Application of machine learning algorithms to statistical physics and biophysics.

Submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

by

MONIKA RICHTER–LASKOWSKA

UNIVERSITY OF SILESIA
IN KATOWICE

February, 2023

DECLARATION

I here by declare that the thesis entitled **“Application of machine learning algorithms to statistical physics and biophysics”** submitted by me, for the award of the degree of *Doctor of Philosophy* to University of Silesia is a record of bonafide work carried out by me under the supervision of prof. Maciej Maśka from Wrocĺaw University of Science and Technology and dr Marcin Kurpas from University of Silesia.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Chorzów

Date: 02/28/2023

Signature of the Candidate

CERTIFICATE

This is to certify that the thesis entitled **“Application of machine learning methods to the physics of phase transitions and biophysics”**, submitted by Ms Monika Richter–Laskowska, Institute of Physics, University of Silesia, for the award of the degree of *Doctor of Philosophy*, is a record of bonafide work carried out by her under my supervision, as per the code of academic and research ethics.

The contents of this report has not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfills the requirements and regulations of the University and in my opinion meets the necessary standards for submission.

Place: Chorzów

Date: 02/28/2023

Signature of the Guide
prof. dr hab. Maciej Maśka

ABSTRACT

This Ph.D. thesis is an attempt to apply ML approaches to selected problems of solid state physics and biophysics. The first part is devoted to phase transitions in physical models - the problem that has been known and studied from the beginning of condensed matter and statistical physics. Despite many efforts that have been put into defining indicators for the classification of phase transitions in numerical modeling, universal ones valid for all microscopic models do not exist so far. A similar problem is the precise determination of the critical temperature of a phase transition. Therefore, it is interesting to check how alternative methods based on *artificial intelligence* will perform in this matter. In the second part of the dissertation, we focus on the possible application of the ML techniques to biosignals describing ion channels' activity. The classical methods of kinetic analysis often fail to grasp the discriminative features of recordings corresponding to different cellular lines and various stimuli. We investigate if the ML approaches for time series analysis are able to find such signal's characteristics from which one can extract an information about the differences in ion's conformational dynamics.

Keywords: *machine learning*, *neural network*, *phase transitions*, *critical temperature*, *first and second–order phase transitions*, *ion channel*, *time series analysis*

ACKNOWLEDGEMENT

With immense pleasure and deep sense of gratitude, I wish to express my sincere thanks to my supervisors **prof. Maciej Maśka** and **Marcin Kurpas**, without their motivation and continuous encouragement, this research would not have been successfully completed.

I am also grateful to the Łukasiewicz Research Network – Krakow Institute of Technology, where I am currently employed, for allowing me to pursue with my research and giving me time and space to finish this dissertation.

I wish to extend my profound sense of gratitude to **my deceased mother** for all the sacrifices she made during my studies, constant strong belief in my abilities and also providing me with moral support and encouragement whenever required.

Last but not the least, I would like to thank my husband **Zbyszek** and my dearest friends **Paulina and Agata** for their constant encouragement and moral support along with patience and understanding.

Place: Chorzów

Date: 02/28/2023 **Monika Richter–Laskowska**

List of Terms and Abbreviations

4-AP	12 4-aminopyridine
Acc	Accuracy
AD	Anomaly Detection
AI	Artificial Intelligence
ANN	Artificial Neural Network
AUC–ROC	Area Under ROC Curve
BC	Blume–Capel model
BCE	Binary Cross Entropy
BCS-BEC	Bardeen-Cooper-Schrieffer – Bose-Einstein condensation crossover
BK	Big Potassium
BKT	Berezinskii–Kosterlitz–Thouless
C	Closed State
CDW	Charge Density Wave
ChTx	Charybdotoxin
CL	Convolutional Layer
CM	Confusion Matrix
CMP	Condensed Matter Physics
CNN	Convolutional Neural Network
CUDA	Compute Unified Device Architecture
c-XY	classical XY model
DBP	Detailed Balance Principle
DFT	Density Functional Theory
DL	Deep Learning
DTW	Dynamical Time Warping

FE	Fibroblast-Endothelium
FH	Fibrolast-Hippocampus
FK	Falicov–Kimball model
FN	False Negatives
FNN	Fully–connected Neural Network
FO	First–Order Phase Transition
FP	False Positives
FPR	False Positive Rate
GPU	Graphical Processing Unit
HE	Hippocampus-Endothelium
HTP	High–Temperature Phase
IbTx	Iberiotoxin
KNN	K-Nearest Neighbors
LBC	Learning by confusion
LTP	Low–Temperature Phase
MC	Monte–Carlo
MD	Molecular Dynamics
mitoBK	mitochondrial Big Potassium Channel
ML	Machine Learning
MSE	Mean Squared Error
Nar	Naringenin
NN	Neural Network
O	Open State
PAA	Piecewise Aggregate Approximation
PAX	Paxilline
PCA	Principle Component Analysis
PES	Potential Energy Surface
PF	Phase–Fermion model
P_{op}	Opening Probability

qP	q–state Potts model
QRLO	Quasi Long–Range Order
q-XY	quantum XY model
RBM	Restricted Boltzmann Machine
ReLU	Rectified linear unit
RG	Renormalization Group
ROC	Receiver Operating Characteristic
SLMC	Self–Learning Monte Carlo
SO	Second–Order Phase Transition
SVM	Support Vector Machine
TC	Threshold Current
TEA	Tetra-Ethyl Ammonium
TN	True Negatives
TP	True Positives
TPC	Tricritical Point
TPR	True Positive Rate
t–SNE	t–Distributed Stochastic Neighbour Embedding

TABLE OF CONTENTS

Appendices

For Removing Extra Space in List of Tables and Figures before every chapter

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

Machine learning (ML) algorithms have been widely applied in various fields of life, such as medicine, industry, and science, with physics not being an exception. The recent renaissance of ML methods is driven by enormous progress in developing massively parallel computing platforms based on supercomputers, multicore GPUs (Graphics Processing Unit), dedicated tensor processors, and parallel programming models (CUDA). Nowadays, a desktop computer equipped with a modern gaming graphics card has computational power easily exceeding hundreds of TFLOPS (TeraFLOPS), while the world-top supercomputer Sequoia built by IBM offers 20 petaFLOPS. Such an availability of high computational power opens great perspectives for condensed matter physics, where the system's complexity is the main obstacle to tackling real problems. ML methods are of special interest here since they constitute a qualitatively new approach to solid state and statistical physics. In contrast to conventional approaches to microscopic physical models, such as Monte Carlo simulations, ML methods use only a fraction of information about the system, usually snapshots of lattice configurations. For example, in the famous Falicov-Kimball model describing the interaction of itinerant electrons with localized adatoms, the snapshots correspond to the configuration of adatoms on the lattice. Electrons are completely neglected here.

This Ph.D. thesis is an attempt to apply ML approaches to selected problems of solid state physics and biophysics. The first part is devoted to phase transitions in physical models - a problem that has been known and studied from the beginning of condensed matter and statistical physics. Despite many efforts that have been made to define indicators for the classification of phase transitions in numerical modeling, so far a universal one valid for all microscopic models does not exist. A similar problem is the precise determination of the critical temperature of a phase transition. Therefore, it is interesting to check how alternative methods based on *artificial intelligence* will perform in this matter. In the second part of the dissertation, we focus on the possible application of ML techniques to biosignals describing ion channels' activity. Classical methods of kinetic analysis often fail to grasp the discriminative features of recordings corresponding to different cellular lines and various stimuli. We investigate if the ML approaches for time series analysis are able to find such signal's characteristics from which one can extract information about the differences in ion's conformational dynamics.

The dissertation is organized as follows. In Chapter 2, we introduce basic notions associated with the physics of phase transitions. The next Chapter 3 describes primary machine learning methods and their applications to some problems of condensed matter physics. Chapter 4 is divided into four sections summarizing the results of the studies conducted as part of this thesis. In Sec. 4.1, we show that the ML methods, based solely on the MC configurations and without any additional knowledge about the underlying physical model, can determine the critical temperature of the topological phase transition. However, we find that the accuracy of this prediction depends on the character of the microscopic model (quantum or classical) and the choice of configurations to be trained. Nevertheless, it is demonstrated that in all studied cases, the algorithm does not trivially learn the order parameter, but probably some more complicated local features of the configurations related to the proliferation of vortices and antivortices in the vicinity of this transition. In Sec. 4.2, we show how some simple unsupervised learning methods can identify the order parameter and hint at the nature of undergoing phase transitions. Some limitations of these algorithms and their interpretation are also discussed. Sec. 4.3 describes the applicability of the learning by confusion (LBC) scheme, a method based on the neural network, in accurate identification of the phase transition nature in the case of the Falicov–Kimball (FK) model for which the classical methods fail. As a result of the application of the LBC scheme, we obtain the characteristic performance curve. We discover that the character of this curve differs qualitatively between two kinds of phase transition, which in the FK model is driven by the value of the potential U. We show that the resulting discrepancies could be associated with the characteristic growth of defects (deviations from the fully–ordered state) occurring in this model. We prove that this phenomenon does not disappear with an increase in lattice size, which suggests that it is the result of interesting physics hidden in the FK model and not a finite–size effect. The remaining part of this chapter concentrates on applying various ML methods to the experimental data that illustrate the activity of ion channels (Sec. 4.4). In this Section, we reveal that it is not always possible to discriminate between BK channel recordings obtained at the same membrane potential but corresponding to different cell lines using classical kinetic analysis methods. However, we find that it is feasible to grasp these disparities using machine learning algorithms. In another study, we discover that some novel data mining algorithms can find qualitative differences in the activation of BK channels treated with various modulators. The last Chapter 5 briefly summarizes the scientific achievements described in previous chapters.

CHAPTER 2

The physics of phase transitions – theoretical preliminaries

2.1 States of matter

Every substance can take one of its forms called *phases*. Water, for example, can be a solid (ice), liquid, or gas, depending on pressure and temperature. Within one phase, matter can also exhibit immense complexity. For example, ice has 17 different polymorphs, and numerical simulations predict even more. The richness of the water phase diagram is presented in Fig. 2.1.

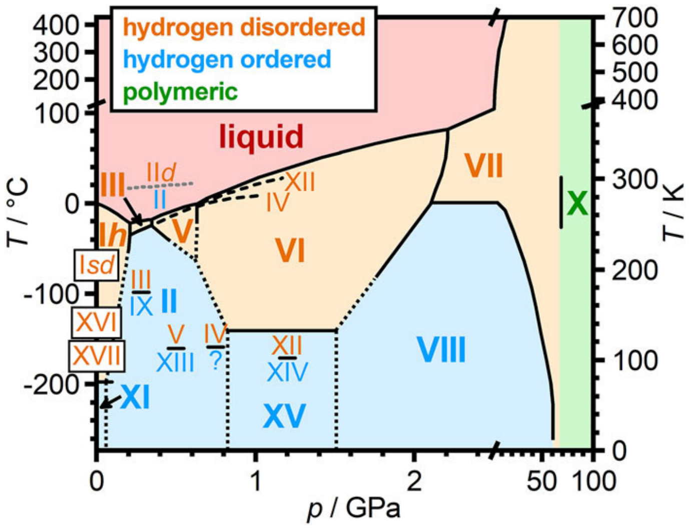


Fig. 2.1 The phase diagram of water including liquid phase (red), hydrogen–disordered (orange), hydrogen–ordered (blue) and polymeric states (green) of ice [1]. Stable and metastable states are denoted by large and small Roman numerals, respectively. The solid and dotted black lines indicate the phase boundaries, where dotted line means that the phase boundary was extrapolated. Dashed lines indicate the metastable melting lines of ices IV and XII.

Different states of matter exhibit various thermal, optical, and electronic properties. They are reflected in the values of measurable quantities such as density, heat capacity, magnetic moments, etc. Generally, a substance can pass from one state to another under the influence of some external parameter (magnetic or electric field, temperature, pressure). Such physical processes are known in the literature as *phase transitions*. In the experiment, these phenomena are identified with a drastic change in some physical observables and, in some cases, with the release of the latent heat at the critical point. Phase transitions occur in a wide variety of physical systems. They are associated with a sudden jump of some physical quantity. In the case of above–mentioned water, it is signaled by an instantaneous change in density value. Another typical example is the ferromagnet–paramagnet transition, which involves an abrupt drop in magnetization. Similarly, the conductor–superconductor phase transition is characterized by a rapid decrease in resistance.

The physics of phase transitions can be understood at the macroscopic level in terms of thermodynamics and statistical physics. We now briefly recall the most important concepts of these two branches of physics, which are essential to understand the remaining part of the thesis. Let us consider a magnetic system undergoing a phase transition governed by the first law of thermodynamics:

$$dU = TdS - MdH,$$

where U is the internal energy of the system, T stands for temperature, S is the entropy, M is the magnetic moment, and H denotes the value of an external magnetic field.

One of the most fundamental quantities in statistical physics is the canonical partition function $\mathcal{Z}(T, H)$:

$$\mathcal{Z}(T, H) = \sum_r e^{-\beta E_r(H)}, \tag{2.1}$$

where sum goes over all possible states, $\beta = \frac{1}{k_B T}$ and k_B is the Boltzmann constant. Knowing $\mathcal{Z}(T, H)$, one can derive all information about the system, including the free energy $\mathcal{F}(T, H)$:

$$\mathcal{F}(T, H) = -k_B T \log \mathcal{Z}(H, T), \tag{2.2}$$

which is necessary to find the values of thermodynamic quantities changing during a phase transition, such as entropy, magnetic susceptibility, or heat capacity (as presented in Fig. 2.2). Note that even though the introduced formalism is written in the magnetic language (so are the other physical quantities), it can be easily generalized for other systems, not only those involving magnetic interactions.

The thermodynamic quantities presented in Fig. 2.2 allow one to tackle the physics of phase transition solely on the macroscopic level. However, a microscopic description is also needed to gain a more profound understanding of the phenomenon. For this reason, it is useful to define a physical observable that can continuously trace changes in microscopic degrees of freedom upon a phase transition. The correlation function $\mathcal{C}$ is one of such quantities. In general, $\mathcal{C}$ measures the relationship between two random variables. Let us suppose that we are dealing with some physical quantity $q(x)$ that

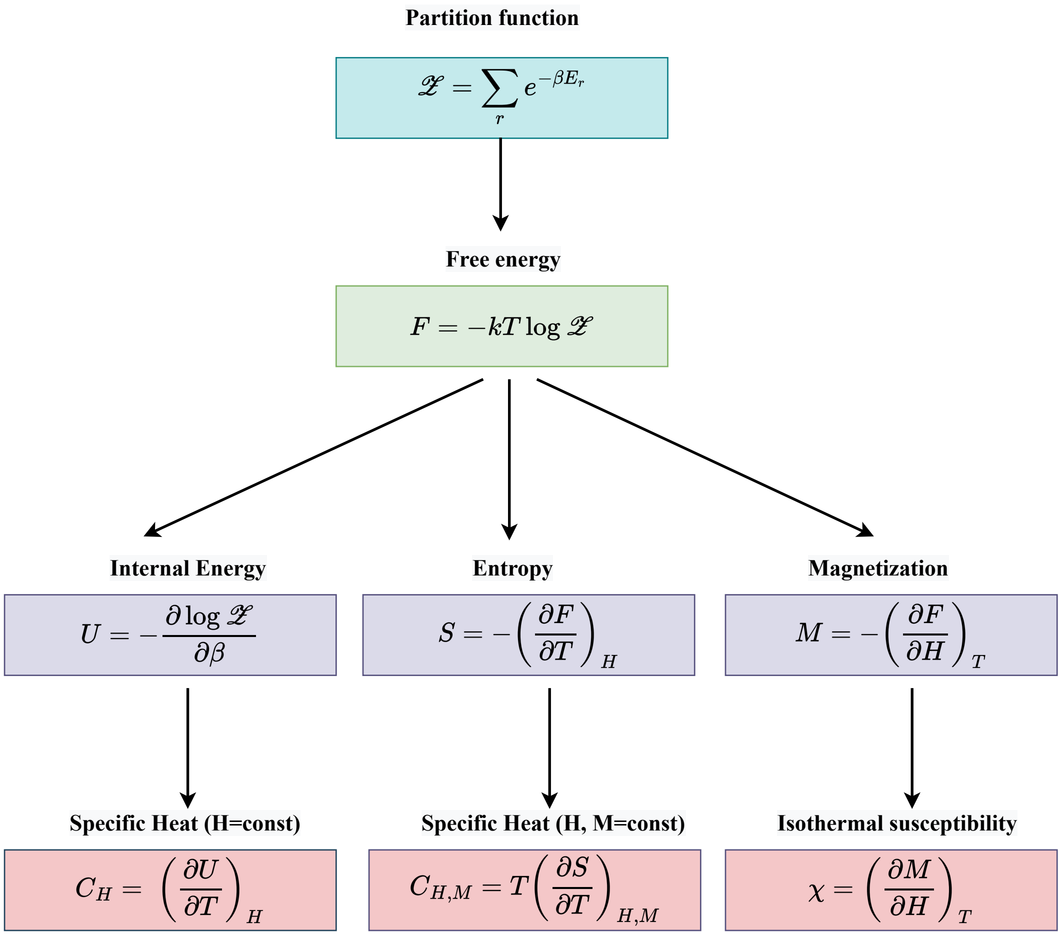

Partition function
$\mathscr{Z} = \sum_r e^{-\beta E_r}$
Free energy
$F = -kT \log \mathscr{Z}$
Internal Energy
$U = -\frac{\partial \log \mathscr{Z}}{\partial \beta}$
Entropy
$S = -\left(\frac{\partial F}{\partial T}\right)_H$
Magnetization
$M = -\left(\frac{\partial F}{\partial H}\right)_T$
Specific Heat (H=const)
$C_H = \left(\frac{\partial U}{\partial T}\right)_H$
Specific Heat (H, M=const)
$C_{H,M} = T\left(\frac{\partial S}{\partial T}\right)_{H,M}$
Isothermal susceptibility
$\chi = \left(\frac{\partial M}{\partial H}\right)_T$

Fig. 2.2 The fundamental thermodynamic variables derived from the canonical partition function $\mathcal{Z}$ for a magnetic system governed by the first law of thermodynamics: $dU = TdS - MdH$.

can take different values depending on the position x. The correlation $C(q(x_1), q(x_2))$ between $q(x_1)$ and $q(x_2)$ is defined as:

$$C(q(x_1), q(x_2)) = \langle q(x_1)q(x_2)\rangle. \tag{2.3}$$

In a similar manner, we can measure the correlation between spins s_i and s_j placed on a discrete lattice:

$$\mathcal{C}(i, j) = \mathcal{C}(r) = \langle s_i s_j\rangle - \langle s\rangle^2, \tag{2.4}$$

where $\langle . \rangle$ denotes the mean value and r stands for the distance between spins s_i and s_j that occupy lattice sites i and j, respectively.

The value $\mathcal{C}$ is not constant and changes under the influence of an external parameter. Below and above the critical point, it takes the following form [7]:

$$\mathcal{C}(i, j) = r^{-\tau} \exp^{-r/\xi}, \tag{2.5}$$

where $r = |i - j|$, ξ stands for the *correlation length* and τ denotes some constant. Such a form of $\mathcal{C}$ indicates that correlations decay exponentially with the distance between spins and vanish for sufficiently large r. In the vicinity of a phase transition, when the long–range order starts to develop in a system, ξ becomes infinite, and Eq. (2.5) can no longer describe correlations. The experiments suggest that, in this case, the correlation function decays according to the power law:

$$\mathcal{C}(i,j) \sim \frac{1}{r^{d-2+\eta}}, \tag{2.6}$$

where d stands for the dimension of the system and η is a constant called the critical exponent.

It is possible to relate the microscopic correlation function with the macroscopic magnetic susceptibility χ defined in Fig. 2.2. A straightforward calculation yields:

$$\chi = \frac{1}{kT}\big(\langle M^2\rangle - \langle M\rangle^2\big) = \frac{1}{kT}\langle\big(M - \langle M\rangle\big)^2\rangle. \tag{2.7}$$

On the other hand:

$$\langle\big(M - \langle M\rangle\big)^2\rangle = \sum_i \big(s_i - \langle s_i\rangle\big)\sum_j \big(s_j - \langle s_j\rangle\big) = \sum_{ij}\mathcal{C}(i,j) = N\sum_i \mathcal{C}(i,0), \tag{2.8}$$

which implies that:

$$\chi = \frac{N}{kT}\sum_i \mathcal{C}(i,0). \tag{2.9}$$

In the last step of evaluating Eq. (2.8), we assumed translational invariance ($\langle s_i\rangle = \langle s_j\rangle$), which is well justified for homogeneous systems. Eq. (2.8) implies that the influence of the phase transition on the microscopic correlation function $\mathcal{C}$ is also reflected in macroscopic magnetic susceptibility χ. In particular, the divergence of $\mathcal{C}$ observed at the critical point makes χ also infinite.

2.2 First and second–order phase transitions

As can be inferred from the previous Section, the phenomenon of phase transition is associated with the divergence of a physical observable. In some cases, the discontinuity appears in quantities being the first–derivatives of free energy, such as internal energy, entropy, or magnetization (Fig. 2.2 and Fig. 2.3ab). In other cases, these physical observables pass smoothly through the critical point – a non–analytical behavior is exhibited in the second–order derivatives of free energy (Fig. 2.3cd). Based on the order of the smallest derivative divergent at the critical point, Ehrenfest called these two kinds of phase transitions, *first* and *second–order phase transitions*.

Nowadays, scientists often abandon the Ehrenfest classification in favor of the

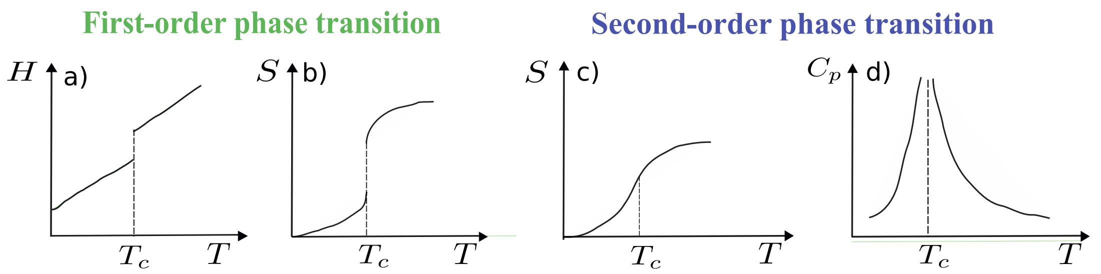


Fig. 2.3 The behavior of some thermodynamic observables around the transition point for the first–order phase transition (a),b)), second–order phase transitions (c), d)).

modern division of the phase transitions, which divides them into two types:

1. *Discontinuous phase transitions* (first–order phase transitions according to Ehrenfest) involving a release of latent heat at the transition point $Q_L = T_C \Delta S$,

2. *Continuous phase transitions* (second–order phase transitions according to Ehrenfest) not accompanied by latent heat.

With regard to the fact that both classifications are still in use in the literature, we will use the terms *first–order* and *discontinuous*, *second–order* and *continuous* interchangeably throughout this thesis.

As highlighted in the previous Section, phase transitions manifest themselves in infinite values of the correlation length, divergent susceptibility (heat capacity), and the power–law decay of the correlation function at the transition. Such a description is accurate only in the case of second–order (*continuous*) phase transitions. During a discontinuous phase transition, we observe the divergence in the first derivatives of free energy. It implies that entropy S defined as $S = -\left(\frac{\partial F}{\partial T}\right)_H$ exhibits a discontinuous jump at the critical point that leads to the release or absorption of latent heat $Q_L = T_C \Delta S$.

A characteristic feature of discontinuous phase transitions is phase coexistence, when the two phases, low and high temperature, for example, exist simultaneously. This phenomenon can be easily visualized in the example of liquid–gas transition (see Fig. 2.4). Before a liquid turns into gas, it passes through an intermediate state where both phases coexist. The proportion of these two states of matter changes linearly with an increase in volume from value V_1 where the substance takes the form of liquid to value V_2 where it changes entirely into gas. The phenomenon of phase coexistence, along with the appearance of the latent heat, are regarded as hallmarks of the first–order phase transitions. In contrast to the second–order phase transitions, the correlation length at this transition is generally finite.

It is important to note that a slightly different approach to phase transitions and their classification was proposed in 1937 by Landau, who associated phase transitions with the symmetry of a system. He found that the disordered phase exhibits a symmetry that is (at least partially) lost during a transition to the low–temperature phase. Let us

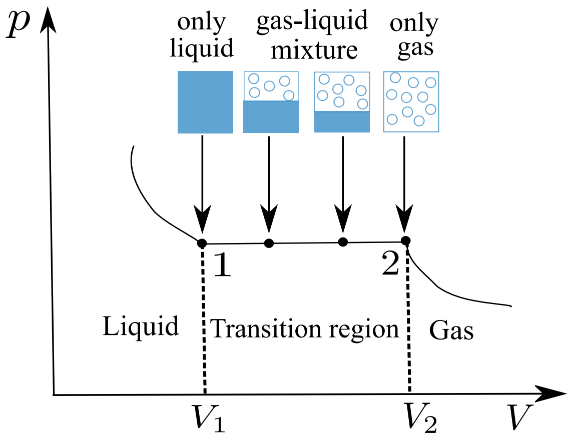


Fig. 2.4 Example of the phase diagram (pressure p vs. volume V dependence) of the liquid–gas first–order phase transition. Below V_1 and above V_2, only one phase exists, either liquid or gas. For values of volume V lying in the range $V_1 < V < V_2$, a substance takes the form of a mixture of coexisting phases. Its proportion changes linearly with an increase of V.

illustrate this with the example of the Ising model. Above the critical temperature, the spins point equally in both directions, and magnetization is zero. In this state, the system is invariant under transformation $s_i \to -s_i$, and we can say that it is characterized by the Z_2 symmetry. Below the critical point, all spins are aligned in one of two possible directions of magnetization, which leads to the spontaneous breaking of symmetry present in the high–temperature phase.

2.3 The order parameter

In the study of phase transitions, it is useful to introduce a variable that drastically changes its behavior at the critical point. We call it the *order parameter*. It takes a non–zero in the low–temperature (ordered) state and drops smoothly to zero at a transition point, where the order is completely lost. There is no general definition of the order parameter. Its choice is not unique and usually dictated by its utility [8]. Some propositions of the order parameter for various classical and quantum systems are listed in Tab. 2.1.

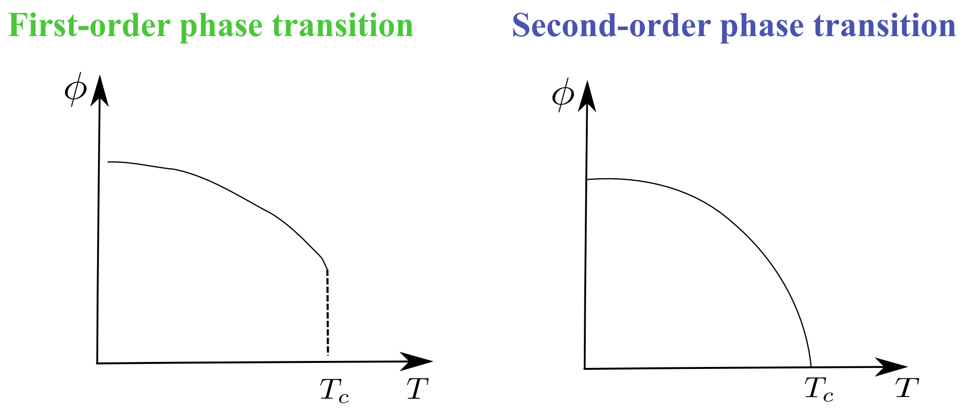
The order parameter ϕ can also be useful in characterizing the nature of phase transitions. We observe a noticeable difference in the behavior of ϕ during first– and second–order phase transitions, as illustrated in Fig. 2.5. While in the case of second–order phase transitions, the order parameter drops smoothly to zero at the critical point, in the case of first–order phase transitions, one observes its discontinuous jump.

It is also possible to create an alternative classification of phase transitions. One of them is associated with the notion of the order parameter and symmetry of the system [9]. It divides phase transitions into two types:

1. *transitions with no order parameter* – the symmetry groups of two phases are not

The physical system	The order parameter	The form	n
Ferromagnet	Magnetization $\vec{M}$	vector	3
Ferroelectric	Polarization P	scalar	1
Liquid	density difference $\rho - \rho_C$	scalar	1
Superfluid ^{4}He	Wave function of the ground state Ψ_0	complex scalar	2
Superconductor	Cooper pair wave function Ψ	complex scalar	2
The Ising model	magnetization m	scalar	1

Table 2.1 Examples of the order parameters for various physical systems. Symbol n denotes the dimensionality of the presented order parameters.



First-order phase transition

Second-order phase transition

Fig. 2.5 The illustration of different behavior of the order parameter ϕ during first (discontinuous) and second–order (continuous) phase transition in the function of temperature.

included in one another (they are always first–order according to the Ehrenfest classification),

2. *transitions with order parameter* – the symmetry group of the low–temperature phase is included in the symmetry group of the high–temperature phase. It can be continuous or discontinuous depending on the behavior of the order parameter at the transition point (see Fig. 2.5).

2.4 Critical exponents and universality classes

As emphasized in the previous Sections, phase transitions are accompanied by divergences in physical quantities. It is instructive to understand the character of emerging singularities. For this purpose, we usually introduce a set of constants called *critical exponents*, which describe the behavior of various thermodynamic quantities at the transition point.

For a thermodynamic function F the critical exponent λ is defined as:

$$\lambda = \lim_{t\to\infty} \frac{\ln|F(t)|}{\ln|t|}, \tag{2.10}$$

where the reduced temperature t:

$$t = \frac{T - T_C}{T_C} \tag{2.11}$$

is a measure of the deviation of a given temperature T from its critical value T_C. In practise, it is more common to use a slightly modified version of Eq. (2.10):

$$F(t) \sim |t|^{\lambda} \tag{2.12}$$

for $t \to 0$.

The form of the critical behavior of several thermodynamic quantities characterizing magnetic systems defined in terms of critical exponents is presented in Tab. 2.2. They are not independent of each other – and many inequalities illustrate relationships between them. They arise from the dependencies existing between thermodynamic variables and mathematical assumptions that explain their behavior. For instance, the easiest to prove is the Rushbrooke inequality [10]:

Observable	Critical exponent
Specific heat C	$C \sim |t|^{-\alpha}$
Magnetization M	$M \sim (-t)^{-\beta}$
Isothermal susceptibility χ	$\xi \sim |t|^{-\gamma}$
Critical isotherm at T_C	$H \sim |M|^{\delta}\mathrm{sgn}(M)$
Correlation length ξ	$\xi \sim |t|^{-\nu}$
Correlation function $\mathcal{C}(r)$ at T_C	$\mathcal{C}(r) \sim 1/(r^{d-2+\eta})$

Table 2.2 The critical behavior of the various physical quantities in dependence on the reduced temperature t. The table is adopted from [7].

$$\alpha + 2\beta + \gamma \geq 2. \tag{2.13}$$

It results from the relation between magnetic susceptibility χ and specific heats C_H, $C_{H,M}$ (Fig. 2.2) and the assumption that $\chi \geq 0$. The other one, known as Griffiths inequality [11]:

$$\alpha + \beta(1 + \delta) \geq 2 \tag{2.14}$$

is a consequence of the convexity of the free energy.

The critical exponents λ's are interesting with regard to their *universality*. While the critical temperature T_C strictly depends on the physical model, which may involve different interactions (as we shall see in the forthcoming Section), the λ's are common to many systems. Their values depend solely on the dimension d of the system and the symmetry of the order parameter [7]. Systems described by the identical values of critical exponents fall into the same universality class. It implies that the critical behavior of complicated models can be derived from the simpler ones as far as they are assigned to identical universality classes.

2.5 Phase transitions in various spin models

The preceding Sections focus on the general description of the formalism used in the physics of phase transitions without specifying any particular system. Here, we fill this gap by introducing microscopic models describing systems undergoing phase transitions. We start with the Ising model, the simplest model of ferromagnetism we study here. Other models of magnetism we discuss are more complex but can be reduced to the Ising model in some exceptional cases.

The Ising Model

The Ising model is a simplified mathematical model of ferromagnetism [12]. It describes a collection of *classical* magnetic moments ("spins") localized on the d–dimensional lattice, which can point in one of two possible directions: either up or down ($s_i = \pm 1$). The model is described by the following Hamiltonian:

$$H = -J \sum_{\langle i,j \rangle} s_i s_j - h \sum_{i=1}^{N} s_i. \tag{2.15}$$

The first term in Eq. (2.15) is responsible for the interaction between spins. It has the short–range character: the interaction takes place only between neighboring spins i, j (which is denoted by $\langle i, j \rangle$). The constant J is called the *exchange energy*. It affects the character of underlying interaction: its positive and negative value favors parallel or antiparallel alignment of spins, respectively. The second term in Eq. (2.15) is optional and describes the interaction of N spins with an external magnetic field h.

This remarkably successful spin interaction model is instrumental in studying phase transitions – it has been demonstrated that for $d > 1$, the model undergoes a second–order phase transition. In further considerations, let us assume that we are dealing with a 2–dimensional Ising system in the absence of an external magnetic field $h = 0$. Then the system, at a critical temperature T_C (called the Curie point), passes from a paramagnetic state, in which the directions of spins are random (equal number of spins point up and down), to a ferromagnetic one, which is characterized by parallel alignment of spins (all spins point either up or down). This phase transition is associated with a loss of symmetry. While the high–temperature state is invariant under transformation $s_i \to -s_i$, the low–temperature state is not. It implies that the ground state exhibits lower symmetry than the state representing the system at the temperature $T > T_C$. As can be inferred from the classification of phase transitions presented in Sec. 2.3, the symmetry of a system can be associated with the notion of the order parameter ϕ. The most obvious choice of ϕ, in this case, is magnetization M, which is defined as:

$$M = \langle s \rangle = \frac{1}{N} \sum_i s_i, \tag{2.16}$$

where N is the number of lattice sites. The analytical formula for the temperature dependence of magnetization $M(T)$ can be derived from the exact analytical solution to the Ising model found by Onsager in 1944 [13]. We plot it in Fig. 2.6. The form of this function is similar to the one presented in Fig. 2.5: the order parameter drops smoothly and reaches zero in $T = T_C = 2.269\ J/k_B$ (as predicted by the analytical solution), which confirms the continuous character of phase transition.

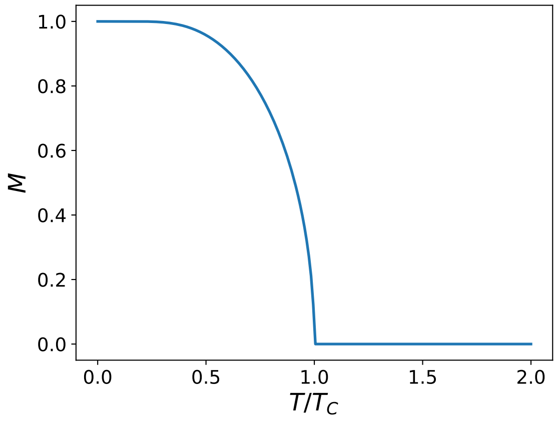


Fig. 2.6 Magnetization M in the function of temperature presented for two-dimensional Ising Model.

