevertheless, the opposite direction, which becomes accessible by posing the question of how can physics further enhance our understanding of machine learning, definitely constitutes an exciting prospect. One can then envisage a multitude of research advances at the intersection of physics and machine learning, to be discovered in the upcoming years.

Appendix A

Architectures and simulation details

For Chapter 3 the training dataset used to create the neural network function f comprises 10^3 configurations per inverse temperature, where 10^2 configurations are used in a cross-validation set. The range of inverse temperatures is $0.32, \dots, 0.41$ in the symmetric phase and $0.47, \dots, 0.56$ in the broken-symmetry phase with step 0.01. Consequently the training dataset does not comprise configurations close to the critical point $\beta_c = 0.440687$. The configurations have been sampled with the Wolff algorithm [99]. The architecture of the convolutional neural network consists of a two-dimensional convolutional layer with 64 with size 2×2 and stride 2, followed by a rectified linear unit (ReLU) nonlinear function. The subsequent layers then comprise a 2×2 max-pooling function followed by a fully-connected layer with 64 ReLUs and the output layer which includes two units and a softmax activation function. The convolutional neural network is trained for lattice sizes $L = 128, \dots, 760$ using Tensorflow and the Keras library [100]. The training is conducted with the Adam algorithm [101], a learning rate of 10^{-4} for $L \leq 256$ that is reduced by a factor of 10 for $L \geq 256$, and a mini-batch size of 12.

For Chapter 4 the Ising-trained convolutional neural network architecture is identical to the one described above. The new neural architecture was trained on the two-dimensional ϕ^4 scalar field theory on values of the squared mass $-1.09, \dots, -1.00$ and $-0.90, \dots, -0.81$ with step size 0.01. The configurations have been sampled by a combination of the Metropolis and the Wolff algorithms [102, 44, 45, 99].

For Chapter 5 the fully-connected architecture comprises a fully-connected layer with 32 neurons and a ReLU function, followed by another fully-connected layers with 2 neurons and a softmax function. The training was conducted with the Adam

algorithm, a learning rate of 10^{-4} , and batch size 8. The 10^3 configurations per each inverse temperature are sampled with the Wolff algorithm. The training range is $0.27, \dots, 0.36$ in the symmetric phase and $0.52, \dots, 0.61$ in the broken-symmetry phase, with step size 0.01.

For Chapter 6 we applied a standard renormalization group transformation on a system of lattice size $L = 32$ to obtain a rescaled system of size $L' = 16$. We then applied 128 transposed convolutions with stride 2 and filter size 2×2 , followed by a final convolution of stride 1 and filter size 2×2 on the rescaled configurations of $L' = 16$. These produced a set of model configurations with size $L_m = 32$. We then minimized the mean squared error function between the configurations of the model system L_m and the configurations of the original system L . The optimization is completed with the Adam algorithm, a learning rate of 3×10^{-4} , and a batch size of 8. The original configurations were sampled with a combination of the Metropolis and Wolff algorithms.

For Chapter 7, during training of the ϕ^4 machine learning algorithms configurations were sampled with the Metropolis algorithm. When empirical data were modelled, proposed degrees of freedom for the ϕ^4 algorithm were chosen uniformly in the range that the data reside, thus guaranteeing that every state is reachable under an arbitrary number of sampling steps. We emphasize that for the training of the ϕ^4 machine learning algorithms one does not need to initiate a new Markov chain at each step or epoch t but one can retain one Markov chain for the entire training process. In Figs. 7.3 and 7.4 the chosen learning rate is 10^{-3} and 10^{-2} , respectively. The update of the parameters θ is conducted based on 50 samples. The size of the original image in Fig. 7.9 is 32×32 , where each site has values in the range $[-1, 1]$. The learning rate is 0.1, and the number of epochs is 4×10^4 . In Fig. 7.8 the parameters are a learning rate of 0.1, a batch size of 4, and the training was conducted for 400 epochs. The ϕ^4 neural network used to produce Fig. 7.11 was trained for 10^4 epochs. It comprises 4096 visible variables and 32 hidden variables, and the training parameters are a learning rate of 0.1 and a batch size of 5.