The studies of critical phenomena in the Ising model initiated by Onsager were later extended by other researchers [14, 15]. They revealed the exact values of critical exponents defined in Sec.2.4 which when $h = 0$ are the following: $\alpha = 0$, $\beta = 1/8$, $\gamma = 7/14$, $\delta = 15$, $\nu = 1$, $\eta = 1/4$. It is easy to verify that they indeed obey the Rushbrooke and Griffiths inequalities $\big($Eq. (2.13), Eq. (2.14)$\big)$ which, in this case, hold as equalities. The universality of the critical exponents makes them general for all models constructed on planar lattices with short–range interactions (only between nearest neighbors).

It is important to emphasize that although the Ising model was introduced to study ferromagnetism, it can be applied to the vast number of interacting two–state systems. For instance, it can successfully describe the phase transition occurring in brass (a binary alloy of copper and zinc) in which the lattice site is either occupied by an atom of copper or zinc. Similarly, the Ising Hamiltonian (Eq. (2.15)) can model the lattice gas system in which an atom can occupy a lattice site or not.

The Potts Model

The 2–dimensional Potts model is a generalization of the Ising model for more than two spin components [16]. The q–component version of this model describes an interaction of spins, each pointing in one of the q equally spaced directions θ_n:

$$\theta_n = \frac{2\pi n}{q}, \quad n = 0, 1, \ldots, q-1. \tag{2.17}$$

The Hamiltonian of this model is given by:

$$\mathcal{H} = -J \sum_{\langle i,j \rangle} \delta_{s_i, s_j}, \tag{2.18}$$

where $s_i = 0, \ldots, q-1$ represents the spin direction θ_i defined in Eq. (2.17) and $\langle i, j \rangle$ stands for the nearest neighbor sites. It is easy to notice that for $q = 2$, the Potts model describes the Ising system. It must thus undergo a continuous phase transition. The calculations based on the graph theory reveal the same character of phase transition also for the 3– state and 4– state Potts models. According to the same theory, for $q > 4$ the phase transition has discontinuous character [17].

The Blume–Capel Model

The Blume–Capel Model can be viewed as a generalization of the Ising model for spin $S = 1$ in which the external magnetic field is replaced by the anisotropy field D [18, 19]. The following Hamiltonian defines the model:

$$\mathcal{H} = -J \sum_{\langle i,j \rangle} s_i s_j + D \sum_i s_i^2, \tag{2.19}$$

where $\langle i, j \rangle$ indicates that summation is made over all pairs of nearest–neighbor spins. The Hamiltonian is composed of two terms: the first indicates the usual spin–spin interaction, while the second describes the single–spin interaction with the anisotropy field D. The model is characterized by the phase diagram exhibiting the tricritical point (TPC) separating the first– and second–order phase transitions [20, 21]. Its sketch is shown in Fig. 2.7. Monte Carlo simulations predict the values for the TPC in the thermodynamic limit at $k_B T_C/J = 0.609(4)$ and $D_C/J = 1.965(5)$ [22].

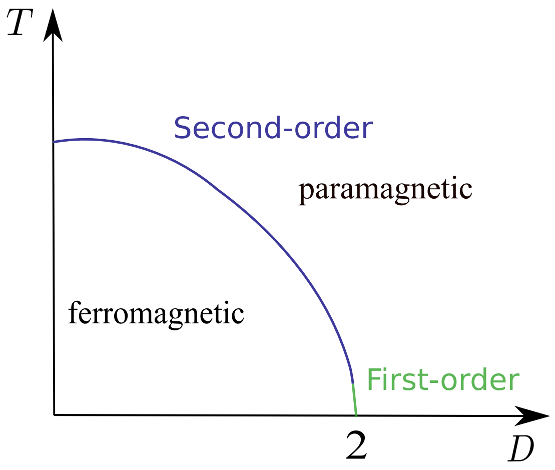


Fig. 2.7 A sketch of a phase diagram of the Blume–Capel model. The character of the phase transition changes at the critical point $(D_C, T_C) \approx (1.965, 0.609)$ for $J = 1$.

The classical XY Model

The classical XY model can be regarded as a generalization of the Ising model for spins pointing in any direction. Within the 2–dimensional version of this model, a spin placed on the lattice site i is described by the classical unit length vector: $s_i = (\cos\theta_i, \sin\theta_i)$. Similarly to the Ising model, it involves only nearest–neighbor interactions, and in the absence of an external magnetic field, it is described by the following Hamiltonian:

$$\mathcal{H}_{XY} = -J \sum_{\langle i,j \rangle} \cos(\theta_i - \theta_j), \tag{2.20}$$

where J stands for the coupling constant and $\langle i, j \rangle$ denotes that the interaction takes place only between spins placed on the neighboring sites i and j. As can be inferred from Eq. (2.20) energy of the system does not change under the influence of the global rotation of spins. Therefore, the Hamiltonian of the model exhibits continuous rotational symmetry $O(2)$. Mermin and Wagner have rigorously proved that continuous symmetries cannot be spontaneously broken at any finite temperature in 2–dimensional systems with short–range interactions [23]. This theorem excludes the possibility of a ferromagnetic phase and, as a consequence, an Ising–type phase transition. Yet, the XY–model does exhibit a peculiar phase transition not accompanied by the spontaneous symmetry breaking. Therefore, it can not be classified within the theories of phase transition presented in the previous Sections. Kosterlitz and Thouless found that it is signaled by a subtle change in the functional form of the correlation function between spins [24, 25, 26]. Above the critical temperature T_{BKT}, the correlation function obeys a normal, exponential behavior (see Eq. (2.5)):

$$\langle S_i S_j \rangle \sim e^{-r/\xi}, \tag{2.21}$$

where r is the distance between the spins placed on the lattice sites i and j and ξ denotes the correlation length. On the contrary, for temperatures $T < T_{BKT}$, it takes the unusual algebraic form:

$$\langle S_i S_j \rangle \sim r^{-\eta}, \tag{2.22}$$

where η is the exponent proportional to the temperature: $\eta = T/(2\pi J)$ [27]. This specific form of $\langle S_i S_j \rangle$ occurring for $T < T_{BKT}$ leads to the development of the so–called *quasi long–range order* (QLRO) in the system. Kosterlitz found [26] that the change in the correlation function can be associated with the appearance of topological defects in a system called *vortices* and *antivortices*. They are defined as points in configurations around which the spins complete the revolution on 2π or -2π, respectively. Examples of these topological excitations are depicted in Fig. 2.8. In the temperature range dominated by QLRO ($T < T_{BKT}$), the vortex–antivortex pairs are bound together. At the critical temperature T_{BKT}, the QLRO is destroyed, resulting in unbinding the vortex–antivortex pairs. Their number proliferates with an increase in temperature. Simultaneously, the average distance between already existing pairs grows in a logarithmic manner [24, 25, 26]. Such evolution of the vortex–antivortex pairs concentration as a function of temperature is depicted in Fig. 2.9.

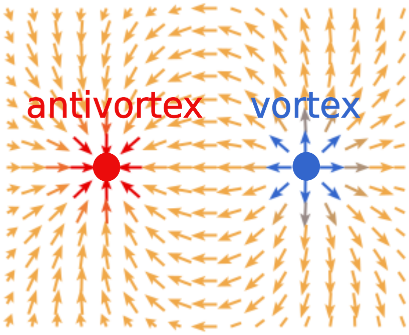
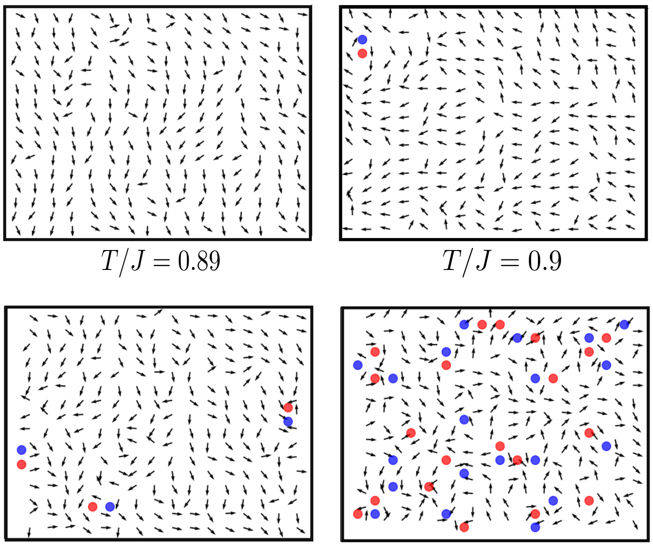


Fig. 2.8 An example of vortex and antivortex pair formation in a spin configuration. The vortex and antivortex correspond to the spins change by 2π and -2π for a path that encloses their cores (marked by blue and red dots), respectively.

As discussed above, the Berezinskii–Kosterlitz–Thouless (BKT) phase transition differs significantly from the 'classical' first– and second–order transitions. It is not accompanied by spontaneous symmetry breaking and possesses no order parameter. Therefore, it is sometimes called *infinite order phase transition*. The theory of BKT phase transition has numerous applications in studies related to liquid helium, liquid crystals, and superconductors.

The quantum XY Model

The quantum XY model is a mathematical model that extends the classical XY model to describe the behavior of quantum spins on a two-dimensional lattice. Its Hamiltonian



$T/J = 0.89$ $T/J = 0.9$

$T/J = 0.95$ $T/J = 1.3$

Fig. 2.9 An illustration of the vortex–antivortex pairs formation in the MC configurations of 2–dimensional classical XY model (lattice size $L = 16$) for different temperatures lying below and above the critical temperature ($T_{BKT} \approx 1.0/J$). Blue and red dots mark the positions of vortices and antivortices, respectively.

is given by:

$$\mathcal{H}_{q-XY} = \frac{E_C}{2} \sum_i \hat{n}_i^2 - J \sum_{\langle i,j \rangle} \cos(\hat{\theta}_i - \hat{\theta}_j), \tag{2.23}$$

where $\hat{n}_i$ is the number operator and E_C is the charging energy. It can be physically realized as a periodic array of ultra-small Josephson junctions [28, 29]. Then the operator $\hat{n}_i$ appearing in the first term of the Hamiltonian (Eq. 2.23) measures the number of Cooper pairs transferred between grains and E_C is regarded as the energy needed to induce a charge of one electron on the junction with capacitance C. The second term can be considered as the Josephson coupling energy between two consecutive grains described by the quantum mechanical wave function $\psi_i = |\psi_i| e^{-i\theta_i}$. The competition between these energy terms results in the BKT phase transition between the normal and superconducting state.

Phase–fermion model

The phase–fermion model (PF) [30] describes fermions coupled locally to fluctuating classical phases. It is given by the Hamiltonian:

$$\mathcal{H}_{PF} = -t \sum_{\langle i,j \rangle, \sigma} \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} + g \sum_i \left(e^{i\theta_i} \hat{c}_{i\uparrow}^\dagger \hat{c}_{i\downarrow} + h.c. \right), \tag{2.24}$$

where $c_{i\sigma}^\dagger$, $c_{i\sigma}$ stand for the creation and annihilation operators of an electron with spin σ placed at the lattice site i, t is the hopping integral, and g denotes the coupling constant between the electron and phase. The model can be regarded as an approximation of the boson–fermion model [31] described by the Hamiltonian:

$$\mathcal{H}_{BF} = -t \sum_{\langle i,j \rangle, \sigma} \hat{c}^{\dagger}_{i\sigma} \hat{c}_{j\sigma} + \hat{g} \left(\sum_{i} b^{\dagger}_{i} \hat{c}_{i\uparrow} \hat{c}_{j\downarrow} + H.c. \right) - \mu (2\hat{n}_b + \hat{n}_f) + E_B \hat{n}_B, \quad (2.25)$$

in which μ is the chemical potential, $\hat{n}_B(\hat{n}_F)$ is the density operator for bosons (fermions) and E_B corresponds to the bosonic level. In the limit of a large number of bosons per lattice site, fluctuations in their number are negligible. We can then use the approximation $\hat{b}_i = \sqrt{n^B_i} e^{i\theta_i}$ which makes bosons classical, localized particles described by the phase θ_i. Furthermore, assuming that $E_B = 2\mu$ (resulting in the lack of the effective bosonic chemical potential) and by replacing the constant $\hat{g}$ with $g = \hat{g}\sqrt{n_B}$ we obtain the Hamiltonian introduced in Eq. (2.24).

In general, the model describes a system of fermions hopping between different lattice sites and scattering off of the classical phases. It leads to the development of the indirect, mediated by mobile fermions long–range–temperature–dependent interaction between bosonic phases. This phenomenon is, in turn, responsible for the occurrence of temperature–driven BCS–BEC and BKT phase transitions.

The model can describe a vast number of realistic physical systems. One of many examples includes an array of superconducting islands placed on top of a graphene sheet [32]. In such a case, the islands are modeled by fluctuating classical phases, while Cooper pairs mediate the interaction between the islands by tunneling from one to another. Surprisingly, the interaction between the islands survives even if the distance between the islands is larger than the range of the superconducting proximity effect.

The Falicov–Kimball Model

The Falicov–Kimball Model [33] (FK) was introduced in 1969 to explain the metal–insulator phase transition occurring in rare–earth materials and transition–metal compounds [34]. It can be regarded as a limiting case of the generalized Hubbard model [35] described by the Hamiltonian:

$$\mathcal{H} = -\sum_{ij\sigma} t_{ij\sigma} c^{\dagger}_{i\sigma} c_{j\sigma} + U \sum_{i} n_{i\uparrow} n_{i\downarrow}, \qquad (2.26)$$

where $c_{i\sigma}$, $c^{\dagger}_{i\sigma}$ stand for the annihilation and creation operators of an electron with spin σ at the lattice site i, $t_{ij\sigma}$ is the hopping–integral, U denotes the strength of an on–site Coulomb interaction and n_i is the occupation number operator for spin σ at a site i.

The simplification in the FK model assumes that one type of electrons, for instance, 'spin–down', are heavy localized particles, while 'spin–down' electrons are light, itinerant ones. This assumption simplifies the Hubbard Hamiltonian to the form called the spinless FK model:

$$\mathcal{H}_{FKM} = -t \sum_{<i,\, j>} c_i^\dagger c_j + U \sum_i n_i w_i + H.c., \tag{2.27}$$

where t is spin–independent hopping integral, symbol $\langle i, j\rangle$ denotes that the hopping takes place only between nearest neighbor sites and the parameter $w_i \in \{0, 1\}$ indicates whether the site i is occupied by a heavy electron ($w_i = 1$) or not ($w_i = 0$).

Perhaps the most straightforward interpretation of the model is when one refers to the localized particles as heavy, classical ions that interact electrostatically with the itinerant electrons. It is the only direct interaction that occurs in the lattice system – the ions interact among themselves only effectively. This effective interaction of ions makes the system's energy state dependent on their distribution.

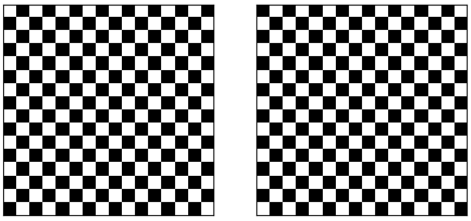


Fig. 2.10 Two ground states of the Falicov–Kimball Model. Black and white squares denote the lattice sites occupied by ions and free of them, respectively.

In particular, it was shown by Kennedy and Lieb [36, 37] that for every bipartite lattice, the half–filled FK model (when the lattice is equally filled with light and heavy particles) exhibits long–range order at low temperatures. The ordered state is the famous checkerboard state (Fig. 2.10), which also turns out to be the system's ground state.

Although the FK model is much simpler than the Hubbard model (Eq. (2.26)), rigorous analytic solutions of the former are known only for one ($d = 1$) [38, 39], and infinite dimension ($d = \infty$) versions of the model [38, 40]. Hence, we need numerical methods to obtain some information about the model in other cases. The results obtained from Monte Carlo simulations suggest that the half–filled FK model undergoes the first–order phase transition for $U \leq 1$ and the second–order phase transition for $U > 1$ [41], as sketched in the phase diagram shown in Fig. 2.11.

It is important to emphasize that the critical interaction $U^* \approx 1.0$ separating between two kinds of phase transition is affected by a significant error. It comes from the fact that it is estimated with the Gaussian method [42] only for one system size (20×20) [41]. To obtain more accurate predictions, we should include the results for much larger systems, which are hard to simulate (the calculations are computationally expensive).

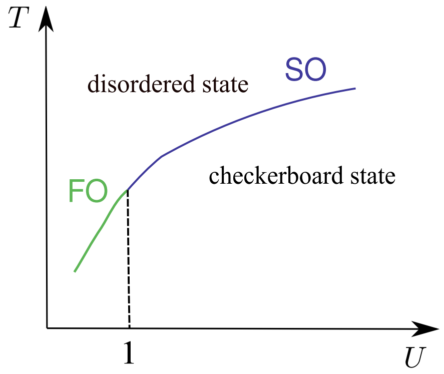


Fig. 2.11 A sketch of the phase diagram of the half–filled FK model. The nature of a phase transition changes approximately at the point $(U_C, T_C) = (1.0, 0.595)$ $(t = 1)$.

2.6 Summary

In this Chapter, we introduced basic concepts of thermodynamics and statistical physics associated with the very rich and complicated theory of phase transition and critical phenomena. We also introduced a few microscopic models' Hamiltonians of systems undergoing different types of phase transitions. They will be more carefully analyzed in the forthcoming Sections of this thesis with new tools based on machine learning algorithms.

CHAPTER 3

Basic concepts of machine learning and its application to problems of condensed matter physics

Nowadays, machine learning methods are widely used in many science areas, with physics no exception. Due to their enormous capacity to deal with high–dimensional data, they are also ideally suited for solving problems of condensed matter physics. Thus, ML methods have the potential to become a new tool in the physicists' box alongside standard techniques such as calculus or Fourier transform [2].

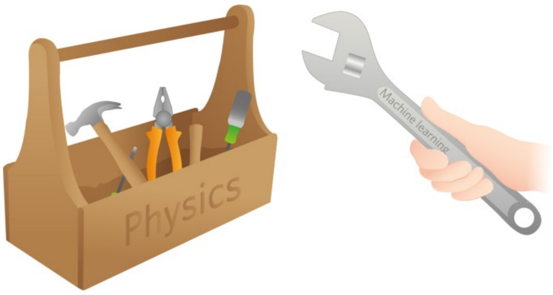


Fig. 3.1 Figure taken from [2] presenting the machine learning methods as a new tools in the box of physicist.

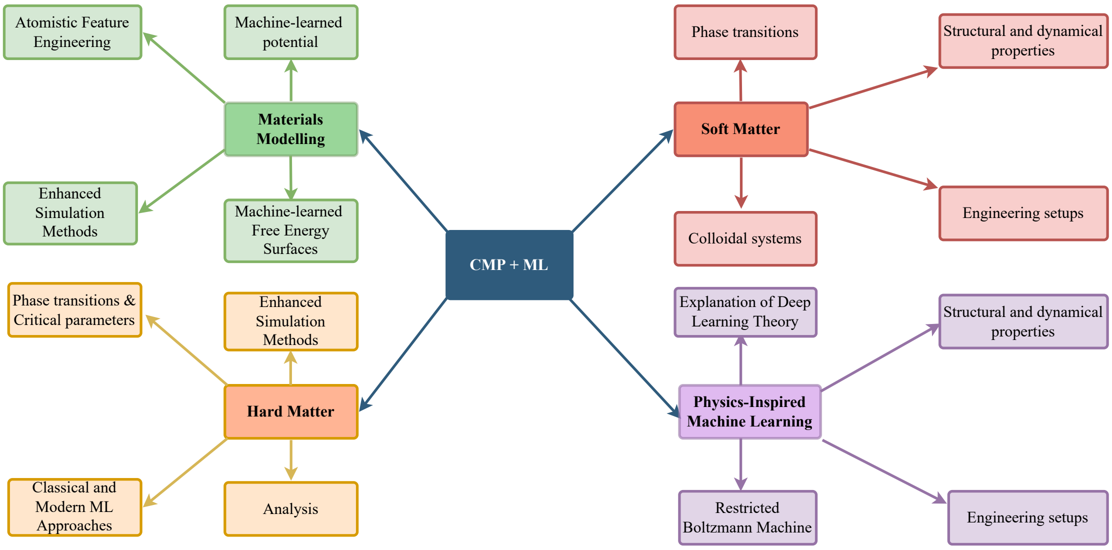
According to Ref. [3], there exist four major areas of condensed matter physics that overlap with ML methods:

- **Hard Condensed Matter** – research area involving studies of critical phenomena, phase transitions (including topological ones), and improvement of classical simulation methods,

- **Soft Condensed Matter** – research area which studies materials prone to the application of external forces. The ML algorithms are applied here to detect phase transitions or topological defects given their configurations,

- **Materials Modeling** – research area with focus on the enhancement (with the ML methods) of the *Molecular Dynamics (MD)* simulations, *Density Functional Theory (DFT)* computations, atomistic feature engineering, reconstruction of the potential energy surface (PES) of the materials,

- **Physics–inspired Machine Learning Theory** – research area where theoretical physics frameworks give new insight into explaining the mechanism behind the DL models.



CMP + ML
Materials Modelling
Atomistic Feature Engineering
Machine-learned potential
Enhanced Simulation Methods
Machine-learned Free Energy Surfaces
Hard Matter
Phase transitions & Critical parameters
Enhanced Simulation Methods
Classical and Modern ML Approaches
Analysis
Soft Matter
Phase transitions
Structural and dynamical properties
Colloidal systems
Engineering setups
Physics-Inspired Machine Learning
Explanation of Deep Learning Theory
Structural and dynamical properties
Restricted Boltzmann Machine
Engineering setups

Fig. 3.2 The schematic overview of the possible application of the ML to the problems of the condensed matter physics [3]. CMP is the abbreviation for Condensed Matter Physics.

Regarding the character of this thesis, we will be more specific only on the **Hard Condensed Matter** Applications.

3.1 Basic ML classification methods

The machine learning methods can be divided into three main categories [43]:

- supervised learning,
- unsupervised learning,
- feature extraction.

The goal of supervised learning is to find a function f that maps an input data $\mathbf{x}$, usually exhibiting some pattern, to the label $\mathbf{y}$, i.e. $\mathbf{y} = f(\mathbf{x})$. The main task of an ML algorithm is to fit $\mathbf{x}$ to $\mathbf{y}$ by minimizing the error between actual labels $\mathbf{y}$ and predicted ones $\hat{\mathbf{y}}$. The loss (error) functions can take various forms (cross-entropy, mean squared error, Kullback–Leibler loss function) and may differ among applied methods.

In contrast to supervised learning, labels are not provided in the unsupervised learning scheme. In this case, the algorithm looks for patterns in the input data and tries to group them into clusters.

The feature extraction methods mainly aim at reducing the dimensionality and selecting the most representative features of the data. The algorithms in this category are instrumental in cases where our data are high–dimensional and highly correlated. It is usually used in the preprocessing stage, just before feeding the data into a "proper" supervised or unsupervised task.

As for the applications of these methods to condensed matter physics, most studies focus on investigating the Ising model [44] since it can be treated as a paradigm to solve more complicated problems. The reason for this choice is obvious: it is the simplest model of ferromagnetism that predicts a phase transition in dimension $d > 1$.

3.2 Linear Regression

Linear regression is a simple technique borrowed by machine learning from statistics, with the aim of fitting input data $\mathbf{x}$ to a known output $\mathbf{y}$ with a linear function. Therefore, it belongs to the class of supervised learning algorithms. The main task of the algorithm is to minimize the mean squared error function L_{MSE}:

$$L_{MSE} = \frac{1}{N} \sum_{i=1}^{N} \left(\hat{y}_i - y_i\right)^2 , \tag{3.1}$$

where N is the number of samples, y_i denotes the label of the sample i and

$$\hat{y}_i = \sum_{i=1}^{N} w_i \mathbf{x_i} + w_o \tag{3.2}$$

is the prediction provided by the algorithm. The weights w_i and the bias w_o are fitted during the training process in such a way as to minimize L_{MSE}.

In the field of condensed matter physics, one can use this extremely straightforward method to speed up Monte Carlo (MC) computations. MC methods are considered one of the main computational approaches [45, 46], used with systems characterized by many degrees of freedom. They can be applied to extract samples of configurations (states) in both classical and quantum many–body systems. Their realization is based on the Metropolis algorithm, which proposes a new state according to some probability distribution. This proposition can be realized either locally or globally. The local update method involves a change only in one lattice site. In contrast, the global update modifies a large number of spins at once. If a new configuration is accepted, it is treated as a new state in the Markov chain. Otherwise, we start from the previous state again and repeat the whole procedure with the new update. The main drawback of the local update method is the long autocorrelation time τ occurring in the vicinity of the phase transition (the phenomenon dubbed the *critical slowing down*). The global update methods such as Wollf [47], Swenden-Wang [48] or "worm" [49] significantly reduce this problem. They are, however, designed only for specific models – it is extremely difficult to create a generic global update method. In order to overcome these issues, some techniques

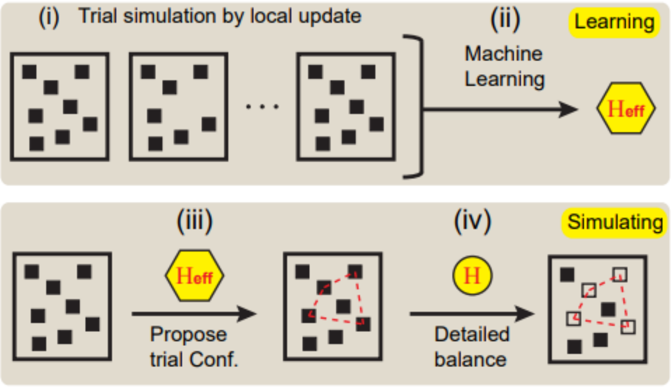


Fig. 3.3 Four steps that make up the SLMC scheme. Step (i) involves the simulation of a system with the local update method. In step (ii), generated MC configurations are fed into a machine learning algorithm that learns an effective Hamiltonian $\mathcal{H}_{eff}$ with the known global update principle. In step (iii), new configurations in the Markov chain are proposed according to $\mathcal{H}_{eff}$. The rejection or acceptance of a new configuration is then verified by the detailed balance principle of the original Hamiltonian $\mathcal{H}$ (step (iv)). The figure is taken from [4].

based on ML have been developed. One of them is called the *Self–learning Monte Carlo* (SLMC) scheme [4, 50]. As presented in Fig. 3.3 it is composed of four steps. The first step involves the generation of MC configurations with the usual local update method. Subsequently, we extract from the MC samples the n-th nearest-neighbor spin-spin correlations $C_n^m = \sum_{\langle i,j \rangle_n} S_i S_j$. The set of C_n^m and energies of the corresponding configurations E^m make up the training data. They are, in the next step, fed into a linear regression algorithm with the aim of fitting C_n^m to the energies E^m according to the equation:

$$E^m = E_0 + \sum_{n=1}^{N} \tilde{J}_n C_n^m, \tag{3.3}$$

where E_0 and $\tilde{J}_n$ are optimized during the training process. Since, as it turns out from the numerical computations, the dominant contribution to the energy E^m comes from the first term involving nearest–neighbor interactions, all coefficients $\tilde{J}_n$ $(n \geq 2)$ can be set to zero. In this way, a more complicated model can be transferred to a much simpler Ising–type model to which we can apply the global–update method. The rejection or acceptance of new propositions is, however, still verified by the detailed balance principle (DBP) of the original Hamiltonian, ensuring that the new configurations are extracted from the proper probability distribution.

3.3 Principal Component Analysis

The Principal Component Analysis (PCA) is an unsupervised learning algorithm that serves as a primary tool for dimensionality reduction [51]. It consists of projecting the N–dimensional input data X into a M–dimensional subspace $(M < N)$ to maximize the variance of the input samples' projections.

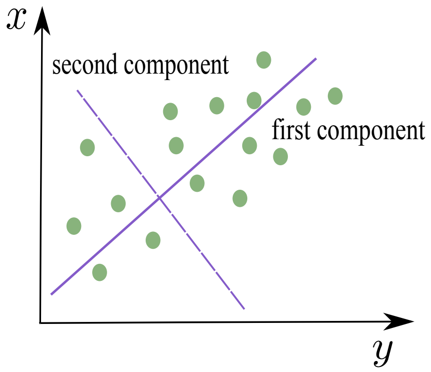


Fig. 3.4 The first and second components obtained after PCA transformation. The components are orthogonal to each other.

The PCA algorithm finds new coordinates (components) Y through the orthogonal transformation O:

$$Y = XO, \tag{3.4}$$

where O is a matrix composed of the eigenvectors $O = [o_1, o_2, o_2, \ldots, o_M]$ that are a solution to the following eigenproblem:

$$X^T X o_l = \lambda_l o_l \tag{3.5}$$

and correspond to the M largest eigenvalues λ_l. This operation ensures that the most meaningful features (responsible for most of the variance) are extracted. In particular, it allows visualization of the distribution of high–dimensional data by projecting it to a 2 or 3–dimensional space.

The usefulness of PCA in condensed matter physics problems has already been demonstrated in several systems. In particular, PCA is able to reveal spatial patterns of order and symmetry breaking. The mean value of the first principal component $\langle p_1 \rangle$ coincides with an order parameter for simple magnetic systems. On the other hand, the second leading component that captures low–energy fluctuations in the system can indicate the order of a phase transition [52, 53].

3.4 k–means

The k–Means algorithm is considered the simplest clustering method. It aims to identify k groups of data points in an unsupervised learning manner [54]. The goal of this algorithm is to assign data points in such a way as to minimize the sum of the squares of the distances of all data points to its closest center μ_K [55], that is, to minimize the objective function J of the form:

$$J = \sum_{i=1}^{N} \sum_{k=1}^{K} r_{nk} ||x_i - \mu_i||^2, \tag{3.6}$$

where $\{x_1, x_2, \ldots, x_N\}$ is the set of unlabeled data points in $D-$dimensional space and

$$r_{nk} = \begin{cases} 0 & \text{when} \quad x_n \notin k^{th} \text{ cluster,} \\ 1 & \text{when} \quad x_n \in k^{th} \text{ cluster.} \end{cases} \tag{3.7}$$

This can be achieved by applying the following procedure. In the beginning, one arbitrarily chooses the number of clusters K and its centers μ_K (based, for example, on the data's visual representation or the problem's specification). The data points are then assigned to their closest centers according to some chosen metrics. Subsequently, the new centers μ_K are calculated as the mean points of the clusters K to which the data points are properly reassigned. The procedure is repeated until there are no further changes in the clusters' assignments.

In the context of physics, the algorithm turns out to be useful in identifying the Monte Carlo configurations corresponding to different phases after initial reduction to 2–dimensional space by the PCA algorithm (Sec. 3.3) [56].

3.5 Support Vector Machine

Another popular ML algorithm is the Support Vector Machine (SVM) [57]. Since it requires data labeling, it is considered a supervised ML technique. Its main objective is to find a hyperplane given by equation $w\mathrm{x} + b = 0$ that separates two classes of data points with a maximum margin, i.e., with the maximal distance of the hyperplane to the closest data points (called support vectors) representing two categories. The working principle of this technique can be regarded as the optimization problem aiming to minimize the following loss function L:

$$L = \sum_{i=1}^{m} \max\{0, 1 - y_i \cdot (wx_i + b)\}, \tag{3.8}$$

where m is the number of data points, w and b are free parameters fitted during the training. The loss function presented in Eq. (3.8) called the *hinge loss* is responsible for maximizing the classification accuracy. It is always supplemented with a regularization term which has the task of finding a hyperplane with the maximum margin. The method is illustrated in Fig. 3.5.

Unfortunately, there exist a lot of datasets that are not linearly separable. In such cases, the SVM cannot be applied. Fortunately, we can adjust the method also to non–linear data sets. The way out of the problem is called a *kernel trick* and consists in

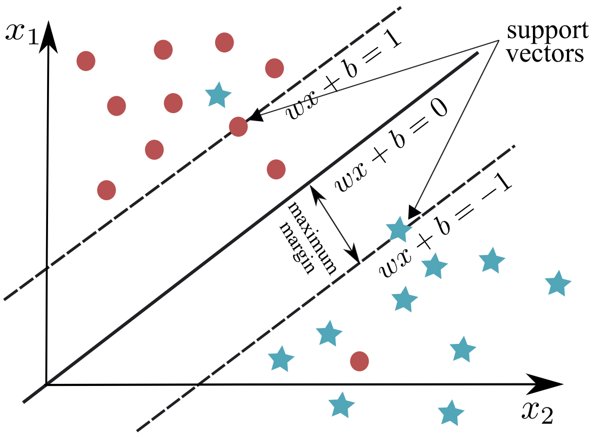


Fig. 3.5 The scheme of the linear Support Vector Machine (SVM) algorithm. The solid line represents the optimal hyperplane separating two classes of data. The dashed lines show two hyperplanes passing through the support vectors defined as two data points (one from each category) for which the distance to the hyperplane is the smallest.

mapping the original input data to a higher dimension using a *kernel function* where a linear decision boundary between classes can be determined. The idea is presented in Fig. 3.6.

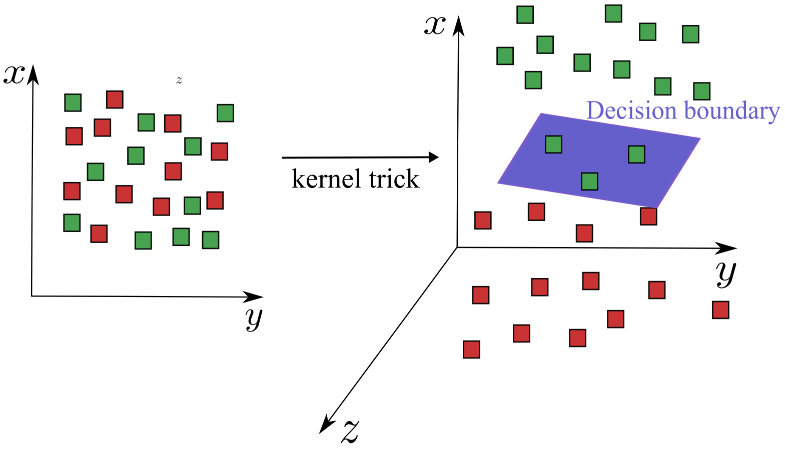


Fig. 3.6 A sketch of the kernel trick method. The set of data points non–separable in lower dimension (left panel) is transformed to a higher dimension, where the separation is possible (right panel).

When it comes to applications in the area of Hard Matter Physics, the algorithm is able to successfully separate Monte Carlo configurations representing two different phases [58, 59]. Moreover, it succeeds in evaluating the critical temperature by revealing the form of the order parameter. The main drawback of this technique is that the model is highly prone to the value of the regularization term, which needs to be appropriately tuned.

3.6 Neural Network

The simple machine learning methods introduced in the previous sections are, in many cases, capable of capturing some interesting physics from MC configurations. Moreover, they are less prone to overfitting, involve a relatively small number of parameters, and do not need so many data samples to return the correct predictions. On the other hand, they do not always deal with more complicated problems as they have poor generalization capability and, in some cases, require extensive parameter tuning. These issues do not apply to Artificial Neural Networks (ANNs), which can be considered to be the most representative models of ML. The neural network is designed in such a way as to mimic the processing of information by the human brain. The basic building block of the neural network is called a *neuron* (Fig. 3.7). It represents a non–linear function f of the weighted sum of the inputs $\mathbf{x}$ [60]:

$$\hat{\mathbf{y}} = f\left(\mathbf{W}\mathbf{x}\right), \tag{3.9}$$

where $\mathbf{W} = (w_0, w_1,\ w_2,\ ,\ldots,\ w_n)$ and $\mathbf{x} = (1, x_1,\ x_2,\ ,\ldots,\ x_n)$ represent the vectors of weights and inputs, respectively. Some common choices for the function f are:

- $f(x) = \max(0, x)$ (ReLU),
- $f(x) = 1/\left(1 + e^{-x}\right)$ (sigmoid),
- $f(x) = \tanh(x) = \left(e^{x} - e^{-x}\right)/\left(e^{x} + e^{-x}\right)$.

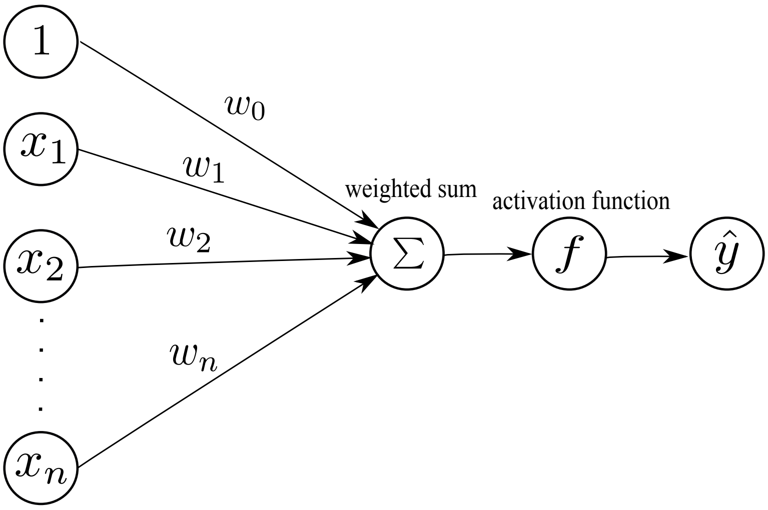


Fig. 3.7 The scheme of a neuron. Here $\mathbf{x} = (1, x_1,\ x_2,\ ,\ldots,\ x_n)$ stand for the input units to the neuron, $\mathbf{W} = (w_0, w_1,\ w_2,\ ,\ldots,\ w_n)$ are the weights and $\sum$ symbolizes weighted sum of inputs $\sum_{i=1}^{n} w_i x_i$. The output $y = f(w_i x_i)$, where f is an activation function.

The neural network comprises many such constructed neurons arranged in the layers as depicted in Fig. 3.8. Such a structure is called a *fully–connected neural*

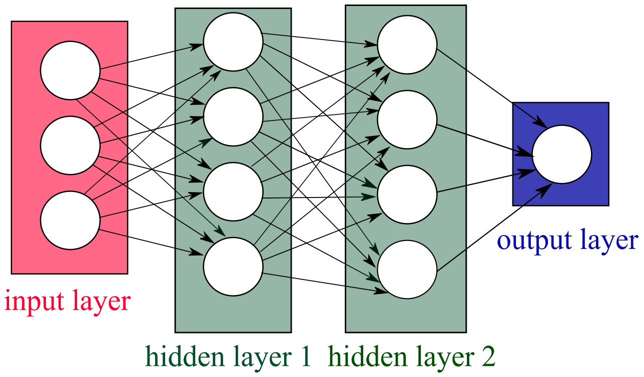


Fig. 3.8 The scheme of a feed–forward neural network (FNN). The white circles indicate the neurons arranged in 4 layers. The black arrows mark connections between neurons in consecutive layers.

network (FNN). The first layer is composed simply of the input data, while the last layer returns the output (prediction of a neural network). The intermediate layers between the input and output are called *hidden*. When a neural network is composed of many layers, we refer to it as a *deep neural network*.

Training of a neural network can be regarded as the process of fitting a known input $\mathbf{x}$ to the output $\mathbf{y}$ by some multiparameter function. Therefore, the neural network belongs to the class of supervised learning algorithms. The training process involves minimization of a loss function L describing deviations of the predicted output $\hat{\mathbf{y}}$ from the actual labels $\mathbf{y}$. The typical forms of the loss functions are the following:

- $L_{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$,
- $L_{BCE} = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i))$,
- $L_{CCE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{k} y_{i,j}\log(\hat{y}_{i,j})$.

Here N denotes the number of samples in a dataset, while k stands for the number of classes. L_{MSE} is usually used when labels take the form of decimal numbers. In contrast, the binary cross-entropy loss function L_{BCE} is applied to binary classification problems, where the samples are labeled either by 0 or 1. The categorical cross–entropy loss function is used in multiclass classification problems where the outputs $\hat{\mathbf{y}}$ of the neural network are one–hot encoded, i.e., written in the form of binary vectors. In such a case, if we are dealing, for example, with three different classes $\mathbf{1}, \mathbf{2}, \mathbf{3}$ and some sample belongs to $\mathbf{2}$, then its label is written in the form $y = [0, 1, 0]$. Weights of a neural network are typically found through minimization of the loss function with the gradient–descent algorithm such as the ADAM (*A method for stochastic optimization*) algorithm [61], realized in the framework of the backpropagation technique [62].

Another type of neural network is the convolutional neural network (CNN). It is characterized by its superior image and signal processing performance compared to FNN. Here, in contrast to FNN, the flow of information through the layers is a little

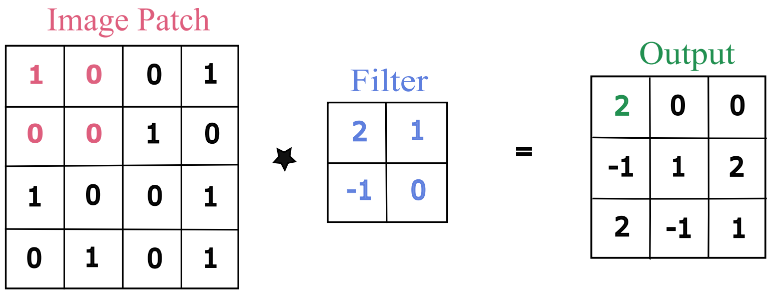


Fig. 3.9 The operation of convolution. The left part of the picture represents the input image with an extracted image patch of the size 2×2. In the middle part of the picture, we present the filter (kernel) with its corresponding weights. The green number in the output image denotes the result of the convolution operation marked with a star.

bit more complicated since the usual matrix multiplication is replaced by convolution (Fig. 3.9). This adjustment substantially reduces the number of parameters to be trained. Moreover, such construction of the CNN makes the algorithm focus on exploring the input data's local features, making it perfectly suitable for image classification problems.

Note that the number of training parameters does not depend on the input size, but rather on the arbitrarily chosen size of the filter. In the typical convolutional neural network presented in Fig. 3.10, one or two convolutional layers (such for which the convolution operation is applied) are placed at the beginning of the neural network. They are frequently supplemented with pooling operations that reduce the dimension of the output layer (by taking the mean or average of the emerging neurons). The CNN layers are then followed by the feed–forward neural network returning the prediction of the algorithm $\hat{y}$. The training of the CNN works on the same principle as in the case of FNN, described above.

The applications of neural networks to the problems of condensed matter physics are vast. One of the most prominent examples of the application of ML in HM physics is the approach invented by Carrasquilla and Melko [63]. In their study, they feed an FNN with both low– (for $T_0 \ll T_C$) and high–temperature (for $T_1 \gg T_C$) Ising configurations and label them with zeros and ones, accordingly. In the next step, they test such a trained NN on configurations corresponding to temperatures ranging from T_0 to T_1. The neural network returns in each case the probability $P(HTP)$ that a given configuration belongs to the high-temperature phase (**HTP**). The analogous probability of a MC sample to belong to the low–temperature phase (**LTP**) is obtained simply by subtraction $P(LTP) = 1 - P(HTP)$.

To illustrate the above–presented method, we apply this technique to 20×20 Ising configurations. The resulting probabilities $P(HTP)$ and $P(LTP)$ are depicted in Fig. 3.11. The critical temperature estimated by the algorithm is associated with the temperature for which a neural network is the most uncertain in its decision $(P(HTP) = P(LTP) = 0.5)$. It has been proved that in this simple case, the NN

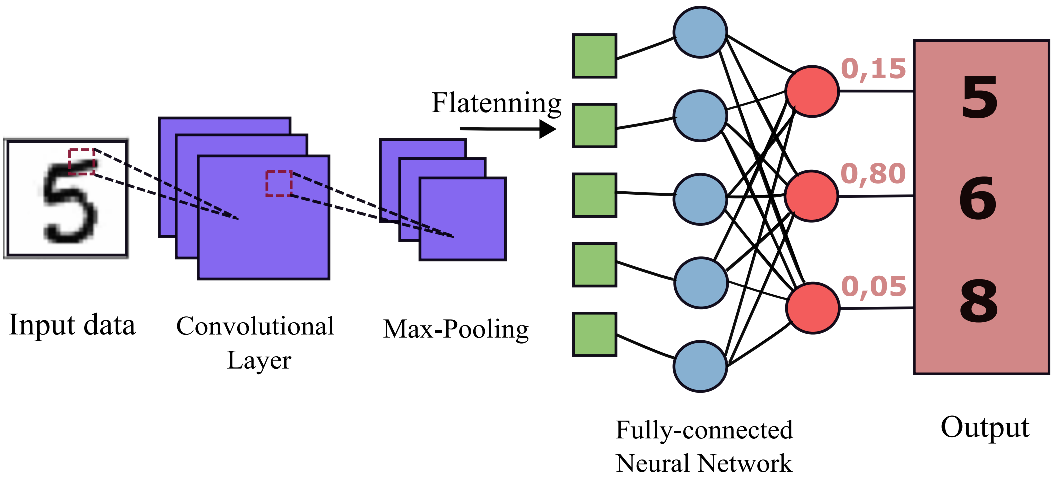


Fig. 3.10 An example of the CNN architecture. An input image is subjected to the convolution operation with a bunch of filters, creating a convolutional layer (CL). As the output of the convolution process, we obtain a set of feature maps. They are subsequently pooled, i.e., the group of neighboring neurons is replaced by its mean or maximum value. In the next step, neurons are converted to a single long continuous vector which comes as an input to the fully–connected neural network (FNN) returning probabilities of predictions.

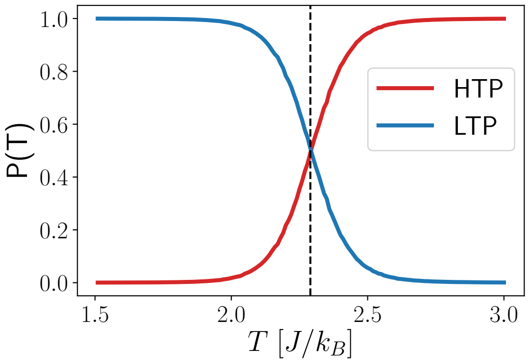


Fig. 3.11 The probability $\mathbf{P}(\mathbf{T})$ of the Ising configuration to represent the low–temperature phase (**LTP**) (marked with blue) or the high–temperature phase (**HTP**) (marked with red). The crossing point of two probability functions indicates the estimated value of critical temperature. The vertical dotted line shows the reference value of the critical temperature $T_C \approx 2.29\ J/k_B$ determined with MC simulations.

learns magnetization – it is therefore trivially to accurately identify T_C. Nevertheless, recent studies suggest that the algorithm also deals quite well with systems where the conventional order parameter does not exist [64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74].

Neural networks also succeed in the identification of the quantum phase transitions. It has been shown that they can distinguish different topological phases [75] and solve frustrated quantum many-particle problems [76]. Finally, it is worth mentioning of their enormous power in accelerating MC simulations [50, 77, 78]. It turns out that it is possible to incorporate them into the *Self–Learning Monte Carlo* scheme described in more detail in Sec. 3.2.

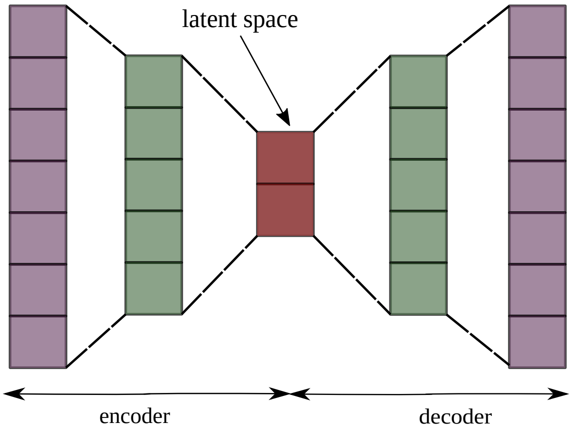


Fig. 3.12 The scheme of an autoencoder neural network. Purple blocks indicate the input and output layers of a neural network. Green ones correspond to an intermediate layer returning the reduced (latent) representation of the input data. Two red squares in the middle denote neurons providing two–dimensional latent representation of a data point.

3.7 Autoencoder neural network

An autoencoder is a neural network that aims to map an input to itself [79]. Because it does not require any data labeling, it is considered an unsupervised technique [80]. During the reconstruction process, it reduces the input data's dimension, eliminating noise and redundancy. Thus, it can be used, similarly to PCA, to reduce the dimensionality of the data. However, regarding its non–linearity, the autoencoder can grasp more non–trivial features of data than PCA, which is considered a linear method [81].

The autoencoder neural network consists of two parts: encoder and decoder, as illustrated in Fig. 3.12. In general, the encoder $\mathcal{E}$ transforms the input data $\mathbf{x}$ to some latent space $\mathcal{L}$, which usually has a lower dimension than the input. Then the decoder $\mathcal{D}$ attempts to reconstruct the input. In mathematical language, it can be written as:

$$\mathcal{E}(\mathbf{x}) = \mathcal{L}, \quad \mathcal{D}(\mathcal{L}) = \mathbf{x}'. \tag{3.10}$$

In its simplest form, the autoencoder is made of several layers creating feed–forward neural network with the following objective function $\mathbf{J}$:

$$\mathbf{J} = \frac{1}{N}\sum_{i=1}^{N} ||x_i' - x_i||^2, \tag{3.11}$$

which guarantees that the latent space $\mathcal{L}$ is constructed in such a way as to obtain the greatest resemblance between $\mathbf{x}'$ and $\mathbf{x}$.

In the field of the condensed matter physics, the autoencoder is used to learn the latent representation of the MC configurations, which is associated with an order parameter of the model [53]. It is important to emphasize that there is little gain in using the autoencoder instead of a much simpler PCA algorithm in the case of the Ising–type models. However, due to its non–linearity, it is extremely useful for analyzing more complicated models of condensed matter physics. One example is the XY model, which is characterized by continuous degrees of freedom. It implies that the number of ordered phase realizations is infinite. It makes their detection with linear methods impossible [53].

It is instructive to mention another popular algorithm that, similarly to PCA and autoencoder, maps high–dimensional input data to the lower dimension while preserving its structure. It is called *t–SNE* (*t-distributed stochastic neighbor embedding*) [82]. This algorithm is able to grasp non–linear features of MC configurations that allow detecting a percolation and Berezinskii–Kosterlitz–Thouless (BKT) phase transitions [83].

3.8 Restricted Boltzmann Machine

Another method belonging to the class of unsupervised learning techniques is called *Restricted Boltzmann Machine* (RBM) [81]. It can be regarded as the two–layer fully–connected neural network for which the training is held in two directions, as depicted in Fig. 3.13). The input dataset v_i received by a visible layer is sent to a hidden layer in a similar way as it happens in feed–forward neural networks. Then the hidden neurons h_i are sent back to the visible layer by the same transformation as before. The resulting output v_i' is in the next step compared to the initial input dataset v_i. The parameters of RBM are updated in such a way as to make the difference between v_i and v_i' the smallest possible. This procedure is repeated until convergence, i.e., until the initial input data are reconstructed.

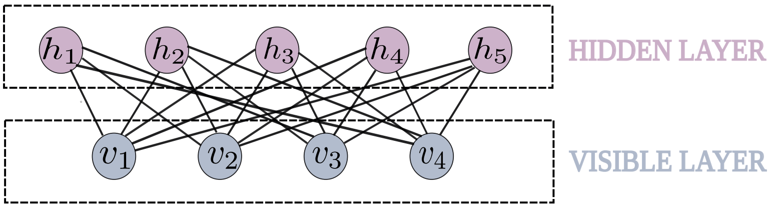


Fig. 3.13 The scheme of the *Restricted Boltzmann Machine* (RBM). The first input layer consisting of visible neurons v_i is fully–connected with the hidden layer composed of the neurons h_j.

RBMs can be used to accelerate MC simulations. It can be achieved by teaching the algorithm the functional form of energy based on the MC configurations derived from the local update method [84]. These energies are then used to propose new configurations according to the Boltzmann distribution $\sim e^{-\beta E}/Z$. The procedure significantly increases the acceptance ratio and autocorrelation time of the simulations, especially when calculating the energy in each MC step requires computationally expensive diagonalization of the Hamiltonian. Other applications of RBM include reconstructing complex quantum wave functions from experimental data [85, 86, 87], learning quantum states from spin configurations, finding the ground state of a system, or describing the unit time evolution of complex interacting quantum systems [88].

3.9 Machine learning methods in time series classification

In the following sections, we introduce two algorithms used in the time series classification problems of biological signals presented further in Section 4.4. Before proceeding with a detailed description, we present basic definitions associated with time series analysis.

3.9.1 Basics of time series analysis

A time series $T = t_1, t_2, \ldots, t_n$ is defined as a set of n real–value observations listed in time order. Every time series can be divided into subseries, i.e., subsequences of length $k < n$ created from sampling of k contiguous positions of the original time series. It is common to use the method of *sliding window*, which divides the entire time series into a set of sequences that can partially overlap. The length of this overlapping region is called the *stride*. The method is illustrated in Fig. 3.14. Unfortunately, the time samples obtained with this procedure may be too long for a machine learning algorithm. To reduce their lengths, one can apply the *Piecewise Aggregate Approximation* (PAA technique) [89, 90]. It downsamples a time series $\mathcal{T}$ of length l to another one of length $w < l$ by diving $\mathcal{T}$ into w equally–sized bins and calculate the mean values of data points within each of them. The sequence constructed out of these mean values $\mathcal{T}'$ is considered a new, reduced representation of the time series $\mathcal{T}$. Such an operation substantially reduces noise while simultaneously preserving the trend of the original sequence.

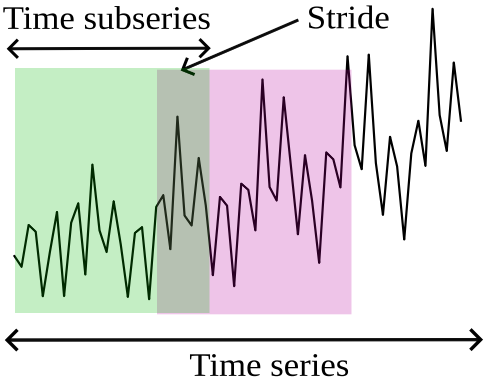


Fig. 3.14 An example of the time series data. Two windows (green and purple) represent subseries that partially overlap. The length of the overlapping region is called the *stride*.

Many time series analysis methods require calculating the distance between two series. We distinguish two basic kinds of such distances $\mathcal{D}$:

- between two different series $\mathcal{T} = t_1, t_2, \ldots, t_L$ and $\mathcal{S} = s_1, s_2, \ldots, s_L$ of the same length L:

$$\mathcal{D}(\mathcal{T}, \mathcal{S}) = \sqrt{\sum_{i=1}^{L} (t_i - s_i)^2}, \tag{3.12}$$

- between a time series $\mathcal{T}$ of length n and its subseries $\mathcal{R}$ of length k ($k < n$) which is defined as the Euclidean distance of $\mathcal{R}$ to its most similar segment in $\mathcal{T}$:

$$\mathcal{D}(\mathcal{R}, \mathcal{T}) = \min_{j=1,\ldots J} \mathcal{D}(\mathcal{R}, \mathcal{T}_j), \tag{3.13}$$

  where j in an index that runs over all possible subseries of the time series $\mathcal{T}$.

It is important to note that the above–presented definitions of the distances are not limited to the euclidean metrics and may be generalized to other, much more sophisticated metrics such as *dynamical time warping (DTW)* [91].

3.9.2 KNN algorithm

The KNN (*K–Nearest Neighbors*) method belongs to the class of supervised learning techniques. Despite its simplicity, it is considered one of the most efficient algorithms in time series classification problems [92, 93, 94]. The principle of this algorithm is straightforward and can be summarized in a few steps enumerated below:

1. Selection of the number of neighbors K and the distance metric d (optimal choice of these parameters can be found by the grid search over parameters' space),
2. Calculation of the distances of a new sample to K nearest points,
3. Assignment of the sample to the most frequent category of the K nearest points as presented in Fig. 3.15.

In physics, KNN can potentially be used to speed up MC simulations [95].

3.9.3 The shapelet method

The shapelet method is a novel data mining technique used for time series classification. In contrast to the more popular KNN algorithm, it focuses on extracting local representative features of the time series rather than the global ones. In many cases, it yields better classification accuracy. Another advantage of this method is that the generated results are easier to interpret, allowing a more insightful study of the data set.

In general, shapelets $\mathcal{S}$ are considered time series subsequences being maximally representative of a class [5]. Their identification requires searching over all possible

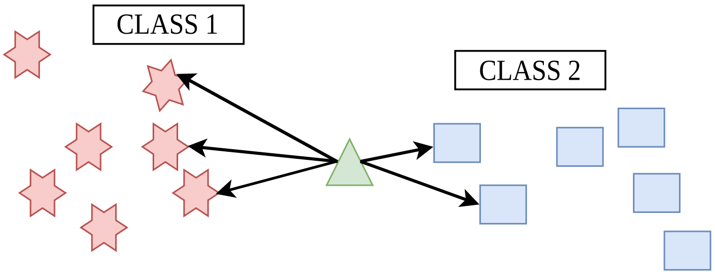


Fig. 3.15 The assignment of a new sample to the proper category. For a new sample marked with a green triangle, the distance to the five nearest points (neighbors) is calculated with some distance metrics (marked with black arrows). The assignment decision is based on the majority voting – in this case, the sample will be assigned to the first class.

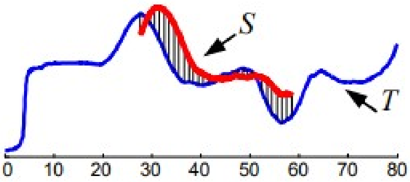


Fig. 3.16 A shapelet S presented along with the whole time series T. The vertical black lines illustrate the euclidean distance $ED(T, S)$. The figure is taken from [5].

subseries of length l (arbitrarily chosen) in the dataset for which the distance to the series belonging to one class is significantly larger than the distance to the series attributed to the other class. In this way, the distance of the shapelet to the time series can be regarded as a discrimination measure between two (or more) classes. The brute–force solution to the problem of finding optimal shapelets is time–consuming and, for more challenging classification problems, requires a lot of computational power. Fortunately, a method called *learning time–series shapelet* [96] significantly accelerates the computations. This algorithm can be summarized in two simple steps:

1. Starting from the random shapelet of the length l,
2. Learning the optimal shapelet by minimizing the classification loss function.

Let us describe the above mechanism in more detail. For simplicity, we assume only two classes of the time series data for which the labels $\mathbf{y_i}$ are either 0 or 1. Then the label $\hat{\mathbf{y}}_\mathbf{i}$ predicted by the algorithm can be calculated with the logistic regression model as [96]:

$$\hat{\mathbf{y}}_\mathbf{i} = \sum_{k=1}^{K} M_{i,k} W_k, \tag{3.14}$$

where W_k (*weights*) are free parameters learned during training. The coefficients $M_{i,k}$ indicate the distance of a time series T_i to a shapelet S_k according to the definition presented in the previous subsection (Sec.3.9.1). The *weights* and *biases* and indirectly shapelets S which are included in the definition of M are found by minimizing the loss function $\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})$:

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i=1}^{N} \big(\mathbf{y_i} \ln \sigma(\hat{\mathbf{y}}_\mathbf{i}) + (1 - \mathbf{y_i}) \ln(1 - \sigma(\hat{\mathbf{y}}_\mathbf{i}))\big), \tag{3.15}$$

where $\sigma(x)$ is the sigmoid. Note that the form of the objective function presented in Eq. (3.14) is not ultimate and can be extended by some regularization terms. The stochastic gradient descent algorithm finds parameters that minimize this function, just as in neural networks.

One can visualize the results of applying the shapelet method in two-dimensional (or three-dimensional) space. With this aim, it is sufficient to choose two (or three) most discriminant shapelets s_1 and s_2 and calculate their distances to all samples in a dataset $\mathbf{k}$. In such a way, two (or three) feature vectors $d(s_1, \mathbf{k})$ and $d(s_2, \mathbf{k})$ of the form

$$\begin{aligned} \mathcal{D}(s_1, \mathbf{k}) &= [\mathcal{D}(s_1, k_1), \mathcal{D}(s_1, k_2), \ldots, \mathcal{D}(s_1, k_n)] \\ \mathcal{D}(s_2, \mathbf{k}) &= [\mathcal{D}(s_2, k_1), \mathcal{D}(s_2, k_2), \ldots, \mathcal{D}(s_2, k_n)] \end{aligned} \tag{3.16}$$

are constructed, where $\mathcal{D}(s_j, k_i)$ stands for the distance of the subsequence k_i to the shapelet s_j.

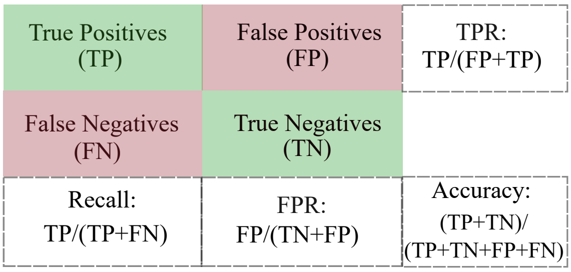
The presented method is highly effective in discriminating biomedical and food spectrography signals, medical images, and faces [97].

3.10 The performance metrics in ML

This Section presents the basic performance metrics used to evaluate machine learning models. One of the most common metrics applied in ML models is the confusion matrix (CM). For a binary classification problem, it is a 2×2 table, whose cells contain the number of correctly and incorrectly classified data (Fig. 3.17). The four matrix cells contain:

- **TP (True Positives)** – number of samples that are labeled as 1 (True) and are classified as such,
- **TN (True Negatives)** – number of samples that are labeled as 0 (False) and are classified as such,
- **FP (False Positives)** – number of samples that are labeled as 0 and are classified as 1,
- **FN (False Negatives)** – number of samples that are labeled as 1 and are classified as 0.

The distinction between well–classified and misclassified samples, on the basis of CM, is thus straightforward. Correctly and incorrectly classified samples are situated on the diagonal and on the off–diagonal elements of the CM, respectively.



True Positives (TP)	False Positives (FP)	TPR: TP/(FP+TP)
False Negatives (FN)	True Negatives (TN)	
Recall: TP/(TP+FN)	FPR: FP/(TN+FP)	Accuracy: (TP+TN)/ (TP+TN+FP+FN)

Fig. 3.17 The scheme of the confusion matrix. The table presents the number of correctly classified (green-shaded cells) and misclassified (red-shaded cells) data samples. There are also introduced common metrics associated with the definition of the confusion matrix (cells framed with dashed lines).

Based on the CM, one can define other metrics to assess the ML models. They are presented in Fig. 3.17. First of them is the **accuracy** (Acc) defined as the ratio of the number of correctly identified samples (TP + TN) to the number of all samples in a dataset (TP+FP+FN+TN). The other two are **true** and **false positive rates** (TPR and FPR). TPR measures the proportion of the number of correctly identified positive samples (TP) to the number of all data points that are predicted to be positive (TP + FP). Similarly, FPR illustrates the proportion of incorrectly classified positive samples (FP) to the number of negative samples (TN + FP). The last one is called the **recall**. It indicates the ratio of the number of correctly recognized positive samples to the number of all positive samples [55].

There also exists another quantity designed to present the diagnostic ability of a binary classifier. It is called *ROC* (*receiver operating characteristic*) curve. It shows the behavior of TPR as a function of FPR [98] for different discrimination thresholds TH $\in [0, 1]$, that is, various values differentiating between positive and negative samples. The overall discriminating power of the classifier is measured with the AUC–ROC (***A**rea **U**nder ROC **C**urve*) metrics defined as an integral under the ROC function. The higher the value of AUC–ROC, the better the classifier makes predictions. In general, when:

- AUC–ROC $= 1$, the classifier is ideal,
- AUC–ROC $= 0.5$, the classifier is random,
- AUC–ROC < 0.5, the classifier performs worse than the random one.

Examples of ROC curves are presented in Fig.3.18.

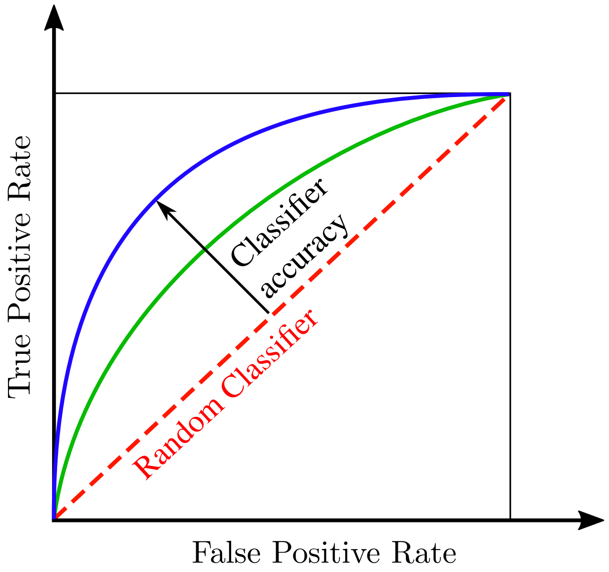


Fig. 3.18 The ROC curve illustrating TPR (*True Positive Rate*) as a function of FPR (*False Positive Rate*) for different discrimination thresholds. The closer the curve is to the upper left corner of the plot, the bigger the model's performance.

3.11 Summary

In this Chapter, we briefly reviewed the most popular machine learning algorithms used in condensed matter physics. It does not exhaust applications of machine learning techniques in this area. Our goal was to show these methods' potential in this field of physics and provide the fundamental concepts essential for understanding the practical part of this thesis.

CHAPTER 4

Results

In this Chapter, we present the main results of the studies conducted as part of the doctoral dissertation. We start by describing an attempt to determine the critical temperature of the BKT transition from the MC configurations (Sec. 4.1). We then introduce the method based on the unsupervised–learning techniques allowing us to identify the order parameter for various spin models and discriminate between strong first–order and second–order phase transitions (Sec. 4.2). The subsequent Section of this Chapter (Sec. 4.3) illustrates the results of the learning by confusion scheme applied to different microscopic models undergoing first– and second–order phase transitions. In the last Section (Sec. 4.4), we demonstrate the applicability of the novel data mining methods to analyze the ion channel activity.

4.1 A study of Berezinskii–Kosterlitz–Thouless phase transition with machine learning algorithms

This Section presents the machine learning approach to the Berezinskii–Kosterlitz–Thouless (BKT) phase transition. We apply it to three different spin models shortly introduced in Sec. 2.5: the classical XY model (**c–XY**), the quantum XY model (**q–XY**), and the phase–fermion model (**PF**). We identify the critical temperature T_C using classical methods of analysis and juxtapose the obtained results with the neural network output.

4.1.1 Estimation of the critical temperature based on the helicity modulus analysis

As we pointed out in Sec. 2.5, the Berezinskii–Kosterlitz–Thouless phase transition represents a special class of topological phase transitions. It is associated with a subtle change of the correlation function and the unbinding of the vortex–antivortex pairs at the critical point T_{BKT}. In contrast to conventional phase transitions, the BKT transition is not signaled by a peak in thermodynamic quantities, such as heat capacity or magnetic susceptibility. Nevertheless, it can be identified by the analysis of the *helicity modulus* Υ. It is a quantity defined as the second–derivative of the free energy with respect to an externally imposed global twist of spins [99]. For an arbitrary lattice size L, Υ approaches the value [100]:

$$\Upsilon\left(T_{BKT}(L)\right) = \frac{2}{\pi} T_{BKT}(L), \tag{4.1}$$

where $T_{BKT}(L)$ is the critical temperature. Knowing Υ from the MC simulations performed for a given lattice size L, we can establish which temperature satisfies the condition imposed in Eq. (4.1) and associate it with $T_{BKT}(L)$. After repeating this procedure for different lattice sizes, we can apply the finite–size scaling technique and provide a rough estimate of the critical temperature T_{BKT} in the thermodynamic limit.

We adopt the above–described method to calculate the value of T_{BKT} for the three models mentioned at the beginning of this section, i.e., the c–XY, q–XY and PF models. The first step requires the generation of spin configurations for different lattice sizes. For this purpose, we perform MC simulations. In the case of the c–XY model, we use the usual Metropolis MC algorithm. It cannot be straightforwardly applied to the PF model, which involves, apart from classical, also quantum degrees of freedom. The modified version of the Metropolis algorithm used to simulate this model is analogous to that applied to the FK model (Appendix C). The main difference between the two approaches is that in this case, we propose a new state by changing the spin direction on one random lattice site (instead of changing the ion position as in the FK model). For the q–XY model, the spin configurations generated from quantum MC simulations are provided by our collaborators [70].