Appendix B

Error Analysis

In this thesis two types of error analysis techniques are used, namely a bootstrap and a binning approach [10]. For the bootstrap method, each dataset is resampled 10^3 times and the error σ for an observable O is obtained by:

$$\sigma = \sqrt{\overline{O^2} - \overline{O}^2}, \quad (\text{B.1})$$

where the quantities are calculated on the resampled datasets.

For the binning error analysis technique, we separate the dataset into $n_b = 10$ bins and obtain the error of an observable O as:

$$\sigma = \sqrt{\frac{1}{n_b - 1}(\overline{O^2} - \overline{O}^2)}. \quad (\text{B.2})$$

In this thesis most of the datasets used to produce results comprise 10^5 uncorrelated configurations.

We emphasize that calculations of the critical point and the critical exponents based on the finite size scaling analyses of Chapters 3 and 4 are conducted with the use of gnuplot which treats errors as relative weights, rescaling the uncertainties on the fit results to report what they would be if $\chi^2/\text{dof} = 1$ exactly. As a result, the true uncertainties of quantities such as the critical squared mass μ_c^2 and the correlation length exponent ν are approximately 15% larger.

Bibliography

- [1] Jean Zinn-Justin. *Quantum Field Theory and Critical Phenomena*. Oxford University Press, Oxford, 2002.
- [2] J. Glimm and A. Jaffe. *Quantum Physics: A Functional Integral Point of View*. Springer, New York, NY, 1987.
- [3] Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, Cambridge, MA, USA, 2016. <http://www.deeplearningbook.org>.
- [4] Giuseppe Carleo, Ignacio Cirac, Kyle Cranmer, Laurent Daudet, Maria Schuld, Naftali Tishby, Leslie Vogt-Maranto, and Lenka Zdeborová. Machine learning and the physical sciences. *Reviews of Modern Physics*, 91(4), Dec 2019.
- [5] J J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982.
- [6] E Gardner. Maximum storage capacity in neural networks. *Europhysics Letters (EPL)*, 4(4):481–485, aug 1987.
- [7] E Gardner. The space of interactions in neural network models. *Journal of Physics A: Mathematical and General*, 21(1):257–270, jan 1988.
- [8] Ernst Ising. Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik*, 31(1):253–258, Feb 1925.
- [9] Lars Onsager. Crystal statistics. i. a two-dimensional model with an order-disorder transition. *Phys. Rev.*, 65:117–149, Feb 1944.
- [10] M. E. J. Newman and G. T. Barkema. *Monte Carlo methods in statistical physics*. Clarendon Press, Oxford, 1999.