In order to calculate the helicity modulus Υ from the obtained MC configurations, we use the following formula [101]:

$$\Upsilon = \frac{1}{L^2}\left(\left\langle \sum_{<i,j>} \cos(\theta_i - \theta_j)(\hat{x}\cdot\hat{\varepsilon}_{ij})\right\rangle - \frac{1}{T}\left\langle (\sum_{<i,j>} \sin(\theta_i - \theta_j)\hat{x}\cdot\hat{\varepsilon}_{ij})^2\right\rangle\right), \tag{4.2}$$

where L is the size of the lattice, $\hat{x}$ and $\hat{y}$ are unit vectors representing two perpendicular directions in the plane of the lattice, T denotes the temperature, and $\hat{\varepsilon}_{ij}$ indicates the direction of the bonds $i-j$. Knowing Υ, we estimate $T_{BKT}(L)$ for different lattice sizes L as the crossing of Υ with the line $\frac{2}{\pi}T$. Then, using the fact that $T_{BKT}(L) - T_{BKT} \propto \log(L)^{-2}$ [101, 102] we find the value of T_{BKT} in the thermodynamic limit. The results of the applied procedure are presented in Fig. 4.1 for all three models.

A more accurate value of T_{BKT} can be determined from the solutions of the Kosterlitz renormalization group (RG) equations [26]. They illustrate a general dependence of the $\Upsilon(T_{BKT}(L)) = \Upsilon(L)$ obtained for a given lattice size L on its value $\Upsilon(T_{BKT}) = \Upsilon(L = \infty) = \Upsilon(\infty)$ in the thermodynamic limit [103, 104]:

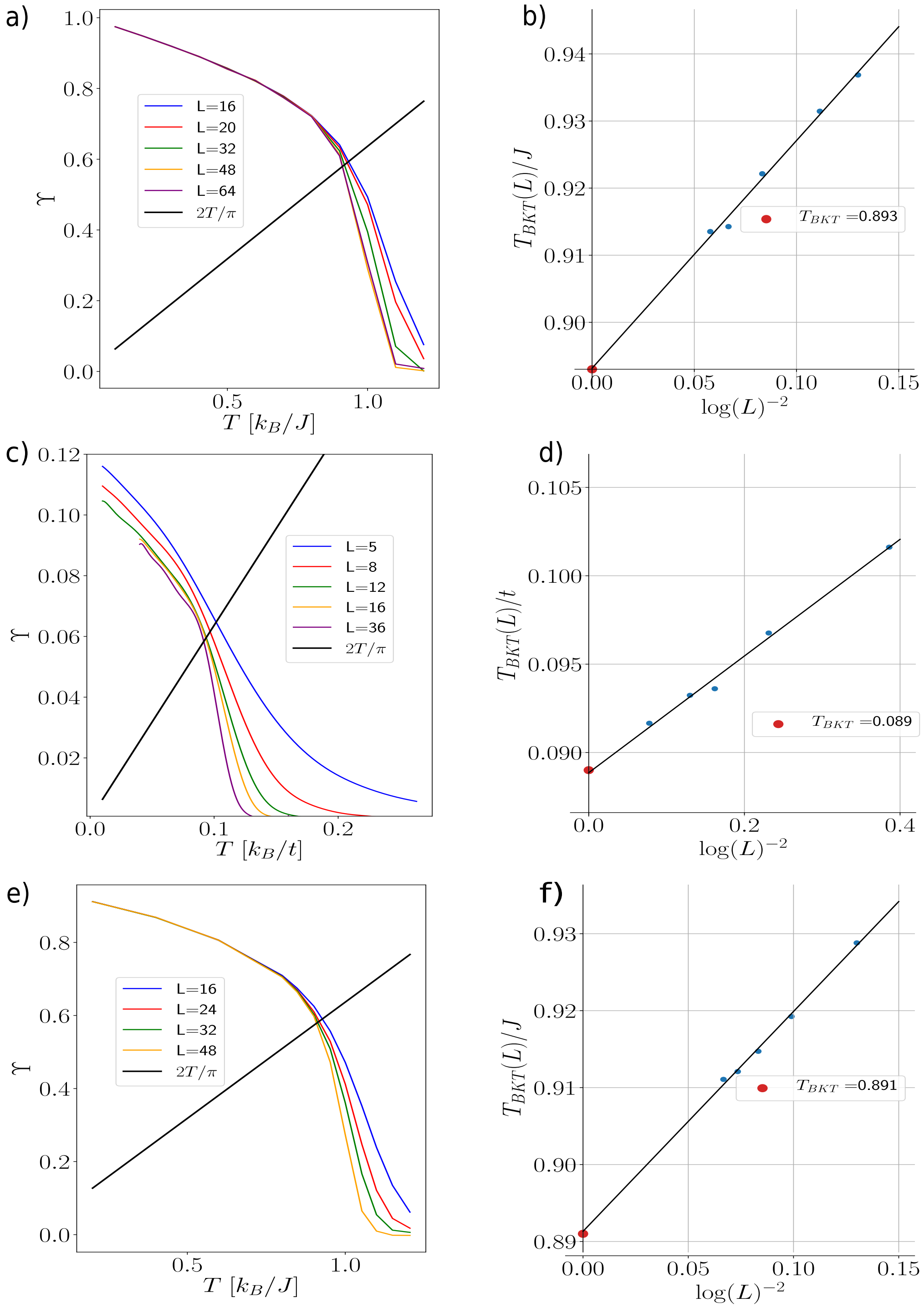


Fig. 4.1 The helicity modulus Υ as a function of temperature for c–XY model a), PF model with $g = 4t$ and c) q–XY models with $E_C = 0.1$ e). The right panel of the figure illustrates the procedure of finite–size scaling of the critical temperatures $T_{BKT}(L)$. Blue dots indicate values of $T_{BKT}(L)$ obtained from the crossings of $\Upsilon(L)$ with the line $\frac{2T}{\pi}$. The black lines are the linear fits to the set of estimated $T_{BKT}(L)$. Crossings of these lines with the vertical axis are associated with the critical temperature T_{BKT} obtained in the thermodynamic limit.

$$\Upsilon(L) = \Upsilon(\infty)\left(1 + \frac{1}{2}\frac{1}{\ln(L) + C}\right), \tag{4.3}$$

where C is an undetermined constant. Making use of the fact that $T_{BKT}(\infty)$ also obeys the Eq. (4.1), we can insert the expression $\Upsilon(\infty) = \frac{2}{\pi} T_{BKT}$ to the Eq. (4.3). As a result, we obtain:

$$\Upsilon(L) = \frac{2}{\pi} T_{BKT} \left(1 + \frac{1}{2} \frac{1}{\ln(L) + C} \right). \tag{4.4}$$

It is important to emphasize that Eq. (4.4) applies only to values of $\Upsilon(L)$ obtained at the critical temperature $T_{BKT}(L)$. Therefore, the fit of Υ corresponding to different lattice sizes L should yield the smallest error exactly at T_{BKT}. Taking this into account, at the fixed temperature T we can try to fit various $\Upsilon(L)$ corresponding to different lattice sizes and temperature T to Eq. (4.4). The constant C is the parameter of this fit. Repetition of this procedure for many different values of T allows us to find a temperature for which the fit to Eq. (4.4) returns the smallest error. We then associate it with the critical temperature in the thermodynamic limit. The results of applying the above-described method are, for all the three models, illustrated in Fig. 4.2. The blue stars indicate values of δ corresponding to different temperatures T. Their positions are fitted with two linear functions (solid orange lines), the crossing of which determines the critical temperature T_{BKT} in the thermodynamic limit. We see that the results obtained with this technique are compatible with those presented in Fig. 4.1 and acquired by other authors [30, 105, 106, 107, 108].

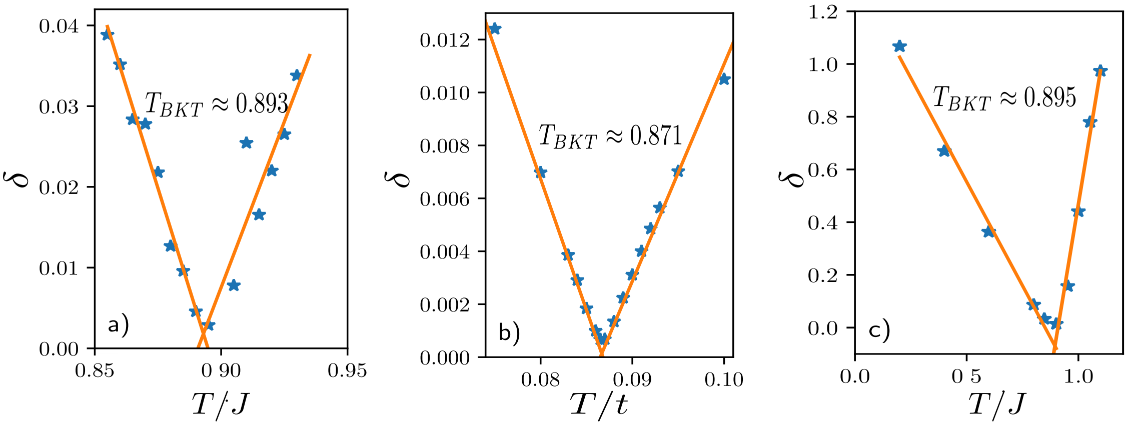


Fig. 4.2 The root mean square error δ of fitting Υ obtained from MC simulations to Eq. (4.3) for the c-XY model (a), the PF model (b), and the q-XY model (c). The blue stars indicate values of δ corresponding to different temperatures T. Solid orange lines show fits of two linear functions fitted to the data points. Their crossing point determines critical temperature T_{BKT} in the thermodynamic limit.

4.1.2 Estimation of the critical temperature with the neural network

In this Section, we verify whether an alternative approach based on the ML can successfully identify a topological phase transition in the models discussed above. Our attempt is motivated by the fact that such techniques are capable of correctly indicating the critical temperature of Ising–type [63, 109, 110] and quantum phase transitions [110, 111, 112]. The problem of estimating T_{BKT} with ML methods is well known in the literature and has been tackled by other authors. Obviously, correct identification of the Berezinskii–Kosterlitz–Thouless phase transition requires the recognition of topological defects such as vortices and antivortices. However, it is much easier for a neural network to find the difference between low– and high–temperature phases based on magnetization M, which according to the Mermin–Wagner theorem [23] can not be treated as an indicator of the BKT phase transition. To overcome this issue, most authors try some feature engineering on the raw spin configurations before feeding them into a neural network. They transform MC samples into vortex configurations [69], histogram of spin orientations [113], or spin–correlation functions [72]. Yet in another paper, the weight matrices are analyzed to estimate the critical temperature [114].

In our approach, we start with an analysis similar to that presented in [63] and briefly described in Sec. 3.6. More precisely, we train a neural network on MC configurations belonging to the low–temperature (T_0) and high–temperature (T_1) phases. Next, we test predictions of such a trained network on configurations not yet seen generated at intermediate temperatures lying in the interval $T \in (T_0, T_1)$. The neural network returns the mean probabilities that the tested configurations belong to the high–temperature phase $P(HTP)$. Then the temperature for which $P(HTP) = 0.5$ is associated with the critical point.

On the basis of the results, we confirm the conclusion made in the other papers: a simple feed–forward neural network without any extra preprocessing steps cannot correctly identify the critical temperature of the BKT phase transition. Therefore, to make predictions more accurate, we transform the MC configurations represented by angles $(\theta_1, \theta_2, \ldots, \theta_N)$ to vectors of trigonometric functions of the form:

$$(\cos\theta_1, \cos\theta_2, \ldots \cos\theta_N, \sin\theta_1, \sin\theta_2, \ldots, \sin\theta_N)\,.$$

In this way, we take into account the continuous character of spin directions, which can take any value from the interval $\langle 0, 2\pi)$. The transformation implies that almost parallel spins corresponding, for example, to the angles 0.01π and 1.99π, although numerically distant, are represented by close numbers. To summarize, the entire procedure includes several steps:

1. Generation of MC configurations at temperatures ranging from $0.1J$ to $1.6J$, $0.1J$ to $1.5J$ and $0.02t$ to $0.2t$ for the c–XY, q–XY and PF models, respectively. For the q–XY and PF models, the number of configurations is increased by operations of rotations and translations.

2. Transformation of raw spin configurations represented by angles $(\theta_1, \theta_2, \ldots, \theta_N)$ to vectors of trigonometric functions of the form:

$$(\cos\theta_1, \cos\theta_2, \ldots \cos\theta_N, \sin\theta_1, \sin\theta_2, \ldots, \sin\theta_N)\,.$$

3. Repetitive training of the neural network on configurations corresponding to m temperatures lying below T_{BKT} and above T_{BKT} (assuming that the arrangement of temperatures around the critical point is symmetric).

4. Test of the neural network on not yet seen MC configurations. The evaluation of the predicted critical temperature and statistical errors of this estimation.

Repetitive training of a neural network consists of applying the 10–cross validation technique [115] repeated 10 times, each time starting from different initial conditions. In this manner, we obtain the total number of 100 values of $P(HTP)$ for each temperature, which is sufficient to measure the uncertainty of a neural network's predictions. A similar analysis has already been performed in [73]. The computational details regarding the neural network architecture and the training process are presented in Appendix A.

The results from applying the above-mentioned steps suggest that the estimated critical temperature strictly depends on the choice of T_0 and T_1. It may stem from the fact that configurations corresponding to the deep low–temperature phase differ from those generated at the temperature closer to the critical point (but still in the low–temperature phase). The neural network learns in these two cases different features that change the prediction of T_{BKT}. To further investigate this problem, we try to answer the following questions: (i) How does the choice of temperatures T_1 and T_2 affect the value of the predicted critical temperature? (ii) What is the optimal distance between T_0 and T_1 that allows one to discriminate between configurations corresponding to temperatures slightly below and above T_{BKT}? To address these queries, we perform a thorough analysis of T_{BKT} for several sets of training data generated for different T_0 and T_1.

For all three models, we illustrate the results obtained as a function of the lowest and highest temperatures T_0', T_1' lying below T_{BKT} taken into the training. We express them in relation to the actual value of the critical temperature T_{BKT} (i.e., indicated by the solutions to the RG equations) with the auxiliary variable τ:

$$\tau = \frac{T_1' - T_0'}{T_{BKT} - T_0'}. \tag{4.5}$$

The estimated values of the probabilities P and the corresponding statistical errors for different values of m and τ are for three analyzed models illustrated in Figs. 4.3, 4.4, and 4.5.

As for the c–XY model, we obtain a quite accurate value of the critical temperature even when the neural network is trained only on two extreme temperatures ($m = 1$, $\tau = 0$) representing fully ordered and completely random spin states. At the same time, we observe that the neural network is extremely uncertain of its own predictions for temperatures lying in the vicinity of the phase transition. This result may suggest that in order to discriminate between configurations corresponding to temperatures $T \approx T_{BKT}$ characterized by large thermal fluctuations, learning the 'pure' low and high-temperature phase is not sufficient. Indeed, as can be inferred from Fig. 4.3, an increase in τ substantially decreases the prediction errors. In Fig. 4.3d) we show T_{BKT} plotted in a wide range of τ. It can be seen that T_{BKT} starts to converge to the actual value of $\tau \geq 0.65$, reaching the highest accuracy for $\tau \approx 0.95$.

It should be pointed out that, contrary to what one might expect, that the neural network does not learn magnetization M. To demonstrate this, we add to Fig. 4.3c) a green dashed line representing M. As can be noticed, it clearly deviates from the solid red line corresponding to the probabilities returned by the ML model. It suggests that the neural network learns more complicated, non–linear features hidden in MC configurations.

When it comes to the q–XY model, the spread of probabilities (standard deviations) is much smaller than that of its classical counterpart. Furthermore, it turns out that $m = 3$ is sufficient to obtain a prediction deviating from the actual one only by 2%.

For the PF model, the situation is a little different. Accurate prediction of critical temperature requires configurations generated at temperatures $T \approx T_{BKT}$ in the process of training – the convergence to the actual value is reached for $m = 6$. For $m = 1$ the estimation error is approximately equal to 20%. The spread of probabilities is also quite large, which indicates that the obtained results are affected by a non–negligible error. This probably stems from the fact that the configurations corresponding to the temperatures lying in the vicinity of the phase transition are much more different from those generated at extreme low and high temperatures. This may be due to the character of this model. While in the c–XY and q–XY models the interaction is direct and limited only to the nearest neighbors, in the case of the PF model we observe the effective interaction mediated by fermions distributed over all lattice. This effect manifests itself in the form of Υ (Fig. 4.1). For the c–XY and q–XY models, the helicity modulus converges even for small lattice sizes. This does not happen for the PF model, for which the values of Υ (and hence the energies) strongly depend on the size of the system. It is a result of the delocalization of fermions, which behave as if they were confined in an infinite quantum well. In this case, the energies of the particles depend on the size of this well.

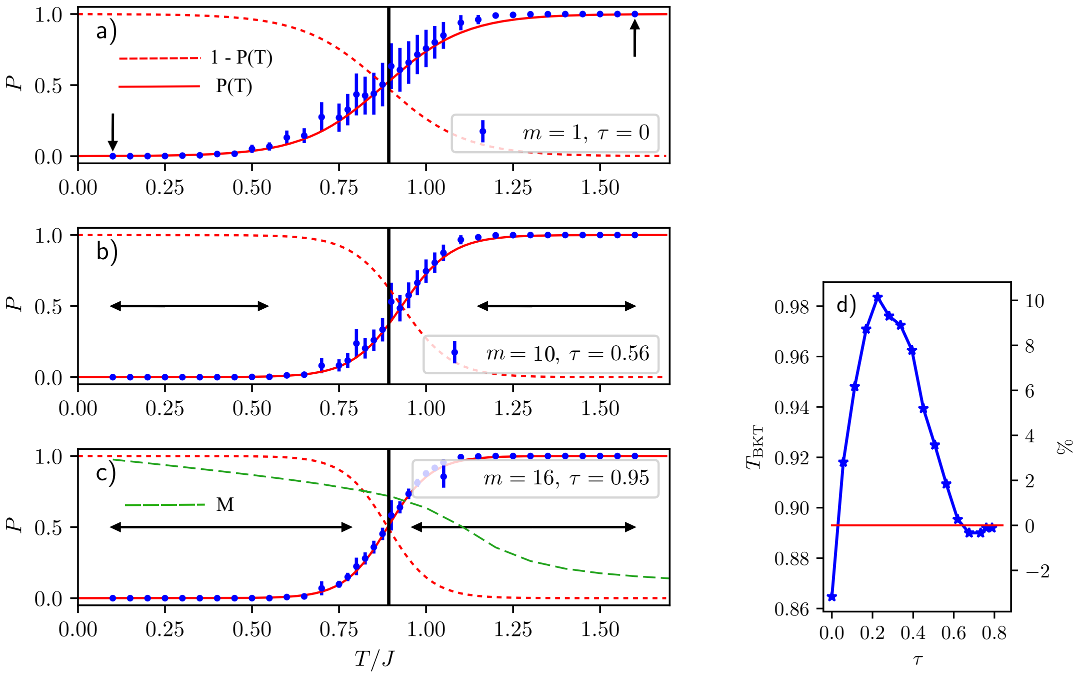


Fig. 4.3 The probability P of configurations to belong to the high–temperature phase of the c–XY model for (a) $m = 1$, (b) $m = 10$, and (c) $m = 16$. The vertical error bars show the standard deviations. The solid red line illustrates the best fit to the function of the form $P(T) = 0.5 \tanh\left[\alpha\left(T - T_{\rm BKT}\right)\right] + 0.5$, where α and $T_{\rm BKT}$ are fitting parameters. The dashed red line shows $1 - P(T)$ interpreted as the probability of a configuration belonging to the low–temperature phase. The black arrows indicate $2m$ temperatures used during the training (m temperatures corresponding to the low–temperature phase and the same number to high–temperature phase, equidistant from the actual critical temperature T_{BKT} marked by the solid black vertical line). The dashed green line in panel c) shows magnetization M. The quantity τ is defined in Eq. 4.5. The temperatures on which the neural network is trained range from $0.10\ J$ to $0.70\ J$ and from $1.10\ J$ to $1.60\ J$ with stepsize $0.05J$ and from $0.750\ J$ to $1.050\ J$ with stepsize $0.025J$. On the right panel one presents the function of predicted temperature T_{BKT} on τ and the error of this prediction expressed in $\%$. The red horizontal line represents the true critical temperature.

4.1.3 Summary

The studies presented in this Section verified the capacity of a neural network to identify the Berezinskii–Kosterlitz–Thouless phase transition. We used a simple feed–forward neural network and fed it with spin configurations transformed into the trigonometric representation. We repeated the procedure for three different models. We found that such a constructed neural network does not learn the global features of configurations such as magnetization. The results depicted in Fig. 4.3 suggest that the slope of magnetization is much smaller than the probability function P obtained from the neural network computations. We suppose that the ML model is able to learn some topological features. On the other hand, it cannot grasp the subtle change in the spin–spin correlation function that occurs at the critical point. Otherwise, it would accurately indicate the actual value of T_{BKT}.

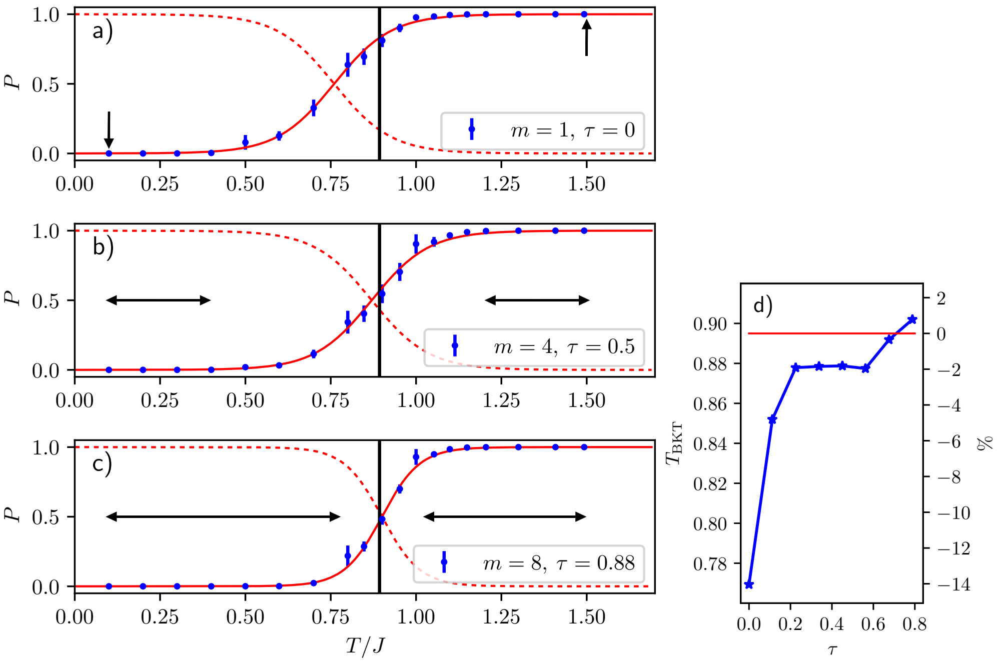


Fig. 4.4 The same as in Fig. 4.3 but for quantum XY model. The temperatures on which the neural network is trained range from $0.1\ J$ to $0.8\ J$ and from $1.2\ J$ to $1.5\ J$ with stepsize $0.1\ J$ and from $0.85\ J$ to $1.15\ J$ with stepsize $0.05\ J$.

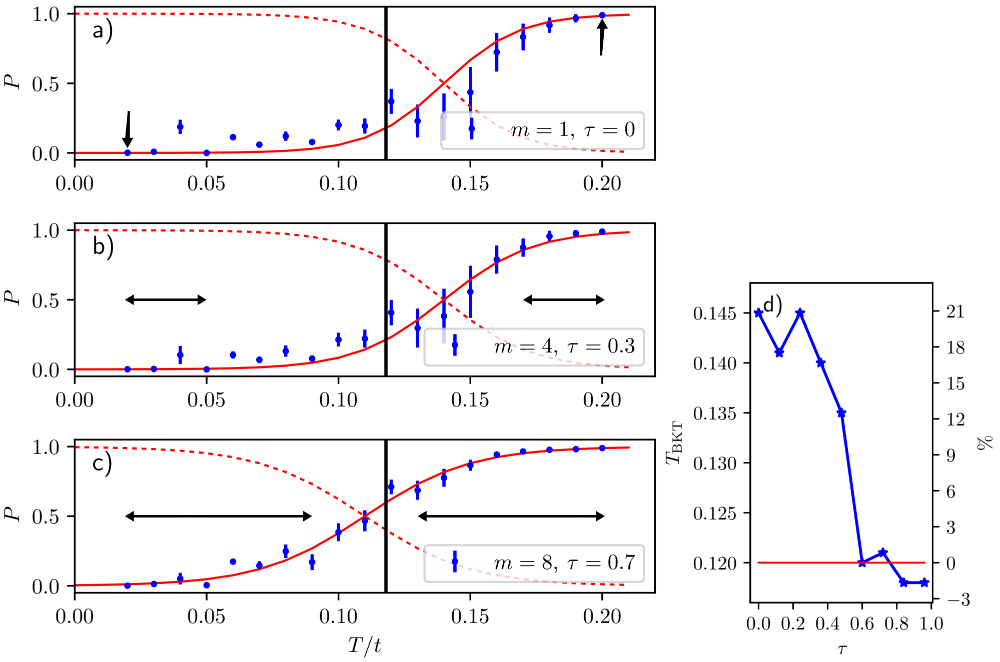


Fig. 4.5 The same as in Fig. 4.3 but for PF model. The temperatures on which the neural network is trained range from $0.02t$ to $0.20t$ with stepsize $0.01t$.

Furthermore, we have shown that for all the three models studied, the prediction of T_{BKT} strictly depends on the number of temperatures $2m$ used for training. As expected, the higher the number of various configurations (higher value of m) in the data set, the better the accuracy of the predictions. However, the minimal value of m required to obtain the precise critical temperature affected by the small error varies between the models analyzed. For the c–XY and q–XY models, it is sufficient to take configurations relatively distant from the critical point to obtain a pretty accurate value of the critical temperature. For the PF model, the situation is different. To receive the correct value of T_{BKT}, it is necessary to train the neural network at the temperatures $T \approx T_{BKT}$. It suggests that the configurations corresponding to the fully ordered state and those representing a highly disordered phase are much more different from those generated at the temperatures lying in the vicinity of the transition. It may come as a consequence of the different characteristics of the PF model. Here, we do not observe the direct interaction between neighboring spins as in the case of the other two models. Instead, we deal with an effective long-range interaction mediated by the itinerant fermions. This phenomenon is responsible for the unusual properties of the PF model manifested, for example, in the character of the helicity modulus Υ.

However, to properly determine the value of the critical temperature, it is beneficial to train the neural network not only on extreme temperatures that are far from the critical point, but also to the configurations corresponding to the temperatures $T \approx T_{BKT}$ containing features crucial for the accurate identification of the BKT transition.

4.2 Unsupervised learning techniques as a tool to study phase transitions

One of the crucial problems regarding the physics of phase transitions is determining the order parameter. Usually, it is represented by a quantity that is non-zero in the low–temperature ordered state and drops to zero above the critical point (Sec. 2.3). For instance, in the Ising model magnetization plays the role of the order parameter.

Whereas for many models, the order parameter can be easily identified, there are systems for which this task is quite challenging. Examples of such systems are topological insulators [116], quantum spin hall states [117], or spin glasses [118]. Therefore, developing new techniques that specify parameters that characterize phase transitions is crucial for a more profound understanding of these phenomena.

Here, we introduce a simple method that involves two basic unsupervised machine learning techniques: PCA and k–Means (more accurate descriptions of these methods can be found in Sec. 3.3, 3.4). The main objective of the PCA method is to reduce the dimensionality of the dataset and simultaneous disclosure of its essential characteristics. Recent studies have confirmed this algorithm's utility in determining the order parameter for various systems [53, 119], its ability to discriminate between different phases, and estimate the critical point [56, 120]. It has been shown that for simple magnetic systems, the mean absolute value of the first principal component $\langle|p_1|\rangle$ coincides with an order parameter [121]. We follow this proposal and verify if $\langle|p_1|\rangle$ can serve as an order parameter for two simple spin models of condensed matter physics: the Potts and Blume–Capel models, introduced in Sec. 2.5. We juxtapose the obtained value of the first principal component $\langle|p_1|\rangle = \langle|FC|\rangle$ with another, newly proposed order parameter constructed from the PCA representation. We extend these studies to the FK model, which, apart from classical, also exhibits quantum degrees of freedom. Furthermore, we demonstrate that the PCA, in conjunction with the k–Means clustering technique, can discriminate between strong first– and second–order phase transitions.

4.2.1 Determination of the order parameter

It is known that a system usually falls into one of its ground states in the low–temperature phase. For instance, in the Ising model, the spontaneous breaking of the rotational symmetry leads to the development of two equally probable ferromagnetic states. It is shown in Fig. 4.6 where we plot the probability distribution of magnetization $P(M)$ out of MC-generated data. For the temperature below the critical point, $T < T_C$ (Fig. 4.6 a)), the two peaks of $P(M)$ at $M = 1$ and $M = -1$ clearly indicate the presence of two possible spin directions: either *up* or *down*. We can distinguish two separated Gaussians, which in the vicinity of the phase transition ($T \approx T_C$, Fig. 4.6 b)) start to expand and overlap. In the high–temperature phase, i.e., for $T > T_C$ (Fig. 4.6c)), they turn into one Gaussian centered at $M = 0$.

In Fig. 4.7, we show the results of PCA applied to the same data set as in Fig. 4.6. The top row displays the distribution of raw spin configurations projected on a 2–dimensional space by the PCA algorithm. The bottom row illustrates the corresponding probability distribution of the first principal component $P(FC)$. A striking similarity between $P(M)$ and $P(FC)$ is evident: $P(FC)$ qualitatively follows the same temperature dependence as $P(M)$.

For other models studied, we observe similar results of PCA. Here, we demonstrate results for the q–state Potts (***q*P**), Blume–Capel (**BC**), and the half–filled Falicov–Kimball (**FK**) models described in Sec. 2.5. They undergo two kinds of phase transition depending on the Hamiltonian parameters. We choose the parameters such that the transition is either strong first– or strong second–order one (Tab. 4.1). It is worth highlighting that all models discussed here are characterized by discrete degrees of freedom. The spin in the *q*P model can take one of q equally spaced directions. In the BC model for spin $S = 1$, a lattice site is either unoccupied or occupied by a spin pointing up or down. The FK model allows for two possibilities: a site occupied by a heavy (localized) particle or a vacant one. Thus, the *q*P, BC, and FK are described by *q*, three, and two degrees of freedom, respectively. Its number coincides with the number of ground states for *q*P and FK models. The situation is different for the BC model. Although this model allows for three degrees of freedom, it is characterized only by

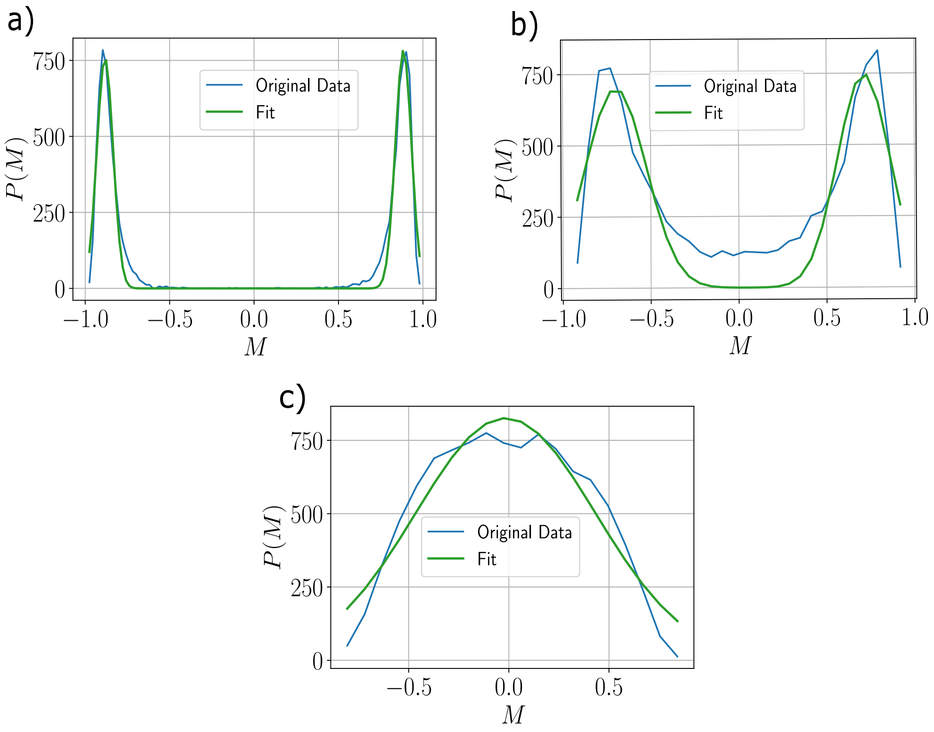


Fig. 4.6 Probability distribution $P(M)$ of magnetization M for the 2–dimensional Ising Model: a) at low temperature $T < T_C$; b) in the vicinity of the phase transition $T \approx T_C$; c) in a high temperature $T > T_C$. The solid blue lines represent $P(M)$ generated from MC simulations, while solid green lines are fits to the numerical data. For a) and b) the fitting function is a bimodal Gaussian distribution, for c), data is fitted to a single Gaussian distribution. MC simulations are performed for lattice size $L = 16$.

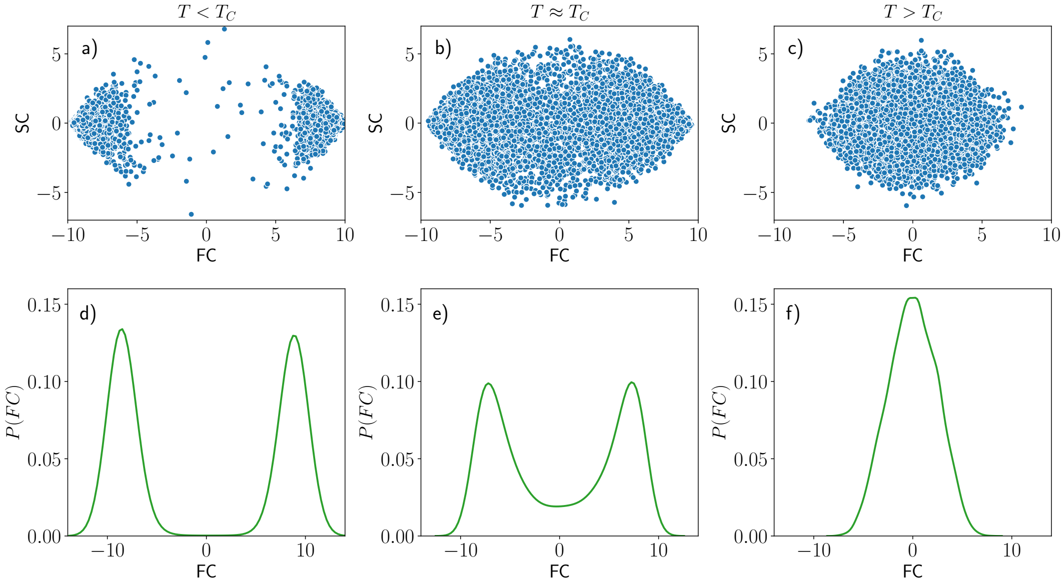


Fig. 4.7 The PCA representation of the Ising configurations (top rows) for a) $T < T_C$, b) $T \approx T_C$ and c) $T > T_C$. The corresponding probability distributions of the first principal component $P(FC)$ are shown in the bottom row. The data correspond to the lattice size $L = 16$.

two ground states.