- [11] J.J. Binney, N.J. Dowrick, A.J. Fisher, and M.E.J. Newman. *The Theory of Critical Phenomena: An Introduction to the Renormalization Group*. Oxford Science Publ. Clarendon Press, 1992.
- [12] R.J. Baxter. *Exactly solved models in statistical mechanics*. 1982.
- [13] A. Milchev, D. W. Heermann, and K. Binder. Finite-size scaling analysis of the ϕ^4 field theory on the square lattice. *Journal of Statistical Physics*, 44(5):749–784, Sep 1986.
- [14] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning, 2018.
- [15] Juan Carrasquilla and Roger G. Melko. Machine learning phases of matter. *Nature Physics*, 13(5):431–434, 2017.
- [16] Evert P. L. van Nieuwenburg, Ye-Hua Liu, and Sebastian D. Huber. Learning phase transitions by confusion. *Nature Physics*, 13(5):435–439, 2017.
- [17] Akinori Tanaka and Akio Tomiya. Detection of phase transition via convolutional neural networks. *Journal of the Physical Society of Japan*, 86(6):063001, 2017.
- [18] Frank Schindler, Nicolas Regnault, and Titus Neupert. Probing many-body localization with neural networks. *Phys. Rev. B*, 95:245134, Jun 2017.
- [19] Kelvin Ch'ng, Juan Carrasquilla, Roger G. Melko, and Ehsan Khatami. Machine learning phases of strongly correlated fermions. *Phys. Rev. X*, 7:031038, Aug 2017.
- [20] Peter Broecker, Fakher F. Assaad, and Simon Trebst. Quantum phase recognition via unsupervised machine learning, 2017.
- [21] Matthew J. S. Beach, Anna Golubeva, and Roger G. Melko. Machine learning vortices at the kosterlitz-thouless transition. *Phys. Rev. B*, 97:045207, Jan 2018.
- [22] Giacomo Torlai and Roger G. Melko. Learning thermodynamics with boltzmann machines. *Phys. Rev. B*, 94:165134, Oct 2016.
- [23] Philippe Suchsland and Stefan Wessel. Parameter diagnostics of phases and phase transition learning by neural networks. *Phys. Rev. B*, 97:174435, May 2018.

- [24] Zhenyu Li, Mingxing Luo, and Xin Wan. Extracting critical exponents by finite-size scaling with convolutional neural networks. *Phys. Rev. B*, 99:075418, Feb 2019.
- [25] Qi Ni, Ming Tang, Ying Liu, and Ying-Cheng Lai. Machine learning dynamical phase transitions in complex networks. *Phys. Rev. E*, 100:052312, Nov 2019.
- [26] Evert van Nieuwenburg, Eyal Bairey, and Gil Refael. Learning phase transitions from dynamics. *Phys. Rev. B*, 98:060301, Aug 2018.
- [27] Jordan Venderley, Vedika Khemani, and Eun-Ah Kim. Machine learning out-of-equilibrium phases of matter. *Phys. Rev. Lett.*, 120:257204, Jun 2018.
- [28] Yi-Ting Hsu, Xiao Li, Dong-Ling Deng, and S. Das Sarma. Machine learning many-body localization: Search for the elusive nonergodic metal. *Phys. Rev. Lett.*, 121:245701, Dec 2018.
- [29] Rodrigo A. Vargas-Hernández, John Sous, Mona Berciu, and Roman V. Krems. Extrapolating quantum observables with machine learning: Inferring multiple phase transitions from properties of a single phase. *Phys. Rev. Lett.*, 121:255702, Dec 2018.
- [30] Sebastian J. Wetzel. Unsupervised learning of phase transitions: From principal component analysis to variational autoencoders. *Phys. Rev. E*, 96:022140, Aug 2017.
- [31] Kelvin Ch'ng, Nick Vazquez, and Ehsan Khatami. Unsupervised machine learning account of magnetic transitions in the hubbard model. *Phys. Rev. E*, 97:013306, Jan 2018.
- [32] Joaquin F. Rodriguez-Nieva and Mathias S. Scheurer. Identifying topological order through unsupervised machine learning. *Nature Physics*, 15(8):790–795, 2019.
- [33] Cinzia Giannetti, Biagio Lucini, and Davide Vadacchino. Machine learning as a universal tool for quantitative investigations of phase transitions. *Nuclear Physics B*, 944:114639, 2019.
- [34] Pedro Ponte and Roger G. Melko. Kernel methods for interpretable machine learning of order parameters. *Phys. Rev. B*, 96:205146, Nov 2017.

- [35] Jonas Greitemann, Ke Liu, and Lode Pollet. Probing hidden spin order with interpretable machine learning. *Phys. Rev. B*, 99:060404, Feb 2019.
- [36] Ke Liu, Jonas Greitemann, and Lode Pollet. Learning multiple order parameters with interpretable machines. *Phys. Rev. B*, 99:104410, Mar 2019.
- [37] Dimitrios Bachtis, Gert Aarts, and Biagio Lucini. Extending machine learning classification capabilities with histogram reweighting. *Phys. Rev. E*, 102:033303, Sep 2020.
- [38] Alan M. Ferrenberg and Robert H. Swendsen. New monte carlo technique for studying phase transitions. *Phys. Rev. Lett.*, 61:2635–2638, Dec 1988.
- [39] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Trans. on Knowl. and Data Eng.*, 22(10):1345–1359, October 2010.
- [40] Kenta Shiina, Hiroyuki Mori, Yutaka Okabe, and Hwee Kuan Lee. Machine-learning studies on spin models. *Scientific Reports*, 10(1):2177, Feb 2020.
- [41] Askery Canabarro, Felipe Fernandes Fanchini, André Luiz Malvezzi, Rodrigo Pereira, and Rafael Chaves. Unveiling phase transitions with machine learning. *Phys. Rev. B*, 100:045129, Jul 2019.
- [42] Dimitrios Bachtis, Gert Aarts, and Biagio Lucini. Mapping distinct phase transitions to a neural network. *Phys. Rev. E*, 102:053306, Nov 2020.
- [43] Alan M. Ferrenberg and Robert H. Swendsen. Optimized monte carlo data analysis. *Phys. Rev. Lett.*, 63:1195–1198, Sep 1989.
- [44] David Schaich and Will Loinaz. Improved lattice measurement of the critical coupling in ϕ_2^4 theory. *Phys. Rev. D*, 79:056008, Mar 2009.
- [45] Will Loinaz and R. S. Willey. Monte carlo simulation calculation of the critical coupling constant for two-dimensional continuum φ^4 theory. *Phys. Rev. D*, 58:076003, Sep 1998.
- [46] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3320–3328. Curran Associates, Inc., 2014.

- [47] Kimihiko Fukushima and Kazumitsu Sakai. Can a CNN trained on the Ising model detect the phase transition of the q-state Potts model? *Progress of Theoretical and Experimental Physics*, 2021(6), 05 2021. 061A01.
- [48] Dimitrios Bachtis, Gert Aarts, and Biagio Lucini. Adding machine learning within hamiltonians: Renormalization group transformations, symmetry breaking and restoration. *Phys. Rev. Research*, 3:013134, Feb 2021.
- [49] Gert Aarts, Dimitrios Bachtis, and Biagio Lucini. Interpreting machine learning functions as physical observables, 2021.
- [50] Kenneth G. Wilson. Renormalization group and critical phenomena. i. renormalization group and the kadanoff scaling picture. *Phys. Rev. B*, 4:3174–3183, Nov 1971.
- [51] Kenneth G. Wilson. Renormalization group and critical phenomena. ii. phase-space cell analysis of critical behavior. *Phys. Rev. B*, 4:3184–3205, Nov 1971.
- [52] Leo P. Kadanoff. Scaling laws for ising models near T_c . *Physics Physique Fizika*, 2:263–272, Jun 1966.
- [53] Kenneth G. Wilson. The renormalization group: Critical phenomena and the kondo problem. *Rev. Mod. Phys.*, 47:773–840, Oct 1975.
- [54] Kenneth G. Wilson and Michael E. Fisher. Critical exponents in 3.99 dimensions. *Phys. Rev. Lett.*, 28:240–243, Jan 1972.
- [55] Kenneth G. Wilson and J. Kogut. The renormalization group and the epsilon expansion. *Physics Reports*, 12(2):75 – 199, 1974.
- [56] Robert H. Swendsen. Monte carlo renormalization group. *Phys. Rev. Lett.*, 42:859–861, Apr 1979.
- [57] Shang-keng Ma. Renormalization group by monte carlo methods. *Phys. Rev. Lett.*, 37:461–464, Aug 1976.
- [58] Robert H. Swendsen. Monte carlo calculation of renormalized coupling parameters. i. $d = 2$ ising model. *Phys. Rev. B*, 30:3866–3874, Oct 1984.
- [59] Dorit Ron, Robert H. Swendsen, and Achi Brandt. Inverse monte carlo renormalization group transformations for critical phenomena. *Phys. Rev. Lett.*, 89:275701, Dec 2002.