Model	Continuous phase transition	Discontinuous phase transition
FK	U=4.0	U=0.5
***q*P**	q=2	q=10
BC	D=0	D=1.98

Table 4.1 Hamiltonians' parameters of three models used in this study. FK, qP and BC are abbreviations from Falicov–Kimball, q–state Potts and Blume–Capel models, respectively. A middle column indicates the parameters for which presented models undergo second–order (continuous) phase transition. The last column illustrates the parameters for which the transition is of first–order (discontinuous).

The PCA transformation of MC configurations representing 10–state Potts in different temperatures is presented in Fig. 4.8. The results obtained for the remaining models listed in Tab. 4.1 are shown in Figs 4.7 (Ising model overlaps with the 2–state Potts model), 4.9, 4.10, 4.11 and 4.12. It can be observed that for all models, in the low-temperature phase ($T < T_C$), data points are collected in separate clusters. As expected, their number is equal to the number of ground states. In the vicinity of the phase transition ($T \approx T_C$), the distances between these clusters drop abruptly. However, we can still discern bunches of overlapping points. In the high–temperature phase, the transformed MC samples create one large, uniform cloud from which one cannot isolate any group of samples. We observe a slight deviation from the above–mentioned description for the BC model $D = 1.98$ (Fig. 4.10). In this case, in the low–temperature phase, one can distinguish two separate clusters corresponding to two magnetization directions. At $T \approx T_C$, an additional cluster appears in the central part of the plot (Fig. 4.10 b)). Plascak et al. [122] have recently discussed the presence of this metastable state and associated it with three possibilities of lattice site occupation allowed by this model.

In order to quantitatively describe the differences in PCA results, we introduce the mean distance $\mathcal{D}$ between the clusters. We apply the k–Means algorithm (Sec. 3.4) to PCA–transformed MC configurations, which divides the data points into groups and identifies their centers c_i. In each case, we set the number of clusters k the same as the number of ground states. Next, we calculate the mean distance $\mathcal{D}$ between two consecutive centers of the clusters $c_i = (x_i, y_i)$ and $c_{i+1} = (x_{i+1}, y_{i+1})$ using the formula:

$$\mathcal{D} = \frac{1}{k-1} \sum_{i=1}^{k-1} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2}, \qquad (4.6)$$

which is a measure of the separation level between emerging ground states as a function

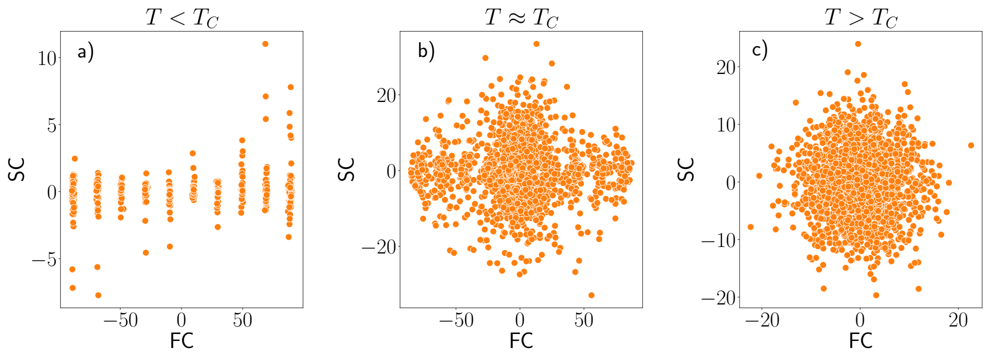


Fig. 4.8 The PCA representation of spin configurations corresponding to 10–state Potts model (linear size of the lattice $L = 16$) for the temperature $T < T_C$ (a)), $T \approx T_C$ (b)) and $T > T_C$ (c)).

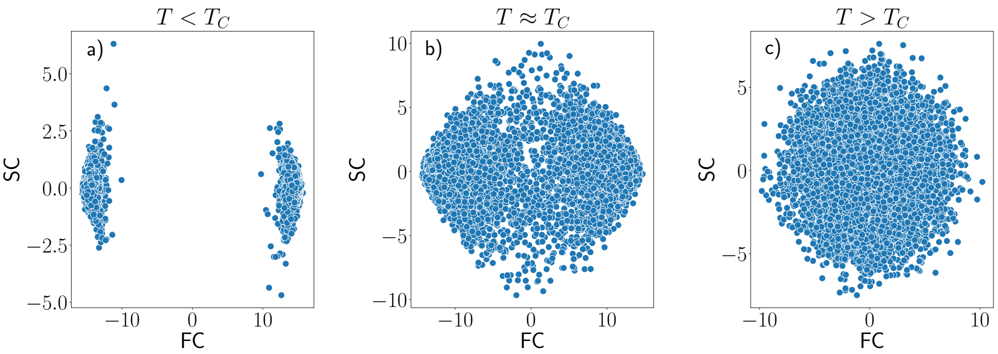


Fig. 4.9 The PCA representation of spin configurations corresponding to the Blume–Capel model $D = 0$ (linear size of the lattice $L = 16$) for the temperature $T < T_C$ (a)), $T \approx T_C$ (b)) and $T > T_C$ (c)).

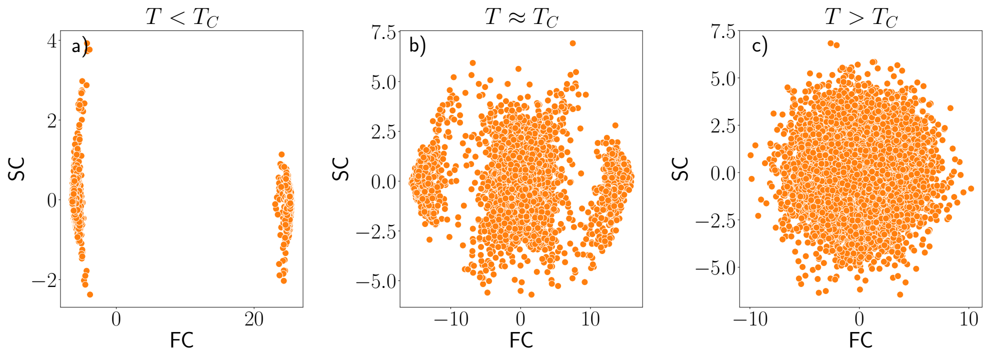


Fig. 4.10 The PCA representation of spin configurations corresponding to the Blume–Capel model $D = 1.98$ (linear size of the lattice $L = 16$) for the temperature $T < T_C$ (a)), $T \approx T_C$ (b)) and $T > T_C$ (c)).

of temperature. The distance $\mathcal{D}$ calculated for the Ising configurations in the low–temperature phase is illustrated in Fig. 4.13.

We then compare $\mathcal{D}$ with the actual order parameter, which for the Potts and Blume–Capel models corresponds to global magnetization M. For the FK model, following [41], we associate the order parameter with the renormalized density–density correlation function G_n, which captures short and long correlations between ions. It is

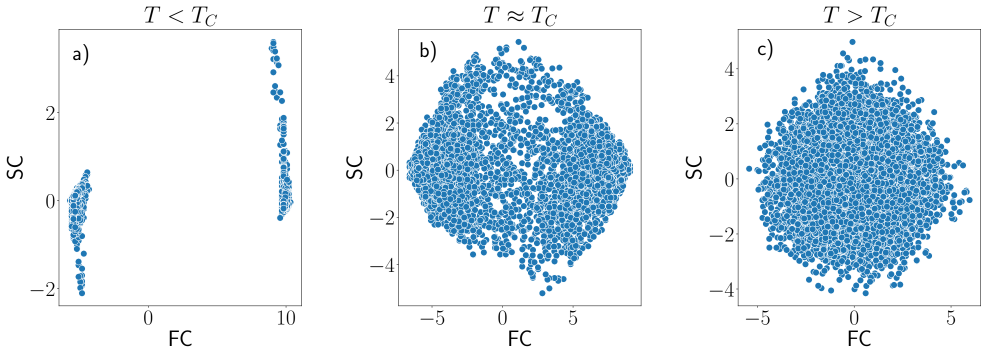


Fig. 4.11 The PCA representation of MC configurations corresponding to the Falicov–Kimball model $U = 4.0$ (linear size of the lattice $L = 16$) for the temperature $T < T_C$ (a)), $T \approx T_C$ (b)) and $T > T_C$ (c)).

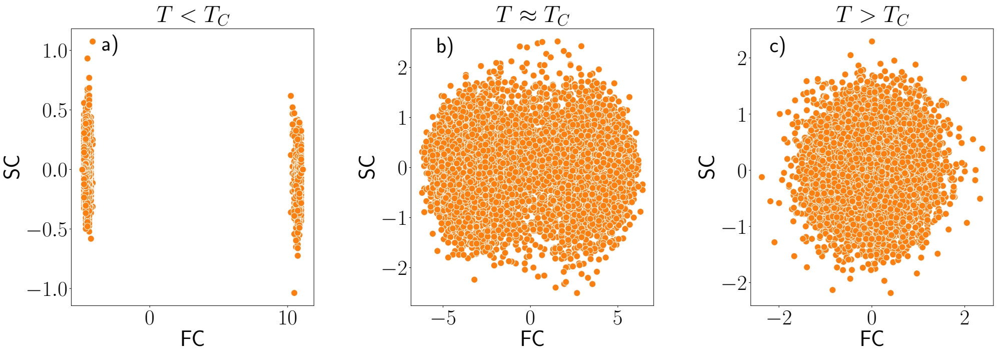


Fig. 4.12 The PCA representation of MC configurations corresponding to the Falicov–Kimball model $U = 0.5$ (linear size of the lattice $L = 16$) for the temperature $T < T_C$ (a)), $T \approx T_C$ (b)) and $T > T_C$ (c)).

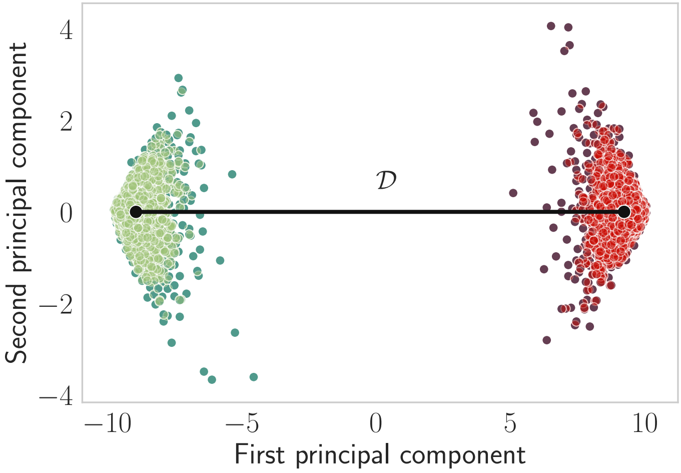


Fig. 4.13 The black horizontal line indicates the value of $\mathcal{D}$ defined as the distance between clusters' centers determined by the k–Means algorithm.

defined as:

$$G_n = -4(g_n - 0.25), \tag{4.7}$$

where

$$g_n = \frac{1}{4N} \sum_{i=1}^{N} \sum_{\tau_1,\tau_2=\pm n} w(r_i) w(r_i + \tau_1 \hat{x} + \tau_2 \hat{y}). \tag{4.8}$$

In Eq. (4.8) N denotes the number of lattice sites, $\hat{x},\ \hat{y}$ are the unitary unit vectors, and $w(r_i) = w_i \in \{0, 1\}$ indicates whether the lattice site i is occupied by an ion ($w_i = 1$) or vacant ($w_i = 0$). This definition of the order parameter ensures that it is equal to 1 for the checkerboard state (the ordered state corresponding to low temperatures) and close to 0 for randomly distributed ions (Sec. 2.5).

To begin with, we compare $\mathcal{D}$ with the mean value of the first principal component $\langle |p_1| \rangle$ and magnetization M for the qP and BC models. The results are presented in Fig. 4.14. We find that in all cases, $\langle |p_1| \rangle$, as well as $\mathcal{D}$, correlates, up to a normalization constant, with magnetization M. In the case of the 2–state and 10–state Potts model $\langle |p_1| \rangle$ fits almost perfectly to M except for the tiny fluctuations in the region before phase transition. On the other hand, we observe a slight deviation of $\mathcal{D}$ compared to M for the temperatures $T > T_C$ in the case of the 10–state Potts model. This deflection can be attributed to the fact that the algorithm has to optimally divide the large and coherent cloud of data points into 10 clusters, which is a challenging task. As for the BC model, we obtain perfect compatibility of $\langle |p_1| \rangle$ with M in both the cases under study, i.e., for $D = 0$ and $D = 1.98$. When it comes to $\mathcal{D}$, it correlates well with M for $D = 0$, but we observe large fluctuations of this quantity in the vicinity of the phase transition for $D = 1.98$. We suspect that the emergence of the additional metastable state (Fig. 4.10b) is responsible for the weird behavior of $\mathcal{D}$ in this temperature region – the k–Means algorithm tries to fit two clusters to a three-cluster pattern, resulting in a significant error in the order parameter estimation.

The situation is the most interesting for the FK model. As has been observed in Fig. 4.15, for $U = 0.5$ the $\langle |p_1| \rangle$ and $\mathcal{D}$ correlate with functions G_1 and G_2. Still, this correlation is smaller when compared to the case of magnetic systems (Fig. 4.14): $\langle |p_1| \rangle$ and $\mathcal{D}$ do not drop to zero in a high–temperature phase. For the bigger value of potential $U = 4.0$, the quantities $\mathcal{D}$ and $\langle |p_1| \rangle$ fit better to G_2 than to G_1 function, especially for $T > T_C$. We suspect that the finite size of the system is responsible for the discrepancy in the rate at which long G_2 and short correlations G_1 drop to zero. It is possible that for larger lattice sizes, the emerging differences between G_1 and G_2 would disappear, and both would converge to the predicted order parameter $\mathcal{D}$. Such a scenario is most likely for $U = 4.0$ – it is known that for a large value of interaction, a system belongs to the same universality class as the Ising model. However, these hypotheses have not yet been confirmed and require further investigation.

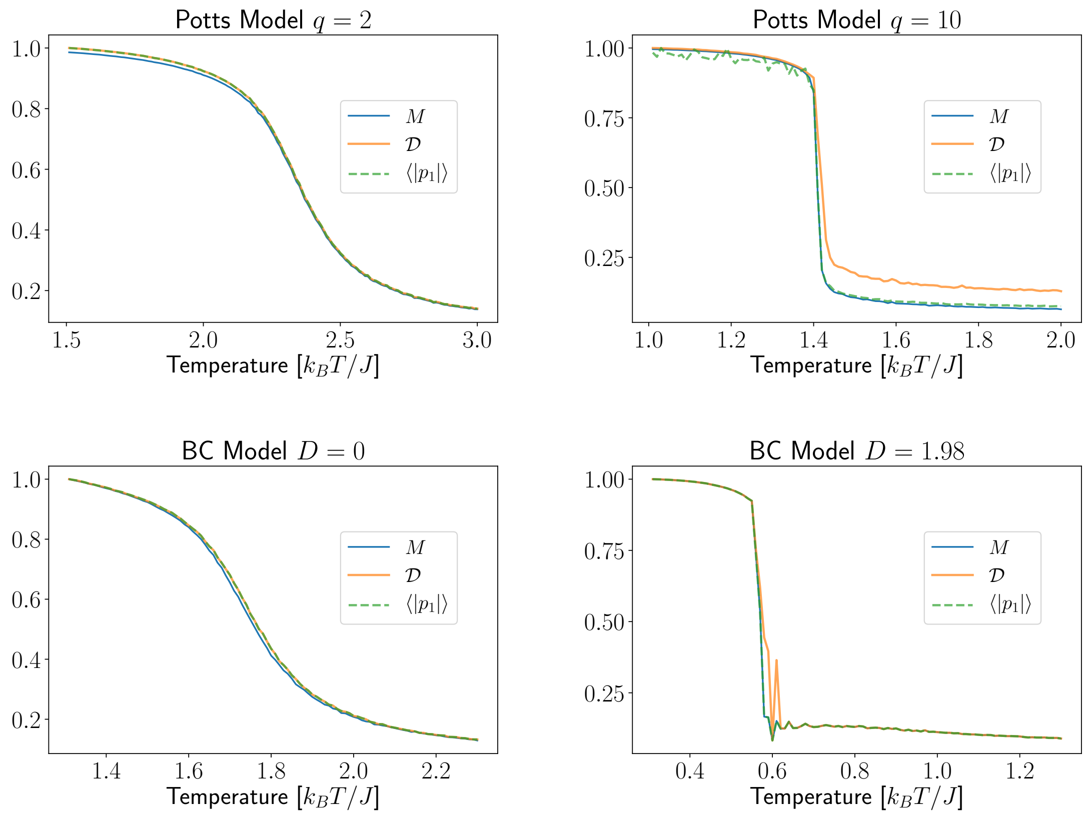


Fig. 4.14 The magnetization M in the function of temperature for 2–state and 10–state Potts model (upper panel) along with the distance between clusters $\mathcal{D}$ and mean absolute value of the PCA first component $\langle|p_1|\rangle$. The analogous plots are presented in the lower panel for the Blume Capel model with $D = 0$ and $D = 1.98$.

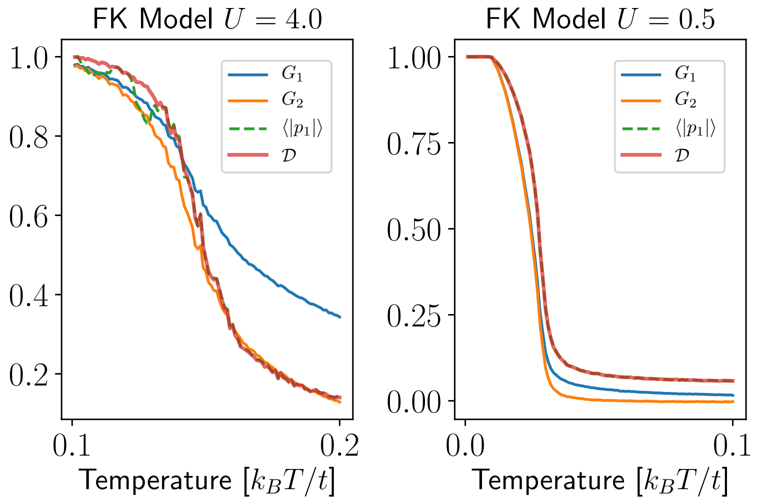


Fig. 4.15 The renormalized correlation functions G_1, G_2 in the function of temperature for the FK model $U = 0.5$ and $U = 4.0$ along with the distance between clusters $\mathcal{D}$ and mean absolute value of the first component of the PCA transformation $|\langle p_1\rangle|$.

4.2.2 Identification of the nature of a phase transition

Apart from the order parameter, another relevant information that can be obtained in the context of a phase transition is its nature. We generally distinguish two types of phase transitions: continuous and discontinuous (more information about the underlying differences between them can be found in Sec. 2.2). Our idea is to find out if the simple algorithm combing two basic unsupervised techniques introduced in this Section is able to, apart from determining the order parameter, also correctly identify the order of a phase transition. With this aim, we additionally study the shape of emerging clusters. Note that this idea is not entirely new. The evolution of the probability distribution of the order parameter and its various moments as a function of temperature for systems that undergo different types of phase transitions have already been widely studied, for instance, in [42, 122, 123, 124, 125, 126]. Furthermore, it has been shown in [127] that differences in the temperature evolution of the Potts ground states can be detected by a neural network and used to estimate the order of a phase transition.

To verify whether the form of emerging clusters can provide essential information about the nature of the undergoing phase transition, we analyze them at two levels of proximities, as suggested in [128]. These are:

- *dissimilarities* – measures how objects associated with different classes are far from each other,
- *similarities* – measures how objects belonging to the same class are close to each other.

The first step has already been taken. In the preceding Section, we introduced the quantity $\mathcal{D}$ (Eq. (4.12)) as the measure of *dissimilarity* between the ground states. As for the *similarity*, we take the silhouette coefficient [128, 129] as its criterion. It is defined in terms of 2 points illustrated in Fig. 4.16:

- **intra–cluster distance** a associated with the mean distance between a given sample and other samples assigned to the same class,
- **mean distance** b between a given sample and all other points in the next nearest cluster.

Then the silhouette coefficient s_i of a sample i reads:

$$s_i = \frac{b - a}{\max(a, b)}. \tag{4.9}$$

As can be inferred from Eq. (4.9), the value of s_i lies within the interval $\langle -1, 1 \rangle$. It has a straightforward interpretation. When b is much larger than a ($s_i \approx 1$), we observe a high level of separation between a sample i and the nearest cluster. On the other hand, when $b \approx a$ ($s_i \approx 0$), the groups of points are weakly separated and partially overlap.

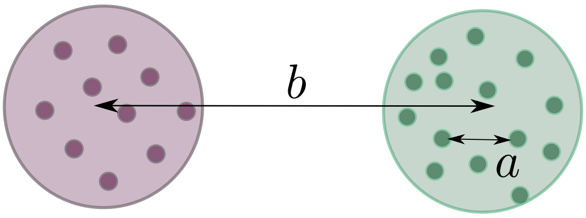


Fig. 4.16 Illustration of two scores a (**intra–cluster distance**) and b (**inter–cluster distance**) used in the expression for the silhouette coefficient formulated in Eq. (4.9).

In this case, we cannot unequivocally assign the sample i to one cluster or another. The worst-case scenario is when a is much larger than b ($s_i \approx -1$). It suggests that the sample is misclassified and should be assigned to another cluster.

In addition to the silhouette coefficient, numerous other methods exist evaluating the clustering performance, such as Calinski–Harabasz index [130] or V-measure [131]. However, we chose this measure because of its simplicity and clear interpretability.

To assess whether the mean value of s can provide some qualitative differences between systems undergoing first– and second–order phase transition, we calculate its dependence on temperature and the number of clusters k. Our findings are illustrated in Figs. 4.17, 4.18 and 4.19, where we present the mean value of the silhouette coefficient $s(k)$ as a function of the number of clusters k at three different moments of a phase transition: just before ($T < T_C$), during ($T \approx T_C$) and just right after ($T > T_C$). In all cases, we find the largest value of $(s_i)_{\text{max}} \approx 1$ for the $T < T_C$ and k corresponding to the number of ground states assigned to a given model. Since we observe well–separated clusters in this temperature region, this result is not so surprising. Then we discover that with an increase in temperature, the maximum value of $(s_i)_{\text{max}}$ decreases. At the same time, the distribution of s_i for different values of k becomes flatter. This means that the algorithm has difficulties in optimally dividing the data.

Moreover, one can notice that the distribution of $s(k)$ differs among models exhibiting first– and second–order phase transitions. We can interpret this observation as follows. When a system undergoes a second–order phase transition, it maintains the 'memory' about the number of the ground states even in a high–temperature phase. It makes the distributions of $s(k)$ similar in three analyzed temperature regimes. The situation is different for the first–order phase transition. In this case, in the vicinity but still in the low–temperature phase, the best separation is obtained when k is equal to the number of ground states. When $T \approx T_C$, the system falls almost immediately into a weakly separable, fully–disordered state – the 'memory' about the number of ground states is quickly lost, and the algorithm cannot select an optimal number of clusters to divide the data at $T > T_C$. To gain a more quantitative picture of the arisen $s(k)$ distributions presented in Figs. 4.17, 4.18, 4.19 but for a broader range of temperatures,

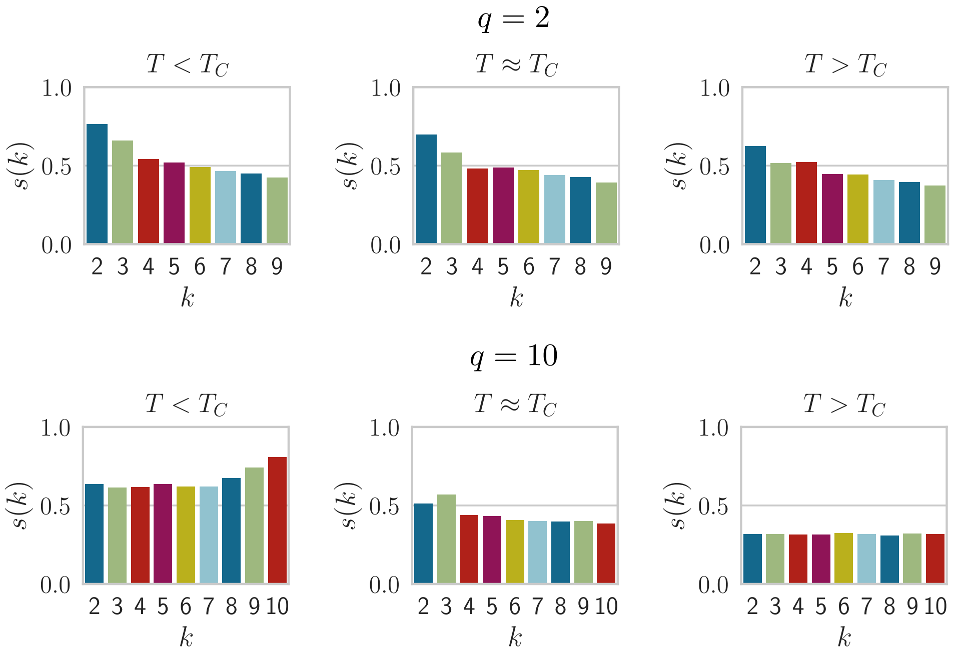


Fig. 4.17 The silhouette coefficient $s(k)$ in the function of the number of clusters k obtained for the Potts model exhibiting first– ($q = 10$, lower panel) and second–order phase transition ($q = 2$, upper panel) just right before ($T < T_C$), during ($T \approx T_C$) and just right after ($T < T_C$) phase transition.

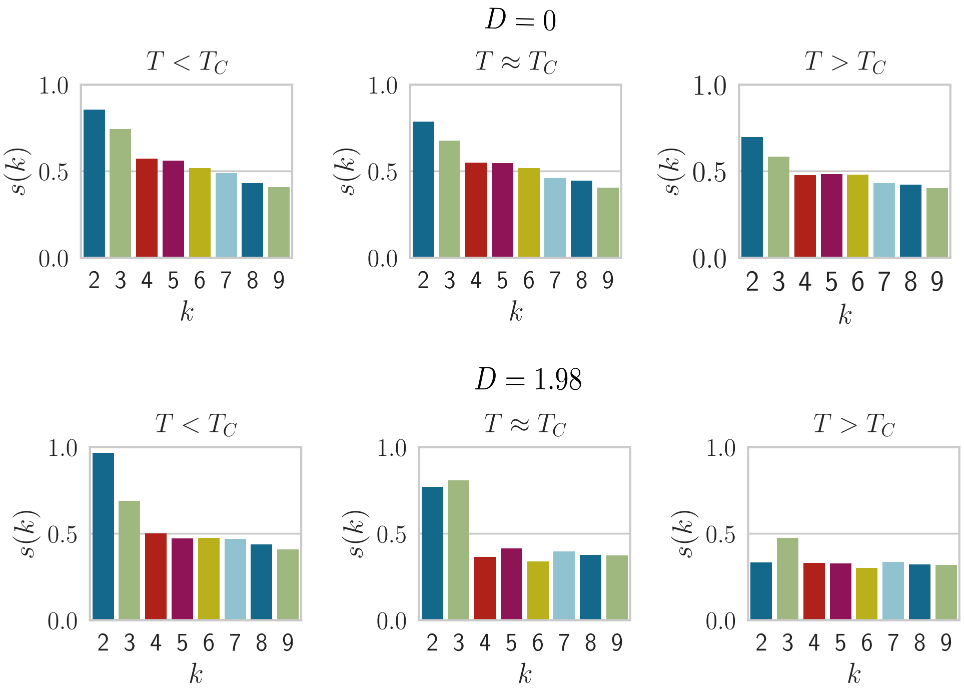


Fig. 4.18 The silhouette coefficient $s(k)$ in the function of the number of clusters k obtained for the Blume–Capel model exhibiting first– ($D = 1.98$, lower panel) and second–order phase transition ($D = 0$, upper panel) just right before ($T < T_C$), during ($T \approx T_C$) and just right after ($T < T_C$) phase transition.

we also plot the standard deviation of the silhouette coefficient $\sigma(s)$ defined as:

$$\sigma(s) = \sqrt{\frac{1}{N} \sum_{k=1}^{N} \left(s(k) - \overline{s(k)} \right)^2}, \tag{4.10}$$

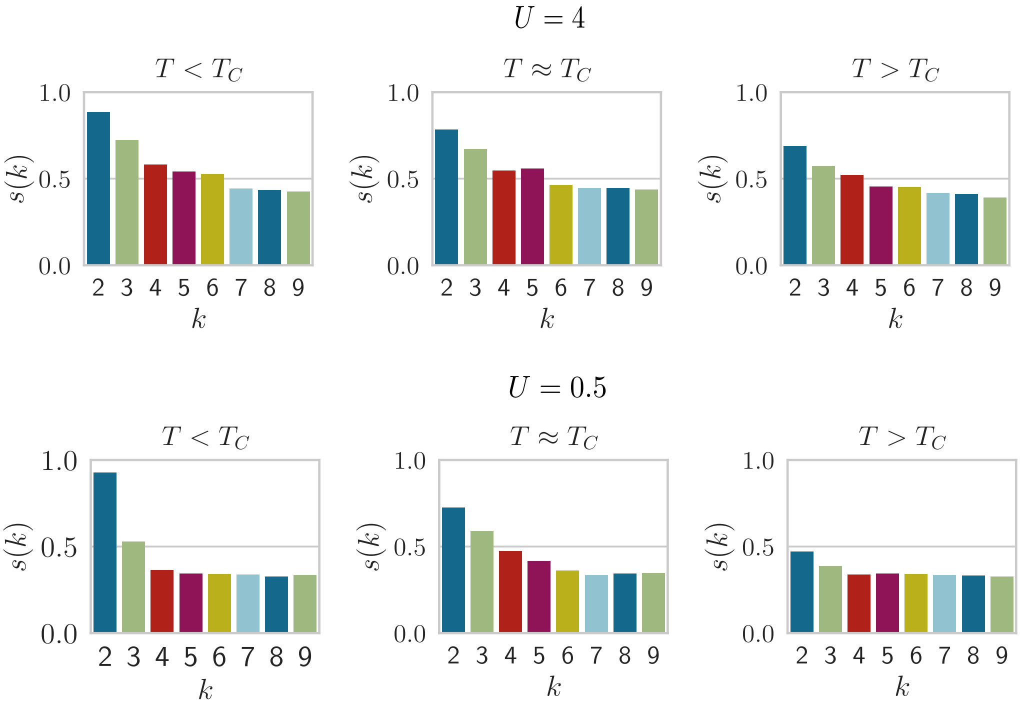


Fig. 4.19 The silhouette coefficient $s(k)$ in the function of the number of clusters k obtained for the Falicov–Kimball model exhibiting the first– ($U = 0.5$, lower panel) and second–order phase transition ($U = 4.0$, upper panel) just right before ($T < T_C$), during ($T \approx T_C$) and just right after ($T < T_C$) phase transition

.

where $N = 9$ is associated with the number of different values of $k = \{2, \ldots, 10\}$, $s(k)$ is the silhouette coefficient associated with k clusters and $\overline{s(k)}$ is the mean value of $s(k)$. These results are depicted in Fig. 4.20. It can be noticed that the criticality of the models undergoing first–order phase transitions is much better pronounced. In this case, the quantity $\sigma(s)$ abruptly drops to zero at the transition point. On the other hand, for systems undergoing second–order phase transition, to observe the uniform distribution of $s(k)$, we need to go far away above the critical point.

4.2.3 Summary

In this Section, we presented studies on emerging clusters obtained after PCA reduction of the raw Monte Carlo configurations (without applying any feature engineering) to 2 orthogonal dimensions. We found that in the low–temperature phase, the data split into several well–separated clusters. Its number corresponds to the number of ground states of a model analyzed. Near the critical point, the groups of points begin to merge. Then, for the temperatures $T > T_C$, the PCA representation of the MC configurations takes the form of one large, non–separable cloud. To quantitatively describe this evolution, we applied the k–Means algorithm and demonstrated that the distance $\mathcal{D}$ between the clusters correlates with the actual order parameter in most cases. Subsequently, we studied the PCA representation of the MC configurations in the context of the nature

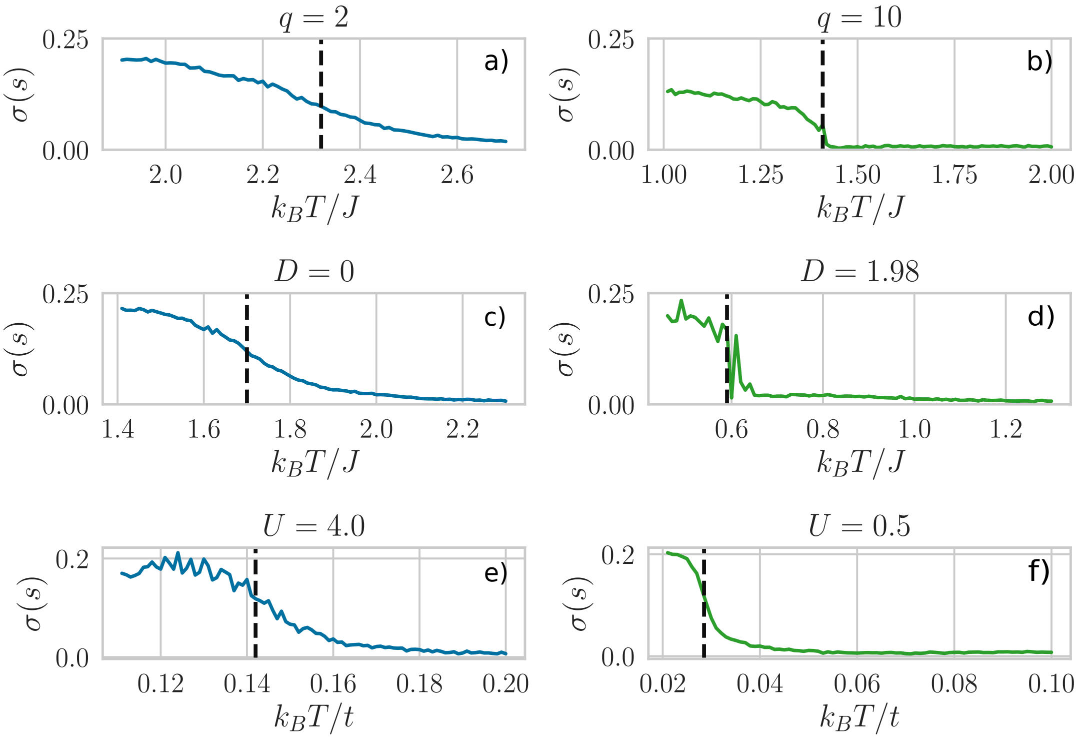


Fig. 4.20 The standard deviation $\sigma(s)$ of the silhouette coefficient s in the function of temperature for three different models undergoing second– (a), c), e)) and first– (b), d), f)) order phase transitions enumerated in Tab. 4.1.

of the phase transition. For three different models undergoing first– and second–order phase transitions, we analyzed the values of the silhouette coefficient $s(k)$ as a function of temperature and the number of clusters k. We discovered that the systems exhibiting first–order phase transitions 'forget' about the number of ground states right after the critical point. On the contrary, systems characterized by the second–order phase transition save this information even at temperatures $T \gg T_C$.

In closing, there are systems for which it is easy to identify the order parameter based solely on physical grounds, the symmetry of the analyzed model, and intuition. There exist, however, physical models for which this task is quite challenging. We hope that our findings will substantially facilitate determining this quantity in such cases. Furthermore, our results suggest that machine learning methods can also capture differences in the finite–size effects of systems undergoing first–order and second–order phase transitions. It means that they can possibly be applied as a tool for extracting information about the transition type from MC configurations.

4.3 A learning by confusion approach to study the nature of phase transitions

Determination of the order parameter is straightforward in the thermodynamic limit, i.e., when the volume of a system goes to infinity, $V \to \infty$. In the case of first–order phase transitions (*discontinuous*), we then observe a singular behavior of the free energy at the transition point. It leads, in turn, to discontinuities in thermodynamic quantities, such as the heat capacity C_v or magnetic susceptibility χ, and experimentally to the release of latent heat. Meanwhile, the correlation length remains finite. On the contrary, during second–order phase transitions (*continuous*), the correlation length diverges at the critical point, but the physical quantities pass smoothly through the transition. This different behavior of the physical observables at the critical point is associated with the phase coexistence occurring during first–order phase transitions (Sec. 2.2).

The above–described discrepancies are, unfortunately, not detectable in Monte Carlo (MC) simulations, which always deal with finite–size systems [132, 133, 134]. Instead of a sharp, well-defined transition, we observe a rounded peak in thermodynamic observables at the critical point in both types of phase transitions. Although the width of the emerging peak and its position scale differently with the lattice size L in these two cases, it is extremely difficult to establish the order of a phase transition unequivocally, based solely on the shape of this peak itself.

Fortunately, the phenomenological theory of the finite–size effects occurring in the temperature-driven first order–phase transition developed in [42, 135, 136, 137, 138] can give us some hint of whether we deal with the first– or second–order phase transition. This theory is based on the fact that the energies of the MC configurations generally obey the Gaussian probability distribution $P(E)$. The mean value of $P(E)$ is related to the infinite-lattice energy E_0, and the width C is proportional to the value of heat–capacity in the thermodynamic limit [42, 138, 139]. Then $P(E)$ can be expressed by the following formula:

$$P(E) = \frac{A}{\sqrt{C}} \exp\big[-\frac{(E-E_0)^2 \cdot L^d}{2k_B T^2 \cdot C}\big], \tag{4.11}$$

where A is a normalization constant, E is the energy, T stands for the temperature, and k_B denotes the Boltzmann constant. The remaining parameters in Eq. (4.11), i.e., L, d, stand for the linear size of the lattice and the dimension of a system, respectively.

However, such a form of $P(E)$ is not maintained in the vicinity of the first–order phase transition. The deviation from the standard energy distribution is attributed to phase coexistence and associated with it, the release of latent heat at the critical point. In this case, $P(E)$ is composed of two weighted Gaussians centered at internal energies

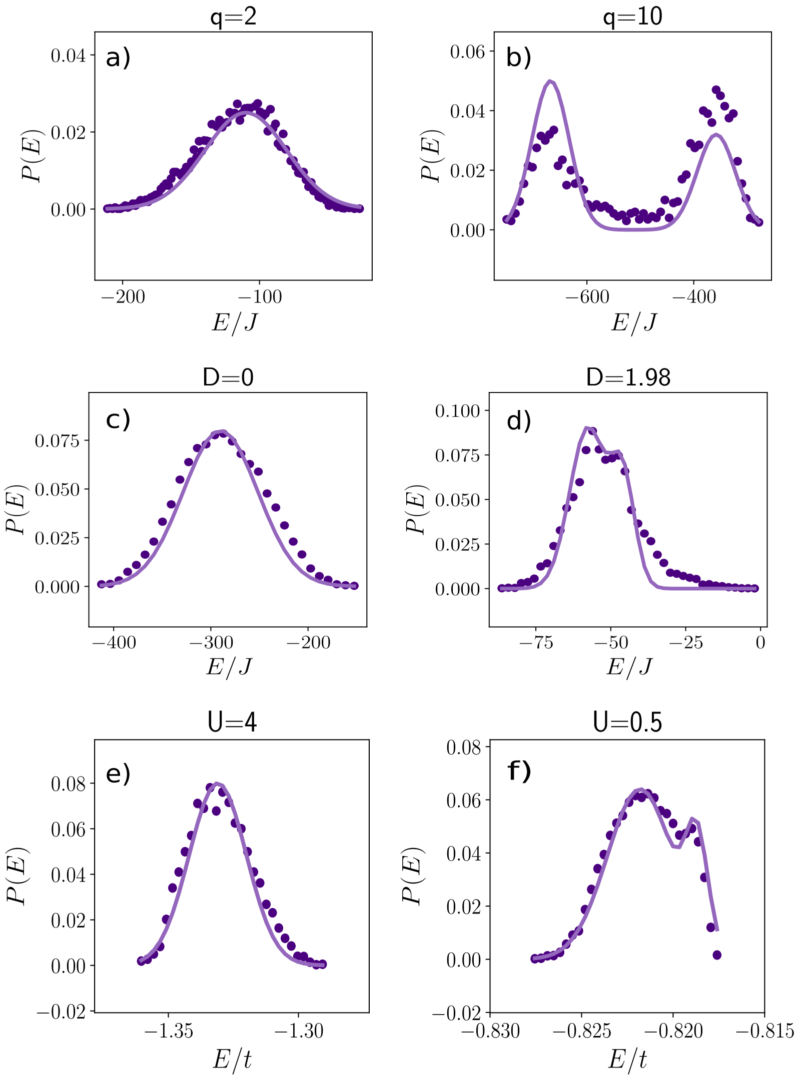


Fig. 4.21 The energy distributions $P(E)$ generated at the critical point for the models undergoing first (right panel) and second (left panel)–order phase transitions: a) The Potts model and the second–order phase transition; b) The Potts model and the first–order phase transition. c), d) same as a), b) respectively but for the Blume–Capel model, e),f) same as a), b) but for the Falicov–Kimball model. Points show MC data, lines are best fits of single/double Gaussian functions to the data. The linear size of the lattice is in all cases set to $L = 16$.

corresponding to coexisting low– (E_-) and high–temperature (E_+) phases:

$$P(E) = A\left[\frac{a_+}{\sqrt{C_+}}\exp\left[\frac{-(E-(E_+ + C_+\Delta T)}{2k_BT^2C_+}\right] + \frac{a_-}{\sqrt{C_-}}\exp\left[\frac{-(E-(E_- + C_-\Delta T)}{2k_BT^2C_-}\right]\right], \tag{4.12}$$

where A is a normalization constant, $C_{+,-}$ and $E_{+,-}$ are specific heats and energies attributed to two phases, $\Delta T = T - T_C$ denotes the deviation of temperature T from the critical point T_C and k_B stands for the Boltzmann constant. The constants $a_{+,-}$ are complicated functions of specific heats $C_{+,-}$, temperatures T and number of coexisting states. Their explicit forms can be found in [42].

In Fig. 4.21, we present the probability distributions $P(E)$ for the Potts, Blume-

Capel and Falicov-Kimball models exhibiting first– and second–order phase transitions described in more detail in Sec. 2.5. It is apparent that the shapes of these distributions are in agreement with the theory mentioned above. We observe a two–peak structure of $P(E)$ in the vicinity of the first–order transition (right column in Fig. 4.21) and only one peak when the transition is of the second order (left column in Fig. 4.21).

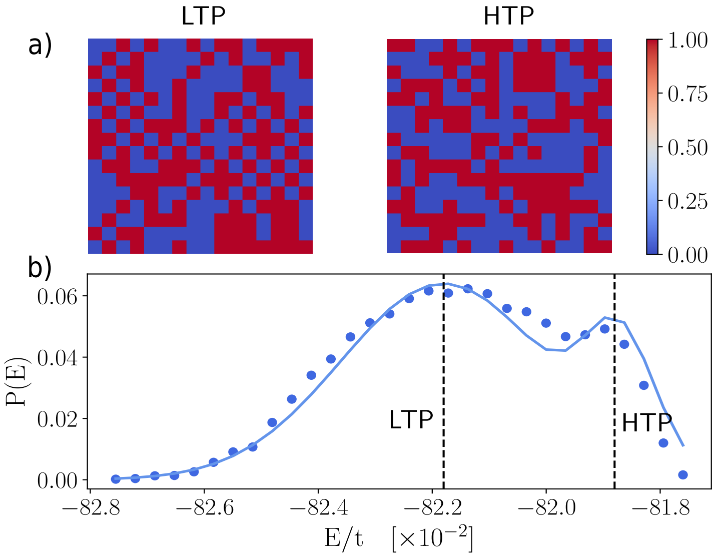


Fig. 4.22 First–order phase transition in the Falicov–Kimball model: a) example snapshots of lattice configurations taken at the phase coexistence point. Left and right panels correspond to the low– (LTP) and high–temperature (HTP) phases respectively; b) probability distribution of energy $P(E)$ at the transition temperature. Points represent MC data, blue line is the best fit to the data. The vertical dashed lines mark the configurations' energies presented in the upper image. The parameters are $T = 0.028\ k_B/t$, $U = 0.5t$.

Additionally, in Figs. 4.22, 4.23 and 4.24, we illustrate two different states generated in $T \approx T_C$ for models undergoing first–order phase transitions. In all cases, one exhibits more disorder than the other, suggesting the coexistence of two distinct phases near the criticality.

Although widely accepted, the Gaussian method suffers from certain limitations. The decision about the order of a phase transition is based solely on qualitative results. It may also require MC simulations for bigger systems when the assessment of the order for smaller sizes is not conclusive (which often happens when we have to discriminate between weak first–– and second–order phase transitions) [127]. To overcome, at least partially, these problems, apart from studying the shape of $P(E)$ at the transition, it is instructive to consider the Binder–Challa cumulant V_L [42, 135, 136, 137] defined as a function of the moments of energy distributions:

$$V_L = 1 - \frac{\langle E^4 \rangle}{3\langle E^2 \rangle^2}. \tag{4.13}$$

It turns out that V_L exhibits quite different behavior during two kinds of phase

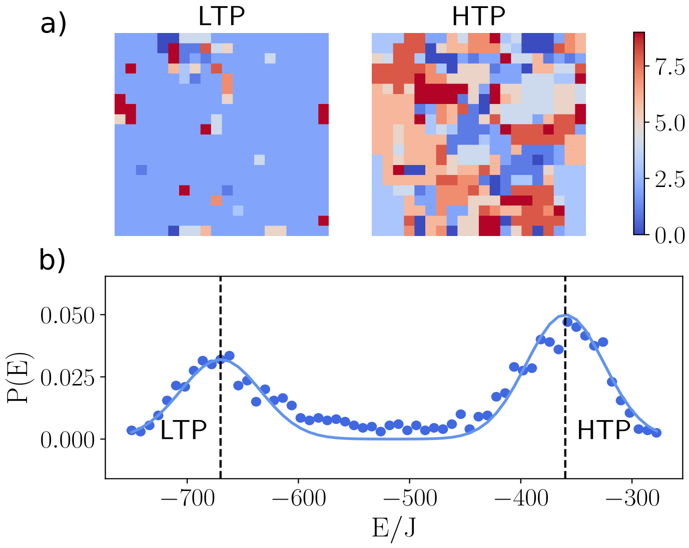


Fig. 4.23 The same as in Fig. 2.10 but for the Potts model. The parameters are $T = 1.4\ k_B/J$, $q = 10$.

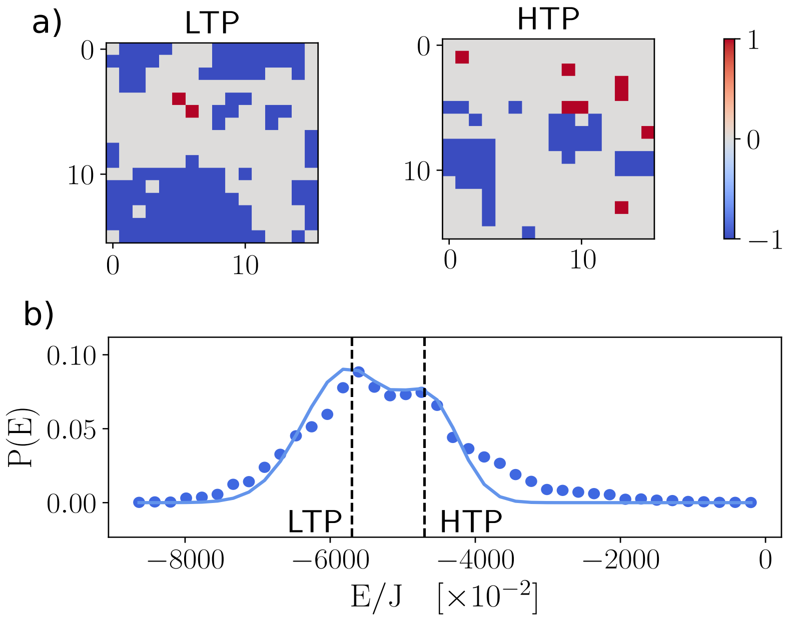


Fig. 4.24 The same as in Fig. 2.10 but for the Blume–Capel model. The parameters are $T = 0.58\ k_B/J$, $D = 1.98$.

transitions. That is, when $P(E)$ is described by a single Gaussian then (Eq. (4.11)) $V_L \to \frac{2}{3}$ in the thermodynamic limit [42]. It happens for systems undergoing the second–order phase transition in the whole temperature range. The situation is slightly different during first–order transitions for which the double peak structure of $P(E)$ appearing near the criticality leads to a non–trivial limit of the cumulant at $T = T_C$:

$$\left(V_L\right)_{min} = 1 - \frac{2(E_+^4 - E_-^4)}{3(E_+^2 - E_-^2)}, \quad L \to \infty. \tag{4.14}$$

Eq. (4.14) suggests that when a system undergoes a first–order phase transition, we

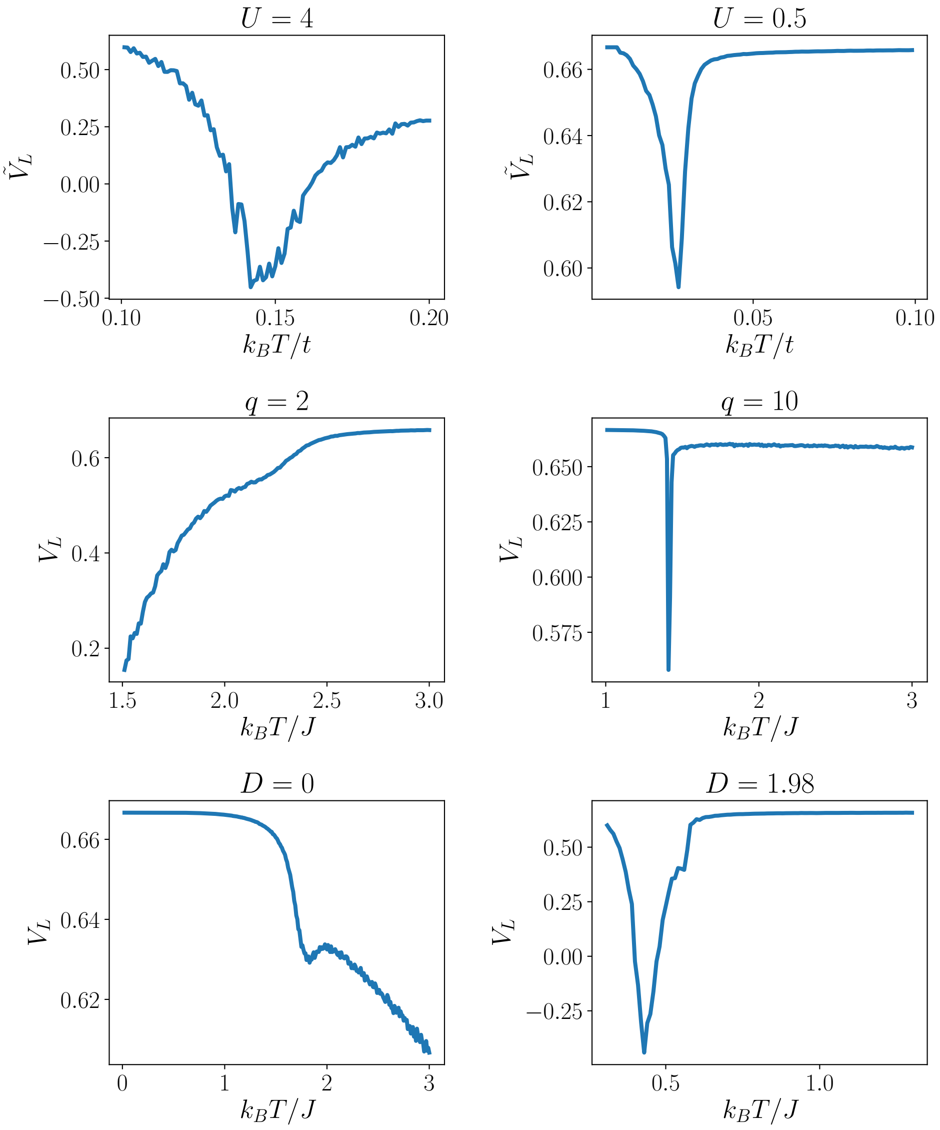


Fig. 4.25 The Binder–Challa cumulant V_L (Eq. 4.13) as function of temperature T for FK, Potts and BC models in the regime of second–order (**left panel** of the Figure) and first–order phase transitions (**right–panel** of the Figure). In the case of FK model the values of V_L are of order 10^{-6}, hence we define $\tilde{V}_L = (V_L - 0.6666) \times 10^4$.

should observe a clear minimum in V_L at the critical point. As presented in Fig.4.25, a well pronounced minimum is observed for the qP and BC models. For the FK model the distinction between two types of phase transition based on the shape of V_L is impossible due to a very small amplitude of V_L and the lack of qualitative differences in V_L for continuous and discontinuous transitions. In both types of phase transitions, the cumulant V_L changes very slowly with temperature – its relative change is of the order 10^{-4} for a continuous phase transition and 10^{-6} for a discontinuous phase transition. Such an unusual behavior of V_L may be attributed to the fact that while qP and BC models are purely classical, the FK Hamiltonian predicts the existence of two subsystems involving localized, classical particles (ions) and itinerant quantum fermions (electrons). It has been demonstrated in Ref. [41] that both subsystems exhibit the same value of critical temperature (based on the positions of the peaks of the charge susceptibility CDW for ions and the heat capacity for electrons). Nevertheless, there exist two different energy scales in this model. The first one is associated with the indirect interaction between ions mediated by the electrons. The second one, defined by the Fermi energy, significantly exceeds the value of $k_B T_C$. Since the quantity V_L is calculated from the fluctuations of the fermion energy, they affect the phase transition only to a small extent. It is apparent in the temperature dependence of the cumulant.

The cumulant method has some limitations also for classical models. The distinction between first– and second–order phase transitions can be challenging for models at the edges of parameters separating the nature of phase transition. Perhaps the best example illustrating this situation is the determination of the phase transition order for 4–state and 5–state Potts models [127]. In this case, MC simulations performed for large lattice sizes are required to determine the order of the transition. Such calculations, in turn, are time–consuming and computationally expensive. The situation is even worse for systems with quantum degrees of freedom, which require exact diagonalization of the Hamiltonian in each MC step. It is therefore crucial to find alternative methods to identify the nature of the phase transition. One of the potential new techniques, based on the *learning by confusion* scheme is presented in the next paragraph.

The learning by confusion (LBC) method is an innovative machine learning algorithm combining supervised and unsupervised learning [110]. It was designed to detect phase transitions in various models of condensed matter physics. It has been demonstrated that the algorithm copes well with complex networks [140], classical spins, and frustrated magnetic models [141]. Furthermore, it is capable of correctly identify the nuclear gas–liquid [142], double, topological [143] quantum phase transitions [110, 144, 145], those that occur in polaritons [146] and ferrimagnetic alloys [147].

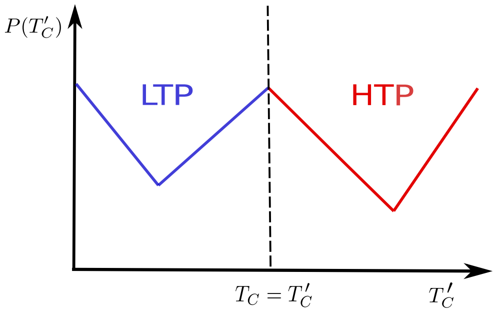


Fig. 4.26 Typical shape of the performance of the neural network $P(T'_C)$ in the learning by confusion method plotted versus the alleged critical temperature T'_C. The middle peak in the characteristic W–shape of $P(T'_C)$ indicates the critical temperature T_C predicted by the neural network.

The method consists in feeding the Monte Carlo configurations into a neural network labeled according to the proposed value of critical temperature $T'_C \in [T_{min}, T_{max}]$. The algorithm is then trained on such provided data, and its performance $P(T'_C)$ on a test set is evaluated. This procedure is repeated for different values of the alleged critical temperature T'_C ranging from T_{min} to T_{max}. Ultimately, we get the overall function of $P(T'_C)$ on $T'_C \in [T_{min}, T_{max}]$. As illustrated in Fig. 4.26 it takes the form of W–shape. We can explain such a form of $P(T'_C)$ in the following manner. When the proposed temperature $T'_C = T_{min}$ or $T'_C = T_{max}$, then we assume that all samples represent either high–(HTP) or low–temperature phase (LTP), respectively. The neural network (NN) learns then trivially the identity function, and its performance is equal to 100%. In contrast, when $T'_C < T_C$ or $T'_C > T_C$ some configurations are mislabelled (Fig. 4.27), i.e., states that belong to the same phase are labeled differently. The algorithm cannot capture the differences between configurations representing the same phase, resulting in a lower prediction accuracy. We then say that the NN is ”confused” (hence the algorithm naming). When T'_C is equal to the actual critical temperature, then the ”confusion” disappears, and the performance of NN is maximal. Therefore, T'_C returning the maximum value of $P(T'_C)$ is associated with the critical point identified by the algorithm.

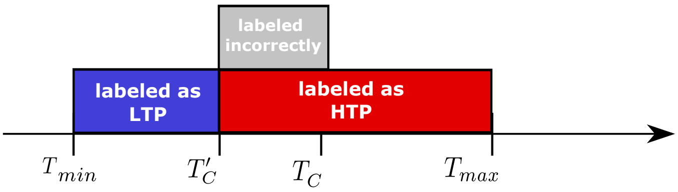


Fig. 4.27 The procedure of labeling configurations within the confusion scheme. T_{min} and T_{max} are the minimum and maximum temperatures taken into account as potential candidates for the critical temperature. T_C and T'_C stand for the actual and proposed values of the critical point, respectively. It implies that all configurations generated at $T \in (T'_C, T_C)$ are mislabeled.

4.3.1 Result of application of the LBC scheme – theoretical analysis

In the following Section we will construct a simple intuitive model of the neural network performance for the LBC method. The model is based on the statistical properties of the continuous and discontinuous phase transitions discussed above. Next, we will juxtapose the theoretical predictions with the numerical results obtained for the discussed models.

4.3.1.1 Continuous phase transitions

When a system undergoes a continuous phase transition, we do not observe phase coexistence at the critical point. Then the fraction α of the configurations corresponding to the low–temperature phase can be expressed as a function of temperature:

$$\alpha(T) = \begin{cases} 1 & \text{for } T < T_C, \\ 0 & \text{otherwise.} \end{cases} \tag{4.15}$$

In such a case, the number of mislabeled configurations increases linearly with the interval $|T_C - T'_C|$, where T_C and T'_C denote the proposed and actual value of critical temperature, respectively. The situation is illustrated in Fig.4.28.

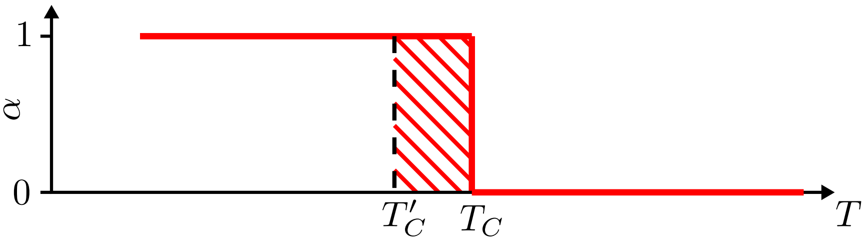


Fig. 4.28 The fraction α of low–temperature phase as the function of temperature T. T_C stands for the actual value of critical temperature (i.e., obtained, for instance, from the MC simulations). The red dashed lines indicate the interval of mislabeled temperatures.

To investigate the influence of $\alpha(T)$ on neural network performance $P(T'_C, T_C)$, we introduce a measure $\mathcal{M}$ of the number of mislabeled data as a function of T'_C:

$$\mathcal{M}(T'_C) = \int_{T_{min}}^{T_{max}} f(T'_C, T) dT, \tag{4.16}$$

where the function $f(T'_C, T)$ returns the fraction of incorrectly labeled data points corresponding to the temperature T'_C, while T_{min} and T_{max} denote the minimum and maximum temperature of the MC configurations taken into account during the analysis. From Fig. 4.28 we conclude that:

$$f(T'_C, T) = \begin{cases} 1 & \text{for } T'_C < T < T_C \text{ or } T_C < T < T'_C, \\ 0 & \text{otherwise.} \end{cases} \tag{4.17}$$

By inserting Eq. (4.17) into Eq. (4.16) we obtain $\mathcal{M}(T'_C) = |T'_C - T_C|$. Assuming that the performance of a neural network $P(T'_C)$ can be, in the vicinity of the phase transition, expressed as $P(T'_C) \approx 1 - \frac{\mathcal{M}(T'_C)}{T_{max} - T_{min}}$ we conclude that in the case of continuous phase transition, $P(T'_C)$ decreases linearly with $|T'_C - T_C|$.

4.3.1.2 Discontinuous phase transitions

The characteristic feature of discontinuous phase transitions is the phenomenon of phase coexistence arising in the vicinity of the critical point. The coexistence of low-

and high-temperature phases requires a modification of the model since fractions of both phases can exist below and above T_C. We introduce two different temperatures:

- T_1 - the temperature below which all configurations belong to the low–temperature phase,
- T_2 - the temperature above which all configurations belong to the high–temperature phase.

The interval $T \in (T_1, T_2)$ corresponds to the area of phase coexistence, with the fraction of the high–temperature phase increasing linearly with T. Taking these assumptions into account $\alpha(T)$ can be expressed as follows:

$$\alpha(T) = \begin{cases} 1 & \text{for } T \leq T_1, \\ \dfrac{T_2 - T}{T_2 - T_1} & \text{for } T_1 < T < T_2, \\ 0 & \text{for } T \geq T_2. \end{cases} \tag{4.18}$$

Note that the linearity assumed in Eq. (4.18) is well justified. In Fig. 4.29, we present the evolution of the energy distribution $P(E)$ generated in the temperatures $T \approx T_C$ for a model undergoing discontinuous phase transition on the example of 10–state Potts model. We can conclude that the heights h of two Gaussians corresponding to coexisting phases (E_- and E_+) vary linearly with temperature T. This dependence is shown in Fig. 4.30, where the crossing of fitted linear functions signals the critical temperature T_C.

From Eq. (4.18), one can infer that the fraction of incorrectly annotated data should be nonzero even when $T'_C = T_C$. Illustration of the number of mislabeled data points for three different values of T'_C is presented in Fig. 4.31. Here, just as in the case of a continuous phase transition, $\mathcal{M}(T'_C)$ increases linearly with the distance $|T_C - T'_C|$ for $T'_C < T_1$ and $T'_C > T_2$. This dependence is different for temperatures $T'_C \in (T_1, T_2)$. However, calculating the accurate expression for $\mathcal{M}(T'_C)$ requires careful consideration of three different cases illustrated in Fig. 4.31. For the sake of clarity, the derivation of $\mathcal{M}(T'_C)$ is presented in Appendix B. Here we present the main results of these calculations:

$$\mathcal{M}(T'_C) = \begin{cases} \frac{1}{\Delta}(T'_C - T_C)^2 + \frac{\Delta}{4} & \text{for } |T'_C - T_C| < \frac{1}{2}\Delta, \\ |T'_C - T_C| & \text{for } |T'_C - T_C| > \frac{1}{2}\Delta, \end{cases} \tag{4.19}$$

where $\Delta = T_2 - T_1$.

It is necessary to emphasize the limitations of this simple theoretical model in the description of both types of phase transitions. In the case of continuous phase transition, it applies solely to the system in the thermodynamic limit. In this situation,

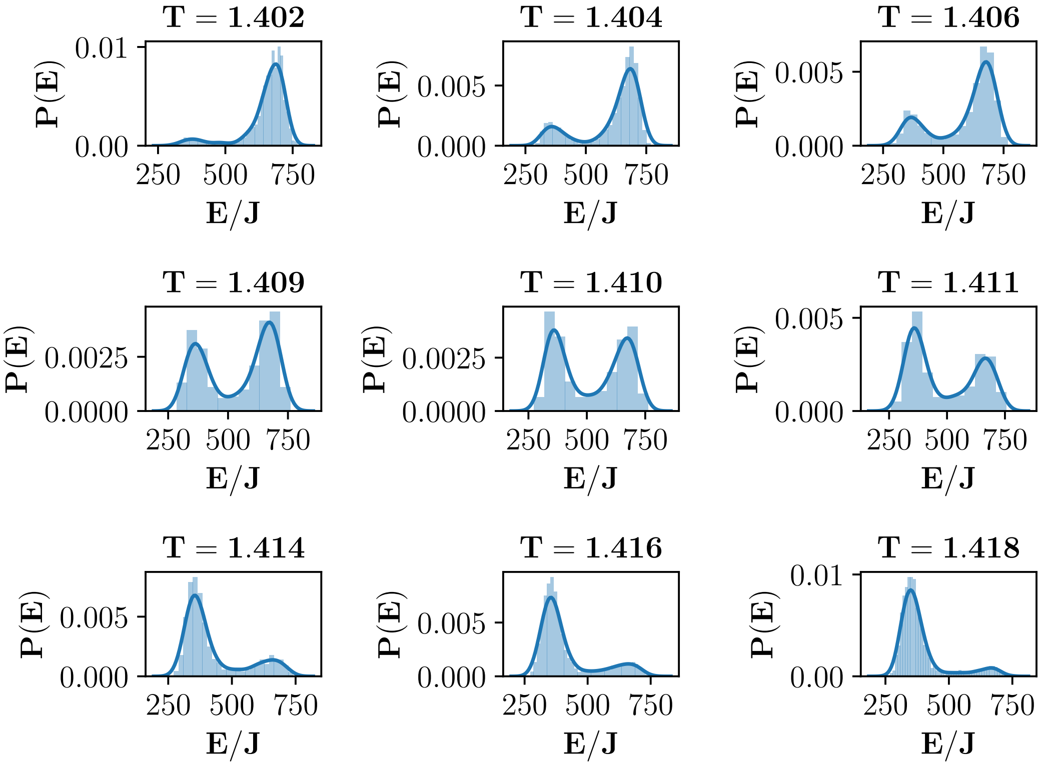


Fig. 4.29 The energy distribution $P(E)$ of MC configurations generated in the temperatures $T \approx T_C$ for the 10–state Potts model.

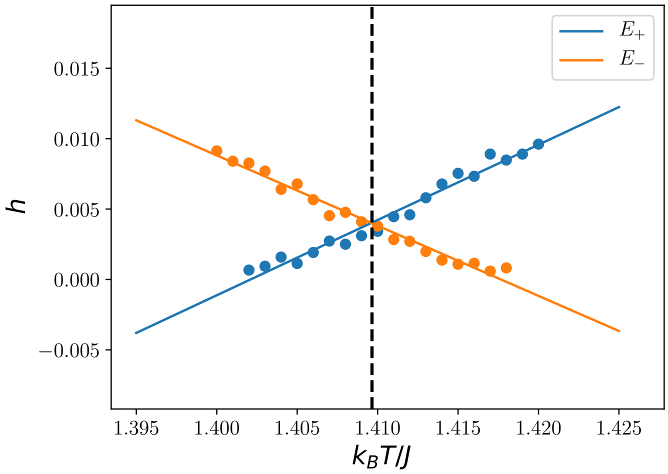


Fig. 4.30 Heights h of the Gaussians emerging from energy distributions $P(E)$ in the function of temperature calculated for the 10–state Potts model. Two solid lines represent the linear fits to the peaks of $P(E)$ corresponding to the low– (E_-) and high–temperature (E_+) phases, respectively.

we do not have to deal with the finite–size effects, which can alter the ultimate neural network performance. The studies presented in Refs. [110, 143, 147, 148] show that

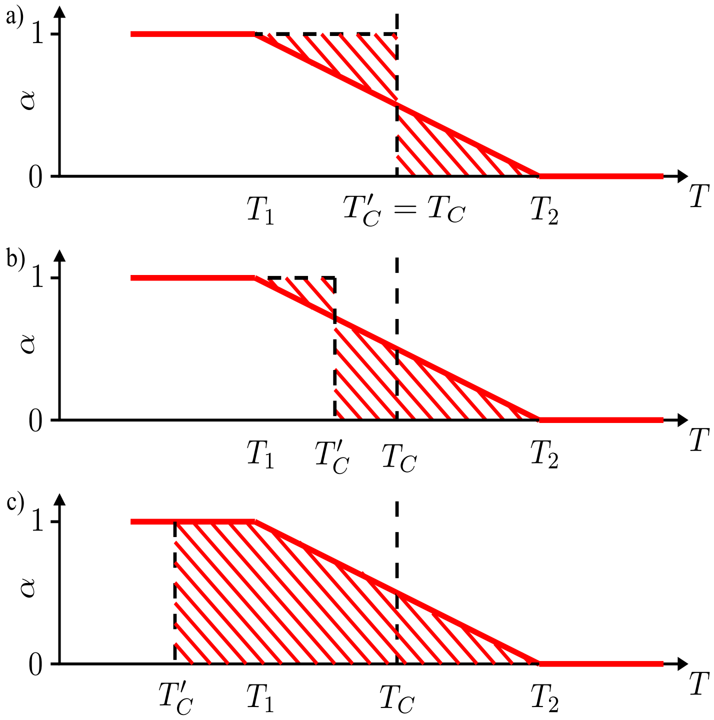


Fig. 4.31 The fraction α of low–temperature phase as function of temperature T. T_C stands for the true critical temperature, while T'_C denotes the assumed one. T_1 and T_2 are boundary temperatures between which two phases coexist. The dashed lines in figure indicate the amount of mislabeled data. The scheme is presented in three different situations: **a)** $T_C = T'_C$, **b)** $T_1 < T'_C < T_C$, **c)** $T_{C'} < T_1$

finite–size effects may be responsible for a decrease in accuracy at the critical point, i.e., $P(T'_C = T_C) < 1$. On the other hand, description of the first–order phase transition refers only to finite–size systems. When $L \to \infty$, the phase coexistence is not observed, and two temperatures T_1 and T_2 in Eq. (4.18) converge to one. Nevertheless, the developed model does give us some idea of what we can expect from numerical calculations. The performance of a neural network predicted by this model for both types of phase transitions is shown in Fig. 4.32. The model gives a well pronounced maximum of $P(T'_C)$ for the continuous phase transition, while for the discontinuous one should expect a rather gentle peak of $P(T'_C)$ around T_C.