- [60] H. W. J. Blöte, J. R. Heringa, A. Hoogland, E. W. Meyer, and T. S. Smit. Monte carlo renormalization of the 3d ising model: Analyticity and convergence. *Phys. Rev. Lett.*, 76:2613–2616, Apr 1996.
- [61] Dorit Ron, Achi Brandt, and Robert H. Swendsen. Surprising convergence of the monte carlo renormalization group for the three-dimensional ising model. *Phys. Rev. E*, 95:053305, May 2017.
- [62] K. Akemi, M. Fujisaki, M. Okuda, Y. Tago, Ph. de Forcrand, T. Hashimoto, S. Hioki, O. Miyamura, T. Takaishi, A. Nakamura, and I. O. Stamatescu. Scaling study of pure gauge lattice qcd by monte carlo renormalization group method. *Phys. Rev. Lett.*, 71:3063–3066, Nov 1993.
- [63] Anna Hasenfratz. Investigating the critical properties of beyond-qcd theories using monte carlo renormalization group matching. *Phys. Rev. D*, 80:034505, Aug 2009.
- [64] Anna Hasenfratz. Conformal or walking? monte carlo renormalization group studies of $su(3)$ gauge models with fundamental fermions. *Phys. Rev. D*, 82:014506, Jul 2010.
- [65] Simon Catterall, Luigi Del Debbio, Joel Giedt, and Liam Keegan. Monte carlo renormalization group minimal walking technicolor. *Phys. Rev. D*, 85:094501, May 2012.
- [66] Th. Niemeijer and J. M. J. van Leeuwen. *Phase Transitions and Critical Phenomena*. Academic Press, 1976. edited by C. Domb and M. S. Green.
- [67] Stavros Efthymiou, Matthew J. S. Beach, and Roger G. Melko. Super-resolving the ising model with convolutional neural networks. *Phys. Rev. B*, 99:075113, Feb 2019.
- [68] Shuo-Hui Li and Lei Wang. Neural network renormalization group. *Phys. Rev. Lett.*, 121:260601, Dec 2018.
- [69] Kenta Shiina, Hiroyuki Mori, Yusuke Tomita, Hwee Kuan Lee, and Yutaka Okabe. Inverse renormalization group based on image super-resolution using deep convolutional networks. *Scientific Reports*, 11(1):9617, May 2021.
- [70] Dimitrios Bachtis, Gert Aarts, Francesco Di Renzo, and Biagio Lucini. Inverse renormalization group in quantum field theory. *Phys. Rev. Lett.*, 128:081603, Feb 2022.

- [71] Dimitrios Bachtis, Gert Aarts, and Biagio Lucini. Quantum field-theoretic machine learning. *Phys. Rev. D*, 103:074510, Apr 2021.
- [72] Dimitrios Bachtis, Gert Aarts, and Biagio Lucini. Quantum field theories, markov random fields and machine learning. *Journal of Physics: Conference Series*, 2207(1):012056, mar 2022.
- [73] Dimitrios Bachtis, Gert Aarts, and Biagio Lucini. Machine learning with quantum field theories, 2021.
- [74] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.
- [75] Andrew Blake, Pushmeet Kohli, and Carsten Rother. *Markov Random Fields for Vision and Image Processing*. The MIT Press, 2011.
- [76] Edward Nelson. Construction of quantum fields from markoff fields. *Journal of Functional Analysis*, 12(1):97 – 112, 1973.
- [77] Koji Hashimoto, Sotaro Sugishita, Akinori Tanaka, and Akio Tomiya. Deep learning and the AdS/CFT correspondence. *Phys. Rev. D*, 98:046019, Aug 2018.
- [78] Koji Hashimoto. AdS/CFT correspondence as a deep boltzmann machine. *Phys. Rev. D*, 99:106017, May 2019.
- [79] James Halverson, Anindita Maiti, and Keegan Stoner. Neural networks and quantum field theory. *Machine Learning: Science and Technology*, 2(3):035002, apr 2021.
- [80] H Erbin, V Lahoche, and D Ousmane Samary. Non-perturbative renormalization for the neural network-QFT correspondence. *Machine Learning: Science and Technology*, 3(1):015027, feb 2022.
- [81] Jaehoon Lee, Yasaman Bahri, Roman Novak, Sam Schoenholz, Jeffrey Pennington, and Jascha Sohl-dickstein. Deep neural networks as gaussian processes. 6th International Conference on Learning Representations, Vancouver, BC, Canada, 2018.
- [82] G. R. Grimmett. A theorem about random fields. *Bulletin of the London Mathematical Society*, 5(1):81–84, 1973.