4.3.2 Result of application of the LBC scheme – numerical analysis

In this Section, we juxtapose the numerical results obtained with those predicted with the simple toy model developed in Sec. 4.3.1.1, 4.3.1.2. The numerical part of our study involves the application of the LBC scheme to the Falicov–Kimball (FK), q–state Potts (qP), and Blume–Capel (BC) models, which were already analyzed with unsupervised learning methods (Sec.4.2).

It is important to emphasize that, in contrast to the assumptions of the theoretical model, our numerical analysis postulates the usage of different performance metrics.

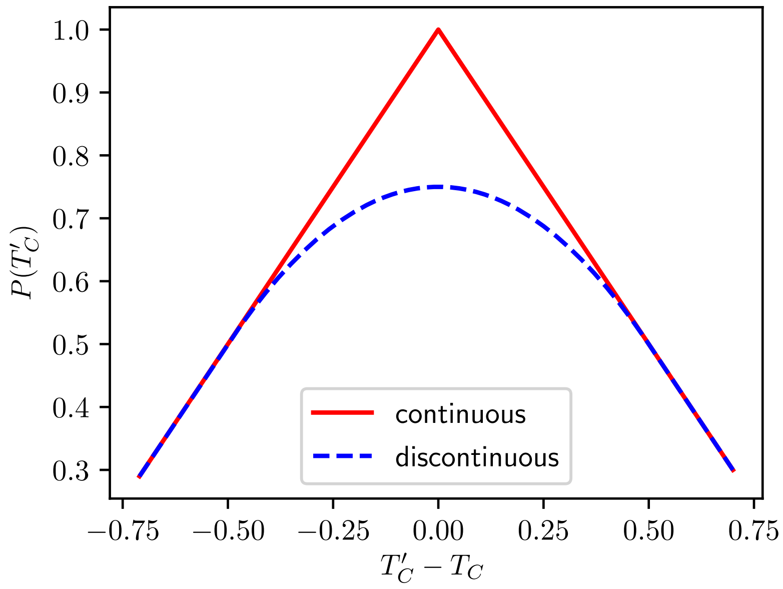


Fig. 4.32 The performance of a neural network $P(T'_C) = 1 - \frac{\mathcal{M}(T'_C)}{\delta T}$ as the function of distance $|T'_C - T_C|$ predicted by the theoretical toy model in the case of continuous and discontinuous phase transition.

Here, instead of the more popular *accuracy*, we use the AUC–ROC metrics defined in Sect. 3.10. The reasons behind such a choice, along with the computational details, can be found in Appendix C.

As the main result of the studies presented in this Section, we obtain the AUC–ROC curves as a function of the proposed critical temperature T'_C. The results for all models are illustrated in Fig. 4.33. For models exhibiting a second–order phase transition (left panel of Fig. 4.33), we obtain a clear maximum that indicates the critical temperature T'_C. As can be seen, T_C obtained from LBC scheme coincides with T_C identified with MC simulations. The situation is more complicated when the transition is discontinuous (right panel of Fig. 4.33). In this case, the shape of the AUC–ROC curve strongly depends on the model under study. We find that for qP and BC models the maximum is readily detectable in the position we expect from the numerical computations. It is, however, a little less pronounced compared to the continuous phase transition. Surprisingly, we discover strikingly different behavior for the FK model (Fig. 4.33b). In this case, the AUC–ROC curve takes the shape of a plateau spreading over a wide temperature range just below T_C. In this temperature region, the neural network 'overfits' gaining the capability to distinguish between configurations belonging to the same, low–temperature phase.

To explain this unusual characteristic, we study low–temperature MC samples simulating the FK model for discrepancies that impact the high accuracy level. In this aim, for all temperatures, we calculate the mean value of the defect concentration $\mathcal{C}(T)$. By a 'defect', we consider any 2×2 window deviating from the checkerboard pattern representing the ground state of the model (Fig. 2.10). Three kinds of such

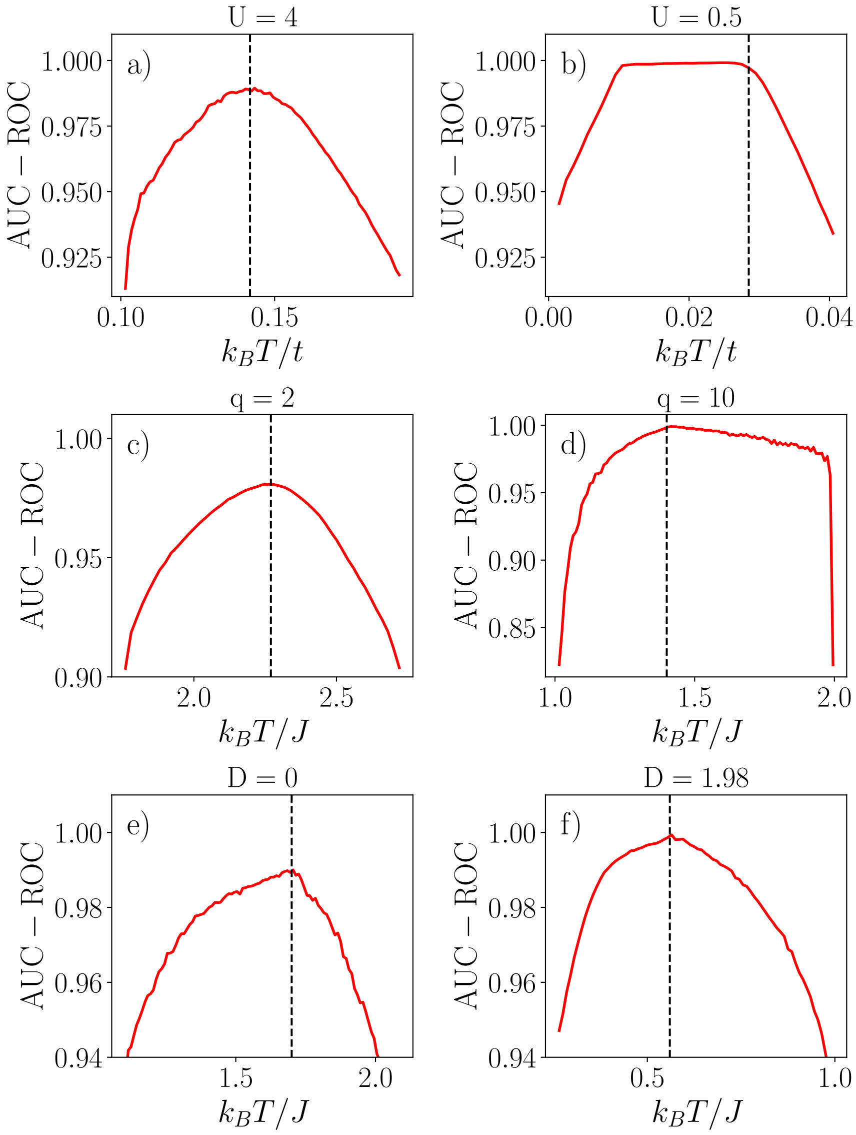


Fig. 4.33 The AUC–ROC metrics as function of temperature for a), b) FK, c), d) qP and e), f) BC models in different parameter regimes enumerated in Tab. 4.1 obtained as a result of application of the LBC scheme. The dashed vertical line indicates the critical temperature determined from MC simulations.

’defects’ (up to the symmetry transformations) are presented in Fig. 4.34. The results of $\mathcal{C}(T)$ calculations performed for four FK models characterized by different values of potentials U are illustrated in Fig. 4.35. We find that in all cases $\mathcal{C}(T)$ grows with temperature. The reason for that is thermal fluctuations that disturb the ideal order: the configurations depart more and more from the checkerboard pattern, which is entirely lost above the critical point. In the case of the FK model $U = 0.5$ (Fig.4.35a)), almost all configurations generated below $T < 0.01$ k_B/t exhibit complete ordering. In the temperatures $T \in [0.01, 0.028]$ k_B/t, ’defects’ proliferate rapidly with the temperature. It implies that the neural network (NN) can learn differences in concentrations and, as a result, discriminate between MC configurations assigned to the same low–temperature phase. It, in turn, leads to the development of the vast plateau in the AUC-ROC function in this temperature range. The situation is similar to the FK model $U = 1$ (Fig.4.35b)). For the higher potential values, i.e., $U = 2.0$ (Fig.4.35c)) and $U = 4.0$ (Fig.4.35d)), we observe clear maxima in neural network performance. Simultaneously, the number of 'defects' and its growth rate are lower compared to the models undergoing first–order phase transition. It makes discrimination between configurations corresponding to the same phase more challenging than in the previous case.

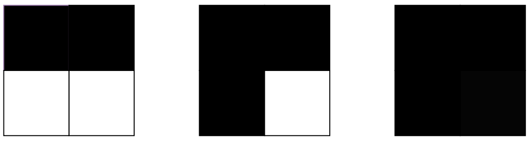


Fig. 4.34 Three different kinds of defects occurring in the FK configurations defined as deviations from the checkerboard pattern. The black and white cells indicate either the presence or absence of an ion at the lattice site.

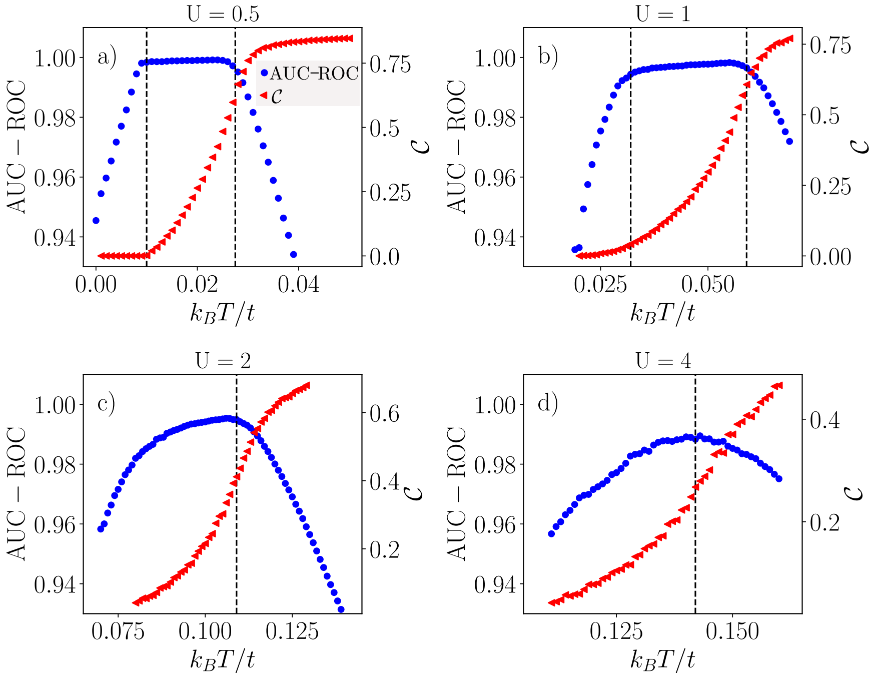


Fig. 4.35 The AUC–ROC (blue dots) metrics juxtaposed with the concentration of defects $\mathcal{C}$ (red dots) as function of temperature obtained for the FK model for different values of interaction U.

To further substantiate our conjecture, we additionally depict Gassians representing distributions of the defects concentration $P(\mathcal{C})$ for the FK model exhibiting second–order (left column of Fig. 4.36) and first–order phase transitions (right column of Fig. 4.36). When the transition is continuous, Gaussians representing three close temperatures almost completely overlap, regardless of whether they correspond to the low– or high–temperature phase. It makes the difference in the concentration of defects $\mathcal{C}$ for two close temperatures marginal and, as a consequence, leads to the inability

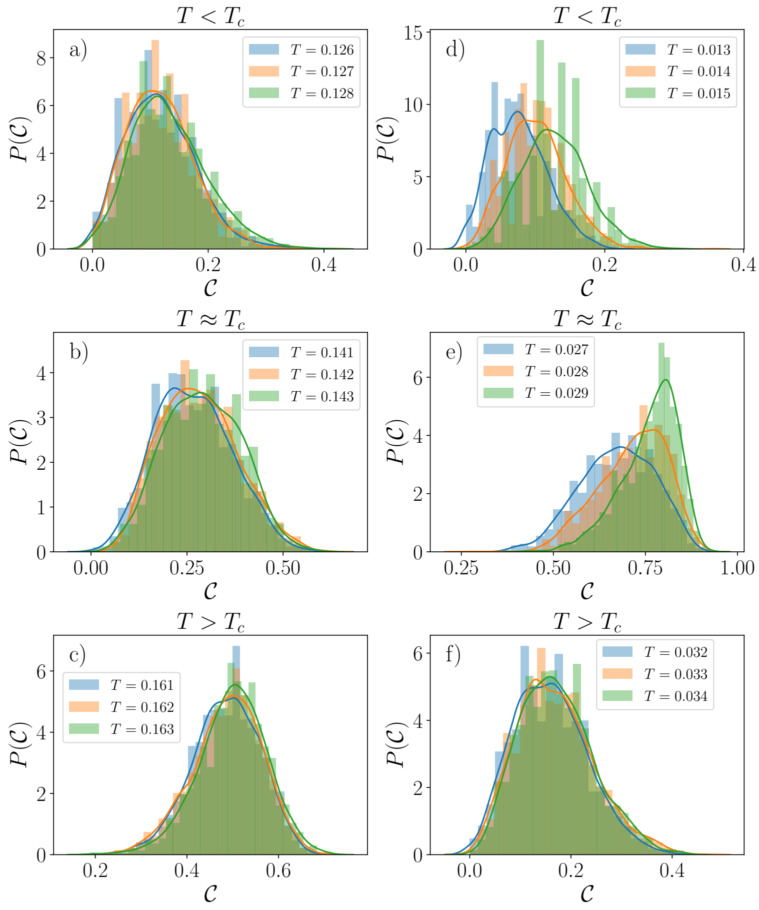


Fig. 4.36 The probability distribution of the 'defects' concentration $P(\mathcal{C})$ for different temperatures. Left column refers to the continuous phase transition (FK model $U = 4$), while the right one represents the discontinuous phase transition (FK model $U = 0.5$).

of the NN to discriminate between configurations. The situation is different when the transition is first–order. In this case, the distributions $P(\mathcal{C})$ overlap only partially for three close temperatures in the plateau region (Fig. 4.36d) and the vicinity of the critical point (Fig. 4.36e). The neural network recognizes these discrepancies, which results in high classification performance.

In order to verify whether the neural network can discriminate between configurations belonging to the same phase we perform a simple theoretical experiment. We take MC samples at two different temperatures, T_1 and T_2 corresponding to the same phase in three various combinations: T_1 and $T_2 < T_C$, T_1 and $T_2 \approx T_C$, T_1 and $T_2 > T_C$, labeled as they would belong to two diverse classes. Subsequently, we feed them into a neural network and repeat this procedure for different values of $\Delta T = |T_2 - T_1|$. The results of this analysis are illustrated in Fig. 4.37. Based on this picture, we can conclude that when a system undergoes a second–order phase transition, the classification abilities of NN are similar in the entire temperature range (Fig. 4.37a)-c)). The situation is more interesting for the first–order phase transition. In the plateau region, Fig. 4.37d)), we achieve almost perfect accuracy, in line with the previous results. This performance decreases near the critical point (Fig. 4.37e)). The neural network loses all its classification power in the high–temperature phase ((Fig. 4.37 f)), which results resulting in the abrupt drop of AUC–ROC function.

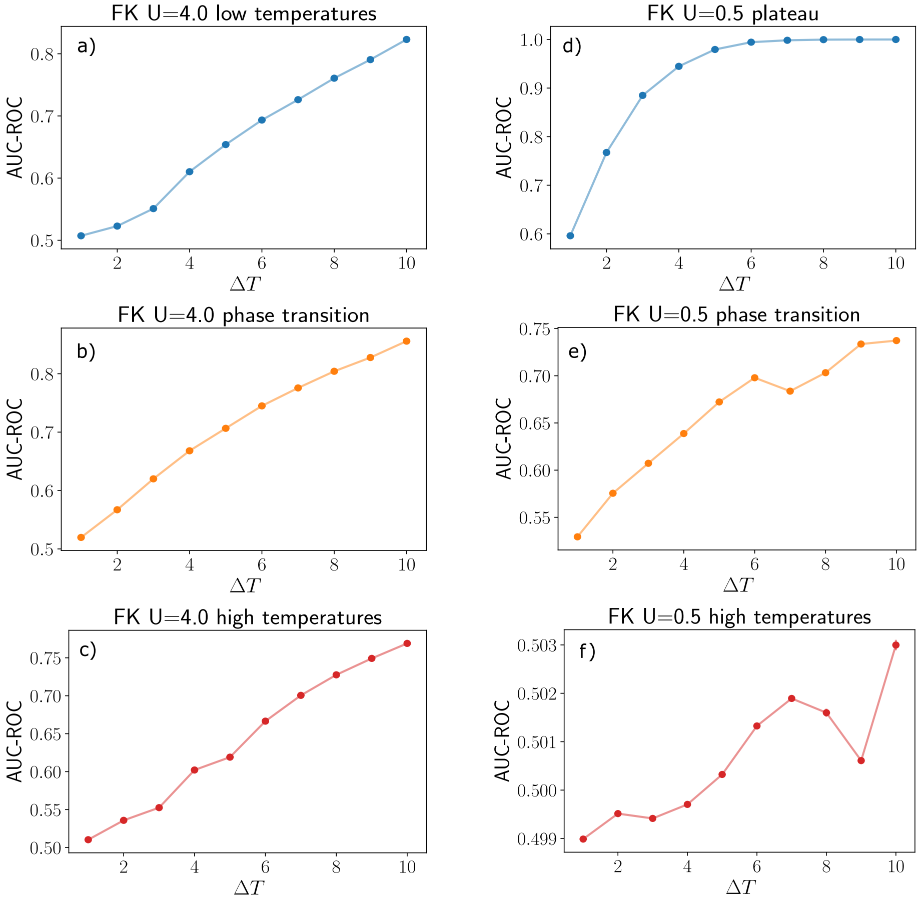


Fig. 4.37 The AUC–ROC metrics as the function ΔT, where $\Delta T = T_1 - T_2$ corresponds to the difference in temperature of MC configurations fed into the neural network. The left and right panels represent the case of continuous (FK model $U = 4.0$) and discontinuous (FK model $U = 0.5$) phase transitions, respectively. The calculations are performed for different positions of temperatures in relation to the critical point.

It is instructive to do the analogous analysis also for the other two models (qP and BC). Here, by a ’defect,’ we understand the deviation from a fully ferromagnetic state occurring in the 2×2 fragment of the MC configuration (Fig. 4.38). The results are presented in Fig. 4.39. The behavior of $\mathcal{C}(T)$ in the case of qP and BC models exhibiting second–order phase transition (Fig. 4.39a), c)) suggests a similar growth of defects as in the FK model $U = 4.0$ (Fig. 4.35d)).

On the contrary, we observe different dependence of $\mathcal{C}(T)$ on the temperatures for the first–order transition (Fig. 4.39b),d)). The defect concentration increases slowly for the temperatures $T < T_C$ and jumps suddenly for $T \approx T_C$. Such a characteristic justifies the lack of plateau in this temperature range – the neural network, with regard to the small number of defects and its slow growth, cannot distinguish between

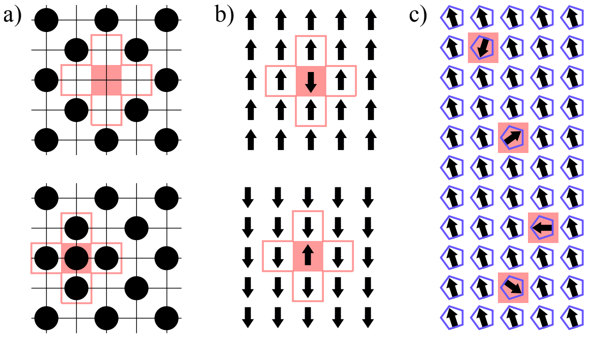


Fig. 4.38 Definitions of defects (marked in red) occurring in MC configurations simulating the FK model a), BC model b) and 5–state Potts model c).

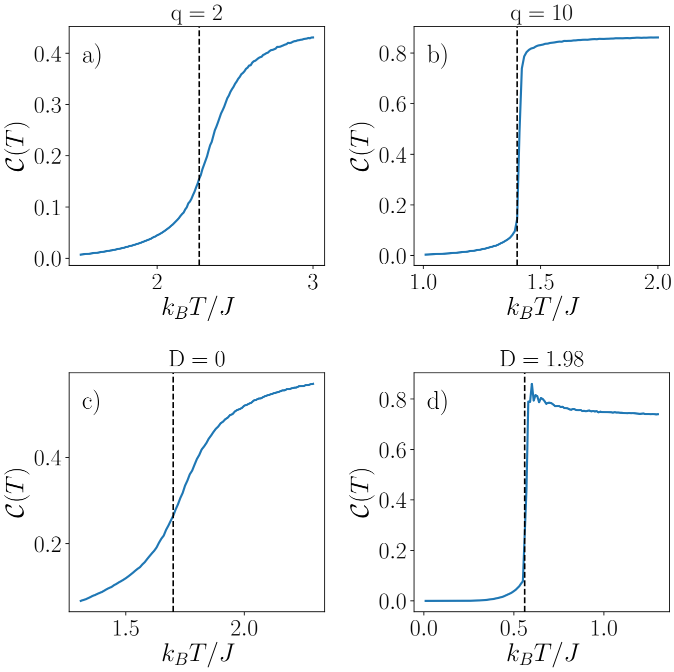


Fig. 4.39 The defect concentration $\mathcal{C}$ as a function of temperature for the qP (panels a) and b)) and BC (panels c) and d)) models.

configurations corresponding to two close temperatures. On the other hand, this step–like the behavior of $\mathcal{C}(T)$ makes the configurations generated for $T < T_C$ and $T > T_C$ easily distinguishable, leading to a higher value of AUC–ROC at $T \approx T_C$ for discontinuous phase transitions. The relation between an order of phase transition and the maximal value of the AUC–ROC function is shown in Fig. 4.41. Indeed, for all three models studied in the context of the LBC scheme, we observe slightly higher values of AUC–ROC in the case of discontinuous phase transition.

To confirm the hypothesis about the relationship between the number of defects and the neural network (NN) performance, we run explicit tests verifying the algorithm's ability to identify different defect concentrations. We train the neural network with configurations corresponding to two close temperatures T_1 and $T_2 = T_1 + \delta$, which are labeled as low- and high-temperature phases ($\delta T = 0.005\ k_B/t$

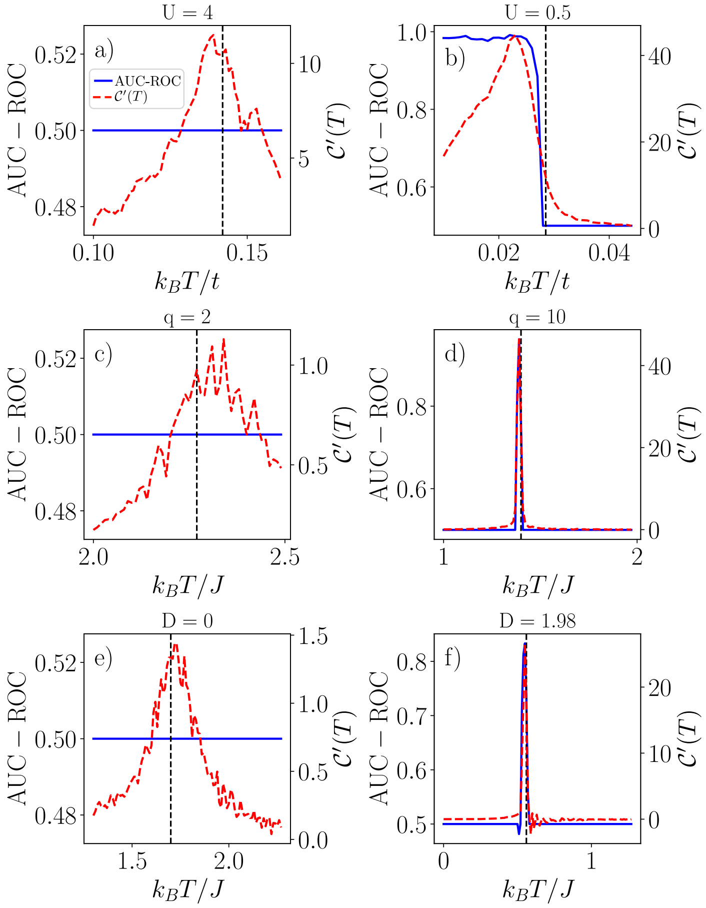


Fig. 4.40 Qualitative differences between configurations corresponding to two close temperatures T_1 and $T_2 = T_1 + \delta$ plotted as a function of $T = T_1$. The solid blue line represents the neural network performance $AUC - ROC$ in discriminating configurations generated at T_1 and T_2. The red dashed line shows the derivative of the defect concentration $\mathcal{C}'(T)$. Panels a), c), and e) present data for discontinuous phase transition, while b), d), and f) for continuous phase transitions, for the FK, qP and BC models respectively. The vertical black dashed line shows the value of T_c extracted from MC simulations.

for the FK model and $\delta T = 0.02\ k_B/J$ for the BC and Potts models). We then evaluate the ability of NN (AUC-ROC) to distinguish between these configurations and repeat the whole procedure for a wide range of different temperatures T_1 and T_2. Subsequently, we investigate how this effectiveness translates into the slope of $\mathcal{C}(T)$. The results are presented in Fig. 4.40, where AUC-ROC is compared to the derivative of $\mathcal{C}(T)$ approximated as $\mathcal{C}'(T) \approx \Delta\mathcal{C}(T)/\delta T$, where $\Delta\mathcal{C}(T) = \mathcal{C}(T + \delta T) - \mathcal{C}(T)$]. It can be seen that the AUC-ROC is exactly 0.5, which is the accuracy of a random classifier, unless $\mathcal{C}'(T)$ is really large. When the transition is second–order (left panel of Fig. 4.40), then AUC-ROC ≈ 0.5 in the entire temperature range. This stems from the fact that a change in defect concentration is not large enough to make the neural network predictions accurate. Note that this result is compatible with the shape of $P(\mathcal{C})$ distributions presented in Fig. 4.35 for the FK model. On the contrary, when the transition is of first–order, we observe a sharp peak around T_c in $\mathcal{C}'(T)$. For the qP and

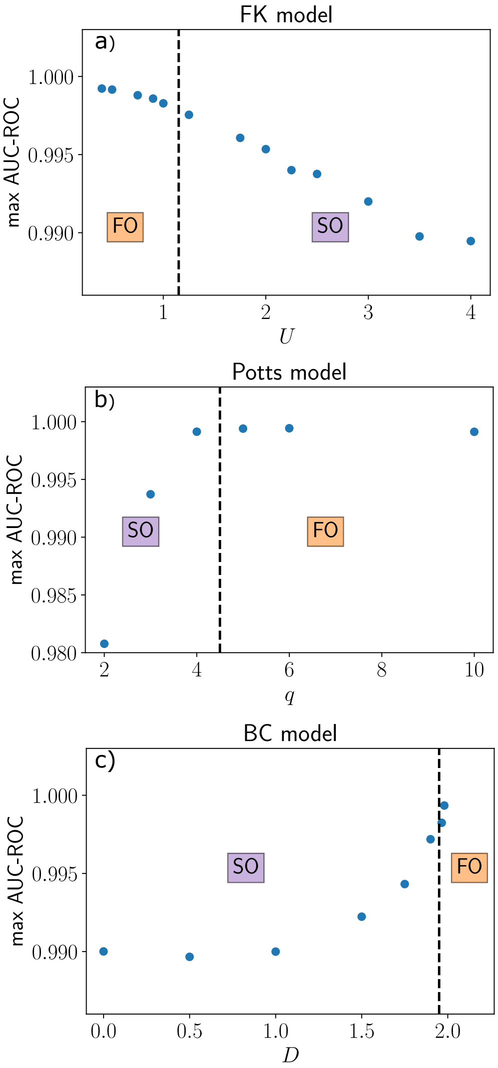


Fig. 4.41 The maximal value of AUC–ROC metrics obtained as a result of the application of the LBC scheme to three different models. The vertical dashed line indicates the value of the model's parameter separating between first–order (FO) and second–order (SO) phase transition.

BC models, it happens for a very narrow temperature range in the vicinity of the phase transition and is caused by the step–like increase in $\mathcal{C}(T)$. The situation is different for the FK model, for which we observe a plateau spreading over a large temperature range in the AUC–ROC function in the same region as depicted in Fig. 4.33 when the network was trained using the entire dataset. In our view, it is enough to prove that the algorithm can qualify the configurations generated in temperatures lying in the plateau as qualitatively different. Our preliminary results suggest that it may be not a finite–size effect. In Appendix D, we present the AUC–ROC curves obtained for the FK model for larger lattice sizes. For $U = 0.5$ we discover that the plateau emerging in the AUC–ROC function is maintained even for larger systems. At the same time, we find out that for $U = 4.0$ the maximum value of NN performance increases with lattice size L and reaches 1 in the thermodynamic limit, in agreement with the theoretical model presented in Sec. 4.3.1.1. This result also indicates that differences in the maximal AUC–ROC occurring between first– and second–order phase transitions (Fig. 4.41) are finite–size effects and can not be used to discriminate between these two kinds of phase transitions.

4.3.3 Summary

In the study presented above, we analyzed results of application of the LBC scheme to three different models undergoing first– and second–order phase transitions. First, we developed a theoretical model designed to predict the shapes of performance curves $P(T'_C)$ for two types of phase transitions. According to it, we should expect higher values of $P(T'_C)$ and a more emphasized maximum in the case of the second–order phase transition. Next, we juxtaposed the predictions from the model with the numerical calculations.

In the case of the classical qP and BC models, the algorithm copes well with the first– as well as the second–order phase transitions. In both cases T_C extracted from the position of maximum of AUC-ROC coincides with T_C from MC simulations, also in agreement with the theoretical model. However, we did not observed substantial differences in AUC-ROC amplitudes for the first and second-order transitions, as predicted by our simple theory.

For the FK model the LBC method gave trustful results only in the case of the second–order phase transition. Similar the other two models, the algorithm's performance displayed a well pronounced maximum at true T_C. For the first–order phase transition ($U = 0.5$), instead of a sharp peak predicted by the theoretical model, we observed a wide plateau in the AUC-ROC curve, with AUC-ROC ≈ 1. The plateau spreads from $T < T_C$ up to $T \approx T_C$ leading to the ambiguity of T_C. Such a result was unexpected since the plateau region goes far beyond the temperature range for the phase coexistence.

To explain the unusual behavior of the $P(T'_C)$ occurring for the FK model exhibiting first–order phase transition, we calculated, for all models under study, the rate at which defects arising in MC configurations grow with temperature T. We find that for the FK model that exhibits a first–order phase transition, the number of defects proliferates rapidly with T for $T < T_C$. It then saturates for $T \approx T_C$. Such a situation does not occur for the second–order phase transitions, for which we observe a slow increase of the defects in the whole temperature range. At the same time, we discover that the dynamics of $\mathcal{C}(T)$ differs for purely classical models. In the case of the qP and BC models, one can notice a sudden increase in the number of defects during the first–order phase transition. On the contrary, for the second–order phase transition, the behavior of function $\mathcal{C}(T)$ is the same for all analyzed models.

Our numerical results deviate from the predictions given by the simple theoretical model. This may result from the fact that phase coexistence has a marginal impact on the obtained shape of the AUC–ROC curves, and the other finite–size effects are more critical. Moreover, we observe that they are not universal and may vary from one microscopic model to another.

To conclude, we have shown that the LBC method can not be, in most cases, used to distinguish the order of the phase transition. Nevertheless, our results for the FK model show, that it is a promising tool capable of extracting new features of well known models that are overlooked by other, well established methods.

4.4 Application of the machine learning algorithms to the biophysical systems

4.4.1 Biology of ion channels

Ion channels are types of proteins located in the plasma membrane that are considered tunnels that having the ability to selectively pass ions in response to a chemical or mechanical signals [6, 149]. The study of ion channels is important with regard to their role in cell pathophysiology leading to many critical illnesses and their potential application as drug targets [150, 151]. Fig. 4.42 represents the basic activity of an ion channel – after a short period of opening (conducting state), the ion channel enters a closed state during which the passage of the ions is blocked.

The switching dynamics between the channel conformations: open (O) and closed (C), called channel gating, is usually very complex and depends on many different factors, including the interaction of the ion channel with other proteins (e.g. β–complexes) and presence of additional stimuli such as natural flavonoids [152], toxins (or other channel ligands).

The studies presented in the forthcoming Sections focus mainly on the analysis of mitoBK channel dynamics. The mitoBK channel is large conductance voltage- and calcium–activated potassium channel placed in the inner mitochondrial membrane [153, 154]. It is responsible for the mitochondrial potassium influx, regulates membrane potential, mitochondrial respiration, and matrix volume [154, 155]. It also contributes to cytoprotection [156]. It can stimulated by membrane depolarization, the calcium Ca^{+} ion concentration, protein–protein, protein–ligand or protein–lipid interactions or, as we shall see later, the presence of natural flavonoids such as naringenin [152, 157].