- [83] J. Hammersley and P. Clifford. Markov fields on finite graphs and lattices, 1971.
- [84] C. J. Preston. Generalized gibbs states and markov random fields. *Advances in Applied Probability*, 5(2):242–261, 1973.
- [85] S. Sherman. Markov random fields and gibbs random fields. *Israel Journal of Mathematics*, 14(1):92–103, Mar 1973.
- [86] Julian Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):192–236, 1974.
- [87] Asja Fischer and Christian Igel. Training restricted boltzmann machines: An introduction. *Pattern Recognition*, 47(1):25 – 39, 2014.
- [88] Geoffrey E. Hinton. *A Practical Guide to Training Restricted Boltzmann Machines*, pages 599–619. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [89] David H. Ackley, Geoffrey E. Hinton, and Terrence J. Sejnowski. A learning algorithm for boltzmann machines. *Cognitive Science*, 9(1):147 – 169, 1985.
- [90] P. Smolensky. *Information Processing in Dynamical Systems: Foundations of Harmony Theory*, page 194–281. MIT Press, Cambridge, MA, USA, 1986.
- [91] M. Mézard, G. Parisi, and M. A. Virasoro. *Spin Glass Theory and Beyond: An Introduction to the Replica Method and Its Applications*. World Scientific, Singapore, 1987.
- [92] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.
- [93] A. Krizhevsky. Learning multiple layers of features from tiny images. 2009.
- [94] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.
- [95] KyungHyun Cho, Alexander Ilin, and Tapani Raiko. Improved learning of gaussian-bernoulli restricted boltzmann machines. In *Proceedings of the 21th International Conference on Artificial Neural Networks - Volume Part I, ICANN’11*, page 10–17, Berlin, Heidelberg, 2011. Springer-Verlag.

- [96] Jan Melchior, Nan Wang, and Laurenz Wiskott. Gaussian-binary restricted boltzmann machines for modeling natural image statistics. *PLOS ONE*, 12(2):1–24, 02 2017.
- [97] Miguel Á. Carreira-Perpiñán and Geoffrey Hinton. On contrastive divergence learning. In Robert G. Cowell and Zoubin Ghahramani, editors, *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, volume R5 of *Proceedings of Machine Learning Research*, pages 33–40. PMLR, 06–08 Jan 2005. Reissued by PMLR on 30 March 2021.
- [98] Geoffrey E. Hinton. Training Products of Experts by Minimizing Contrastive Divergence. *Neural Computation*, 14(8):1771–1800, 08 2002.
- [99] Ulli Wolff. Collective monte carlo updating for spin systems. *Phys. Rev. Lett.*, 62:361–364, Jan 1989.
- [100] Francois Chollet et al. Keras, 2015.
- [101] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.
- [102] Richard C. Brower and Pablo Tamayo. Embedded dynamics for φ^4 theory. *Phys. Rev. Lett.*, 62:1087–1090, Mar 1989.