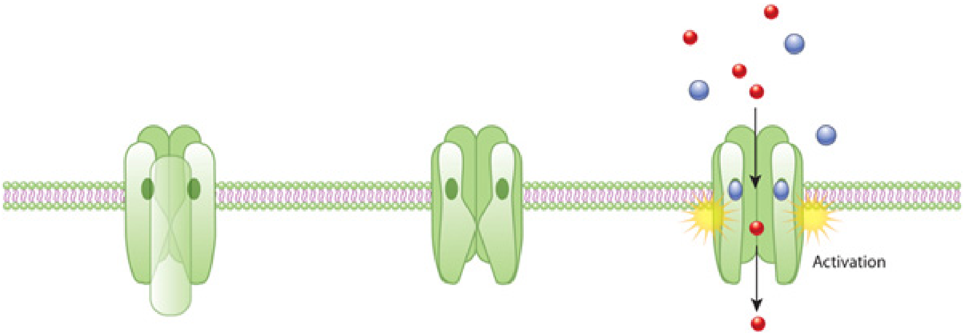


Fig. 4.42 Illustration presenting a basic activity of an ion channel [6]. After a short period of opening (conducting state), the ion channel enters a closed state during which the passage of the ions is blocked.

The primary source of knowledge about ion channel activation is the data obtained from the patch–clamp experiment [158]. It consists of placing a micropipette on a small piece of membrane called a patch. Such a micropipette registers currents flowing through the fragments of the cellular membrane, yielding important information about its gating and activation. From a mathematical point of view, such currents take the form of time series (described in more detail in Sec. 3.9). They can be examined with various statistical analysis methods, including machine learning algorithms. Such techniques can detect single–molecule events [159] and recognize different genetic features of ion channels [160]. They can also be considered, as we shall further see, good classificators revealing different mechanisms driving ion channel gating and activation. The development of new methods of ion channel analysis is important because they can provide potentially crucial information about the channel's conformational dynamics without using cumbersome and computationally expensive MD (*molecular dynamics*) simulations.

In the forthcoming Sections, we present applications of different data mining algorithms to the analysis of the activity of the mitoBK ion channel. We demonstrate that they provide information about the ion channel activity, which can not be extracted from classical methods of kinetic analysis.

4.4.2 Classification of mitoBK traces registered in different cell types

The channel proteins can be associated with different types of regulatory β (1–4) and γ (1–4) subunits [161, 162]. They are exhibited in the tissue–specific manner. As a part of this study, we aim to find single–channel activity patterns coming from three different cell types: human endothelial cell line (EA.hy926), primary human dermal fibroblasts cell line (HDFa), and embryonic rat hippocampal neurons. Examples of recordings obtained for all three cell types at different values of micropipette potentials acquired by our collaborators [163] ranging from $U_{min} = -60$ mV to $U_{max} = 60$ mV are presented in Fig. 4.43.

The recordings represent various levels of mitoBK ion channel activation increasing non–linearly with an increase of U as presented in Fig. 4.44. As can be noticed, the mean open state probabilities P_{op} discriminate to some extent between recordings corresponding to different cellular lines for intermediate potentials U. They become, however, indistinguishable for $U = 60$ mV. Moreover, the P_{op}'s do not, provide any information on the variability of the recorded signals on short time scales. More specifically, from p_{op}, one cannot find the differences in local conformational dynamics that occur in various cellular lines.

To overcome this problem, we create a simple algorithm based on time series analysis in conjunction with machine learning methods. Its main objective is to answer the following questions: 'Are the signals corresponding to different cellular lines characteristic enough to be distinguishable even at the short–time scales? In particular, can they be identified even in the case when the mean level of ion channel's activation

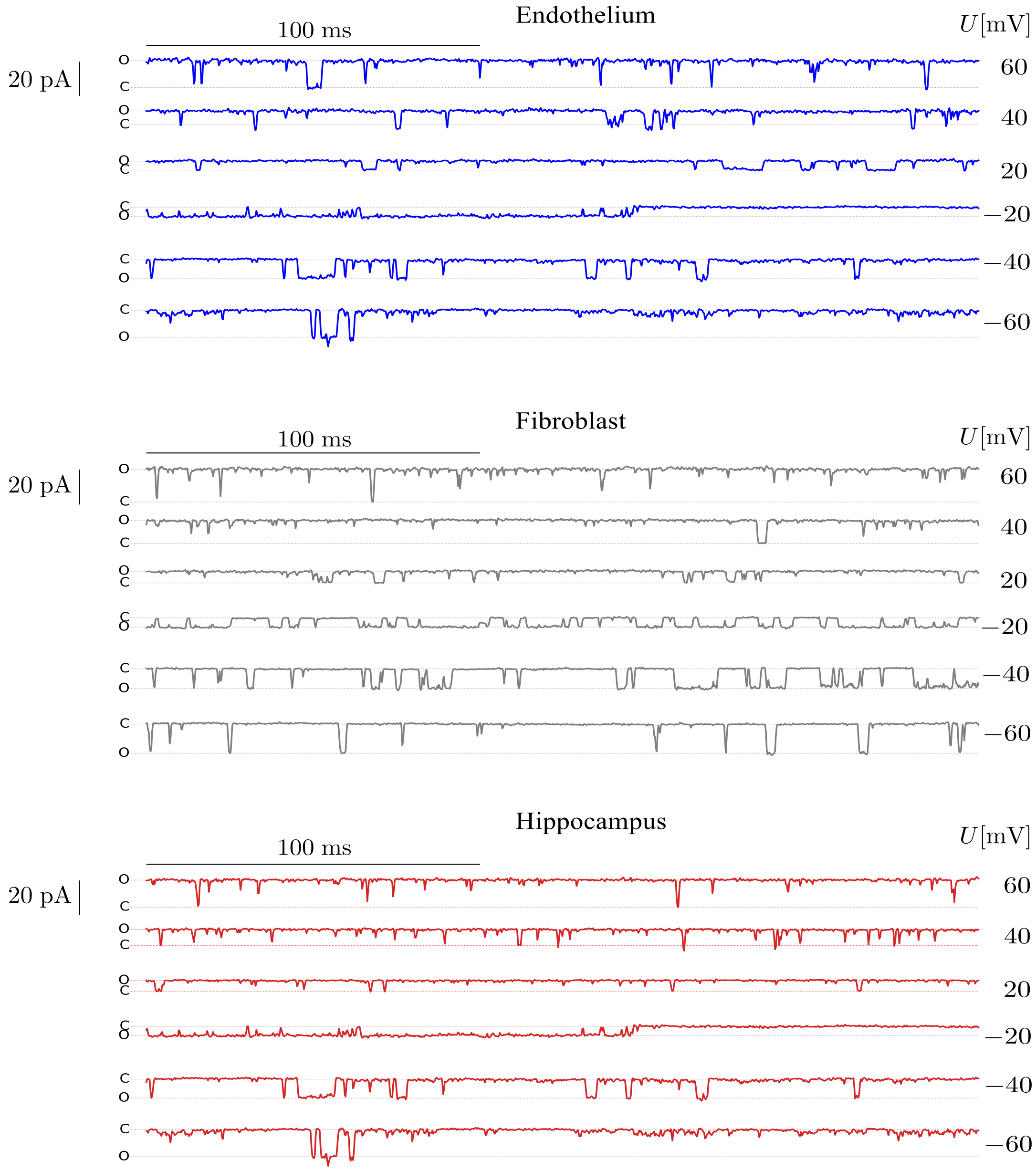


Fig. 4.43 A set of recordings coming from the patch-clamp experiment performed on different cell types: endothelium, fibroblast and hippocampus at different values of pipette potentials U. The value of 20 pA denotes the amplitude of the obtained ion current. The symbols C and O stand for the closed and open states of the ion channel, respectively.

is approximately the same?'

Our analysis consists of several basic steps:

1. Data–preprocessing stage,

2. Removal of outlier recordings with the use of an autoencoder neural network,

3. Data division into training and testing datasets,

4. Classification of samples with the KNN algorithm.

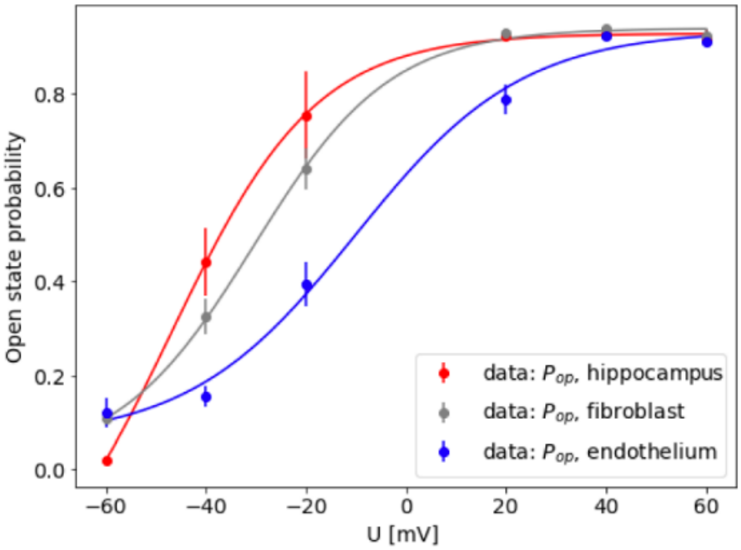


Fig. 4.44 The mean open state probability P_{op} in function of micropipette potential U for three different cellular lines.

The data consists of mitoBK current's recordings obtained from the patch–clamp experiment. For analysis, we use traces corresponding to different cellular lines and membrane potentials U_m. Each recording consists of 25 000 samples obtained at the sampling frequency $f_s = 10$ kHz. They are then divided into a set of overlapping subsequences of length $l = 1000$ with a stride $r = 200$ (Sec. 3.9). Such created subseries are sufficiently long to grasp the characteristic features of the ion channel activity. They are then reduced from $l = 1000$ to $l' = 200$ with the PAA method (Sec. 3.9.1) illustrated in Fig. 4.45b). This step is necessary to prevent overfitting during the evaluation of a machine learning algorithm. Note that the application of this technique does not change signal's trend preserving information about the ion channel activity and its conformational dynamics.

The next step of our analysis involves removal of outlier recordings obtained at the same external conditions (membrane potential U_m and cellular line). For this purpose, we apply a simply, fully–connected autoencoder neural network (see Sec. 3.7) which encodes raw input signal into 2–dimensional latent space. Exemplary results of this transformation are illustrated in the upper panel of Fig. 4.45a), where different subsequences of length $l' = 200$ obtained from the same recording are marked with the same colors. In the presented example, one can see that one recording clearly stands out from the rest. It is then treated as an outlier and discarded from further analysis.

Subsequently, we split remaining recordings into the training and testing datasets. We apply the following train-test split strategy. At the fixed value of potential U_m we choose one recording of each cell type. Then subsequences created out of these three recordings fall into the testing dataset. The subseries corresponding to the remaining traces make the training dataset. We repeat this procedure until all possibilities of such a division are exhausted.

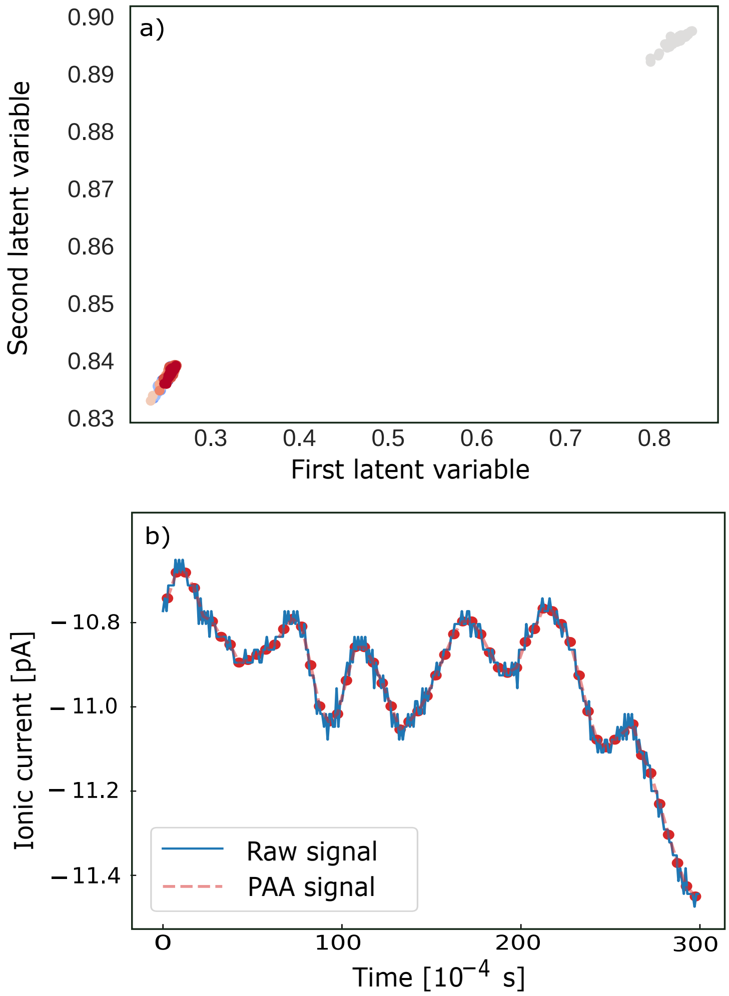


Fig. 4.45 Elements of the preprocessing stage and the outlier removal. The upper (a)) figure illustrates the representation of time series data points after reduction performed by an autoencoder neural network. One of the clusters stands out from the rest – it is treated as an outlier and discarded from further analysis. The lower figure (b)) represents the PAA method. The solid blue line represents a raw signal. The orange dots mark the signals obtained after the PAA transformation.

Ultimately, subsequences are fed into a KNN (Sect. 3.9.2). The performance of such a trained algorithm is presented in Tab. 4.2. For comparison, we show the classification accuracies obtained in two cases: with application of the autoencoder anomaly detection technique (AD) in the preprocessing stage and without. As expected, we observe a significant improvement in the results when outliers are removed prior to training. The Tab. 4.2 shows also that the worst results are obtained for $U_m = -20$ mV. We suspect that this moderate value of the pipette potential has a too small impact on a channel pore. As a consequence, the energetic differences between the ion channel conformations corresponding to different cellular lines are, in some cases, indistinguishable for the KNN.

Additionally, to make our studies more profound, we illustrate the samples in

Potential U_m	Acc. Without AD	Acc. With AD
60 mV	96.6 %	96.6 %
40 mV	72.9 %	99.5 %
20 mV	75.5 %	96.5 %
-20 mV	84.3 %	84.3%
-40 mV	59.9 %	95.9 %
-60 mV	77 %	100 %

Table 4.2 The classification accuracies of the mitoBK channel recordings corresponding to different cellular lines. The compared samples are obtained at the fixed value of the membrane potential U_m applied during the patch–clamp experiment. The middle and last columns shows the performances without prior use of the anomaly detection techniques (AD) and after dispose of outliers, respectively.

the 2–dimensional latent space of an autoencoder (Fig. 4.46). In almost all cases, we observe a clear separation between clusters corresponding to samples representing different cellular lines. This separation varies for different values of U_m. For $U_m = -20$ mV the clusters representing the hippocampus and the fibroblast cells clearly overlap, making the ML classification much more difficult. It is compatible with the results presented in Tab. 4.2 - the lowest value of accuracy is obtained for $U_m = -20$ mV. To quantitatively compare the level of separability between recordings corresponding to different cellular lines, we apply the k–Means ($k = 3$) algorithm (see Sect. 3.4). It identifies the center's positions of the emerging clusters c_i. Then we treat the pairwise Euclidean distances (presented in Tab. 4.3) between the clusters' centers c_{ij} as a measure of data separability. For the sake of clarity, the presented data are normalized with a reference to the smallest distance between the clusters, which is set to 1 for a fixed pipette potential U_m.

In general, for most potentials (excluding 40 mV) we observe the greatest distances between data points representing the hippocampal and endothelial cells. Furthermore, the hippocampal and fibroblast cellular lines seem to be closer to each other compared to the endothelial line, which appears to be more distant from both the hippocampus and the fibroblasts. This result is not so surprising taking into account the mean open state probability functions presented in Fig. 4.44 which reflect similar dependencies.

In summary, this simple analysis of mitoBK channel ion currents reveals qualitative differences in the ion channel activities corresponding to different cellular lines at the fixed value of the pipette potential. It suggests that the mechanism governing the mitoBK ion channel gating depends on the cell type. The fact that disparities in the

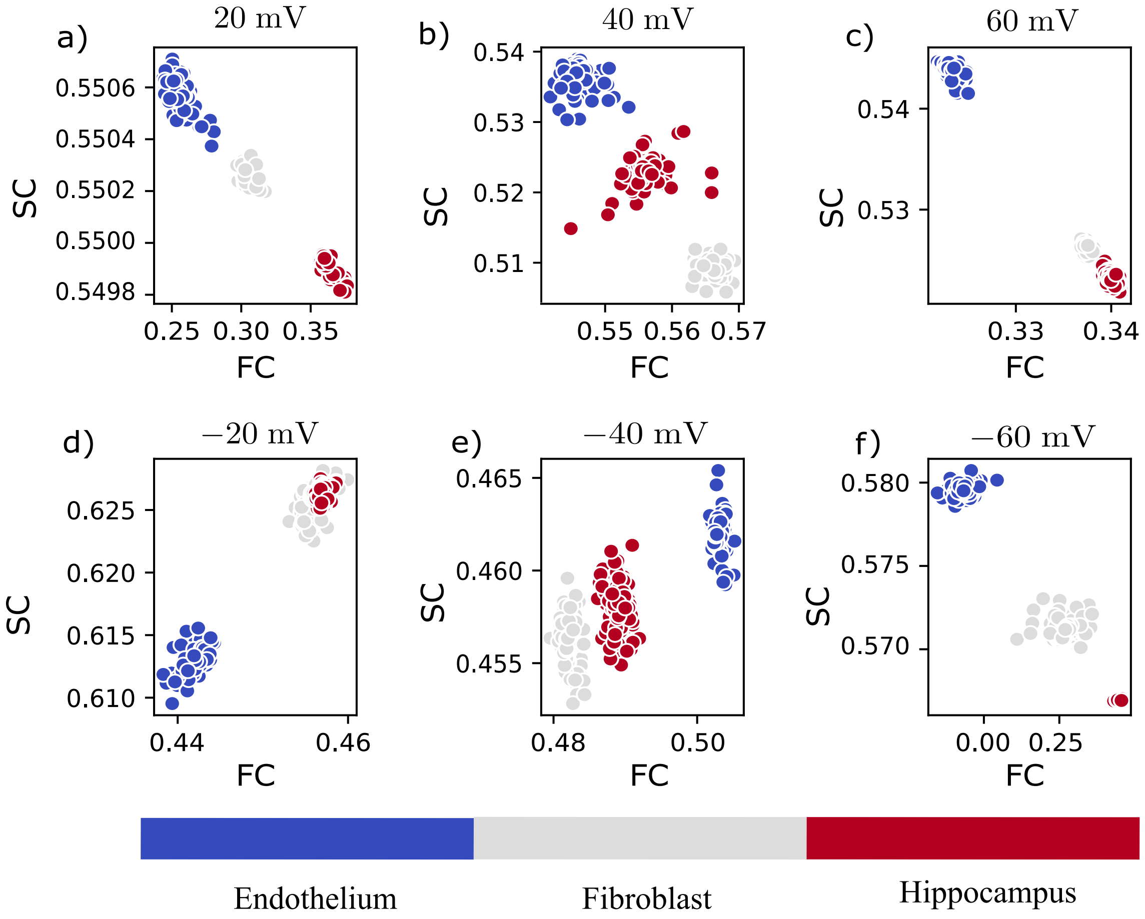


Fig. 4.46 The visualization of time series samples, obtained at different values of micropipette potential, presented in the 2–dimensional latent space of the applied autoencoder. Different colors of data points correspond to subsuquences representing 3 distinct cellular lines.

signal's characteristics occur even at the same level of activation (which happens, for instance, for $U_m = 60$ mV) further strengthens our hypothesis.

4.4.3 Study of the impact of naringenin on the mitoBK channel activity

In general, mitoBK channels are modulated by two basic stimuli: membrane depolarization and Ca$^+$ ions [164]. However, there exist many other modulators of the mitoBK channels such as a mechanical strain [165], channel openers NS1619, NS11021, CGS7181, CGS7184 paxilline (PAX), charybdotoxin (ChTx), iberiotoxin (IbTx), 12 4-aminopyridine (4-AP) or tetra-ethyl ammonium (TEA) [166, 167, 168, 169]. Unfortunately, the majority of them exhibit broad off-target effects, including their cytotoxicity [170]. Therefore, it is extremely important to find other sources of modulation. One of the possible choices are flavonoids that are cost-effective, non-toxic and easily available [152]. Among them, the naringenin (Nar) due to its antioxidant, cytoprotective and anti–inflammatory properties, seems to be a perfect mitoBK channel modulator. Many studies confirmed that its presence significantly increases the level of mitoBK channel activation [167, 171, 172].

Potential U_m	Dist FH	Dist FE	Dist HE
60 mV	1	$1,92$	$2,46$
40 mV	$1,72$	$1,99$	1
20 mV	$1,06$	1	$1,74$
-20 mV	1	$1,73$	$2,37$
-40 mV	1	$1,64$	$1,87$
-60 mV	1	$1,97$	$2,42$

Table 4.3 The normalized distances between the clusters representing different cellular lines for various potentials U_m normalized to the smallest distance. FH, FE, HE represent the distances between centroids of the 'fibroblast' and 'hippocampus', 'fibroblast' and 'endothelium', 'hippocampus' and 'endothelium' datapoints, respectively.

The purpose of our study is to find out, if apart from the increase in the probability of opening of the channel P_{op}, the addition of the naringenin shapes locally patch–clamp recordings. To this end, we analyze the traces obtained at different values of pipette potential U and naringenin concentration [Nar]. Some representative sequences obtained at different external conditions are presented in Fig. 4.47.

At the beginning, we categorize the data into three sets characterized by approximately the same value of P_{op}s. They are presented in Tab. 4.4 where they are marked in three different colors. It is important to note that although recordings assigned to the same group exhibit the same mean level of mitoBK channel activation, they are obtained at various values of naringenin concentration [Nar] and micropipette potentials U_m.

	0 μM	**1 μM**	**3 μM**	**10 μM**
20 mV	0.59 ± 0.04	0.60 ± 0.03	0.65 ± 0.02	0.71 ± 0.02
40 mV	0.56 ± 0.03	0.58 ± 0.02	0.63 ± 0.02	0.66 ± 0.02
60 mV	0.53 ± 0.05	0.53 ± 0.02	0.58 ± 0.05	0.60 ± 0.02

Table 4.4 The mean open state probabilities of the mitoBK channel obtained at different values of pipette potentials $U_m \in \{20\ \text{mV},\ 40\ \text{mV},\ 60\ \text{mV}\}$ and the naringenin concentrations $[Nar] \in \{0\ \mu\text{M}, 1\ \mu\text{M}, 3\ \mu\text{M}, 10\ \mu\text{M}\ \}$. The cells representing the groups of recordings compared by ML algorithms are marked with the same colors.

Subsequently, we apply a ML algorithm separately for each of the highlighted groups of recordings. We discover, however, that raw patch–clamp recordings return poor classification accuracies. For this reason, prior to training, we transform them into

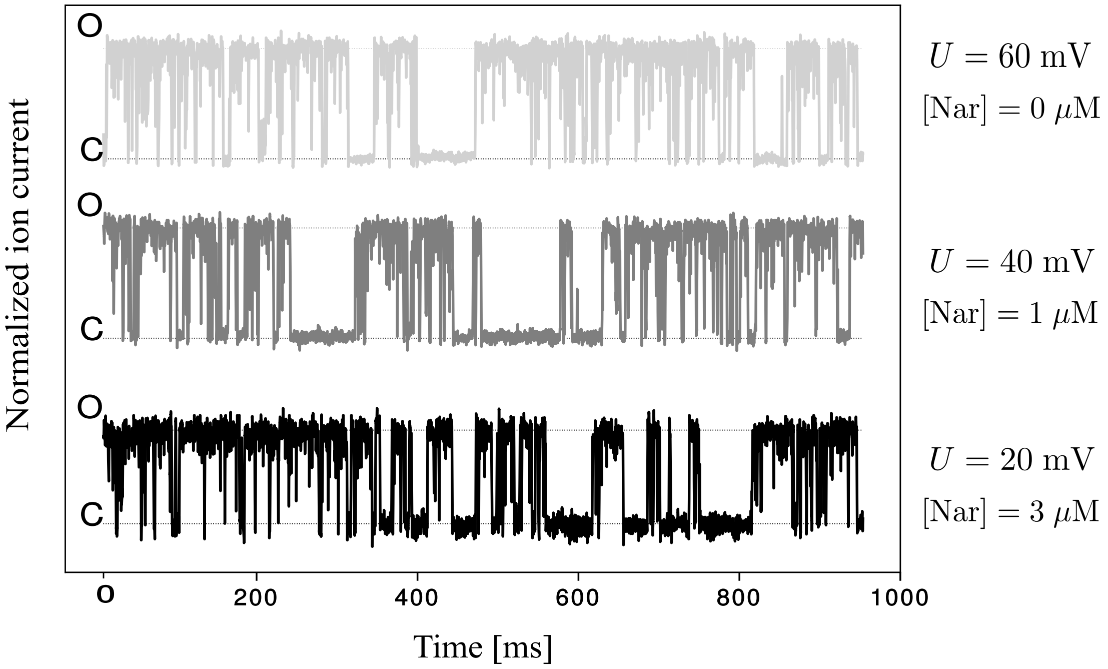


Fig. 4.47 The examples of normalized patch–clamp recordings obtained from patch–clamp experiment for different values of naringenin concentration $[Nar]$ and pipette potentials U. The letters O and C indicate the open and closed states, respectively.

the dwell–time series. They are defined as a normalized series of subsequent durations of open (O) and closed (C) states (Fig. 4.48). The construction of a dwell–series requires however the determination of a threshold current value (TC) discriminating between conducting and non–conducting states. It can be achieved using the procedure described in more detail in [173].

The reason behind the unsatisfactory results obtained from the analysis of the raw experimental data is twofold. First of all, dwell–time sequences are much shorter than original raw samples. It entails the condensation of the information about the ion channel activity - our data set consists of shorter time series containing more knowledge about switching dynamics, which turns out to be crucial for the right recordings' classification. Moreover, such a constructed data set does not exhibit spontaneous fluctuations, which can be a consequence of a highly complex alternation pattern between possible protein conformations. Such fluctuations have no impact on the general trend of signals, but can substantially impede the classification performed by an ML algorithm.

To classify the sequences of dwell-time series, we decided to apply, on the first try, a simple KNN algorithm with Euclidean distance metrics. This choice is motivated by its effectiveness in ion channel classification corresponding to different cellular lines, as shown in the previous Section. Apart from that, it is fast and does not require extensive parameter tuning (the only free parameter is the number of neighbors).

However, before feeding dwell-time series into the ML algorithm, it is necessary

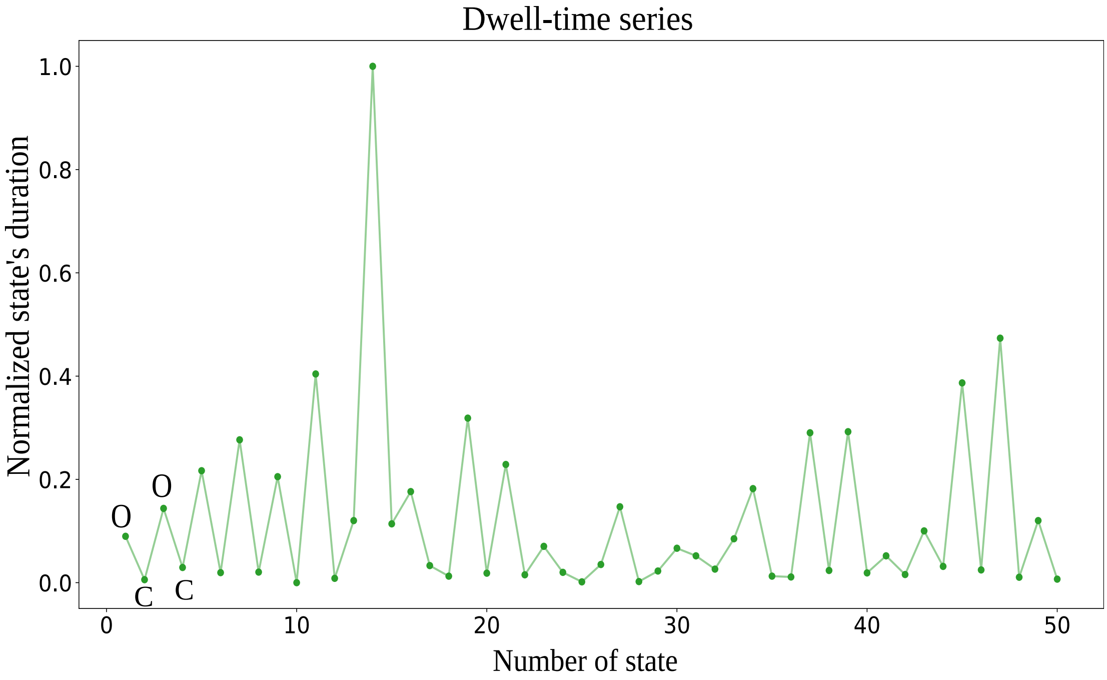


Fig. 4.48 An example of dwell-time series of length 50 representing normalized values of duration of subsequent open (O) and closed (C) ion channel states.

to divide them into a set of smaller subseries. We discover that an optimal length of a single dwell-time subsequence is N = 50. It turns out to be the best compromise between the dimension of the training data set and the number of samples (the larger the lengths of individual samples, the smaller their number). We apply no overlap between consecutive subseries to make the samples more 'independent'. Additionally, to avoid the class imbalance problem [174], the number of data points for each of the groups analyzed is set the same. It requires shortening some traces to the same length. In such a manner, for each of the 3 classes of recordings, we obtain approximately 100 samples (dwell–time subsequences). Subsequently, they are globally (separately for each class) normalized into the range $[0, 1]$. The illustration of general steps taken during preprocessing–stage is presented in Fig. 4.49.

After applying the KNN method to such a created data set, we find poor prediction accuracies. Although the results obtained suggest that the algorithm classifies the recordings in a non–random manner (we obtain $Acc \approx 60\%$ for binary classification problem and $Acc \approx 50\%$ when algorithms have to deal with three categories), there is much room for further improvement. To increase the classification performances, we make a second attempt with the shapelet method described in Subsec. 3.9.3 which, contrary to the KNN gives much more insight into the data by analyzing it *locally* rather than *globally*. The comparison of these two algorithms for three groups of recordings elucidated in Tab.4.4, is presented in Tab.4.5.

Indeed, we discover that the shapelet method deals better with the classification of dwell-time series than the KNN. Taking into account that the compared groups

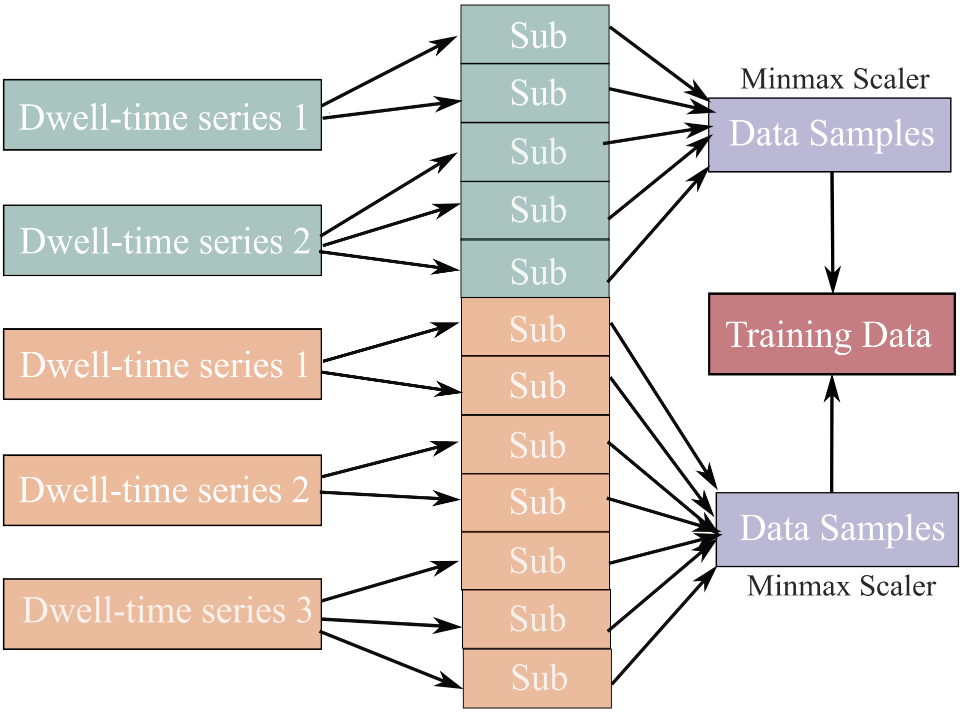


Fig. 4.49 Illustration of the data–preprocessing stage. Raw dwell–time recordings corresponding to different classes are, in the first step, divided into a set of smaller subseries. All subseries assigned to the same category are then globally normalized into the range $[0, 1]$ with the MinMax Scaler. The ultimate training (and testing) data set is created by combining all such prepared subsequences.

of recordings are characterized by the same values of mean open state probabilities P_{op}s (thus exhibiting the same global feature), this result should not be surprising. However, although the shapelet method exhibits more discriminative power than the KNN technique, the ratio of misclassified samples is still significant. One can interpret that in terms of a switching mechanism between conducting and non–conducting states common to both modulators.

Another interesting question that arises in the context of conducted studies concerns the significance of the impact of two basic external stimuli on mitoBK channel activation. To investigate this problem, we perform the ML classification for the dwell–time series corresponding to two different naringenin concentrations [Nar] and fixed pipette potential U_m, and vice versa. The results obtained both with the KNN and shapelet methods are shown in Tab. 4.6.

The first group of recordings is characterized by a fixed membrane potential ($U_m = 60$ mV) in the absence of naringenin (0 μM) and in its presence (10 μM). As one can see, the KNN algorithm deals better with discrimination of this group of recordings. This result may suggest that the presence of naringenin shapes primarily large-scale features of the channel gating. The second group of recordings corresponds to a fixed naringenin concentration ([Nar]=10 μM) and different values of pipette potentials $U_m = 60$ mV and $U_m = 20$ mV. In this case, the predictions of the shapelet

Potential U_m	[Nar]	Acc. KNN	Acc. Shapelet
	First group		
60 mV	1 μM	62%	70%
20 mV	10 μM		
	Second group		
60 mV	3 μM	64%	71%
40 mV	10 μM		
	Third group		
60 mV	0 μM	49%	56%
40 mV	1 μM		
20 mV	3 μM		

Table 4.5 The comparison of classification accuracies of the KNN and *shapelet learning* methods presented for three different groups of patch–clamp recordings characterized by the same mean open state probabilities.

method are more accurate. It is probably due to the fact that the membrane potential has a greater impact on local shaping of the mitoBK activity than naringenin.

Furthermore, to make our analysis more insightful, we show in Figs.4.50 and 4.51 the shapelets obtained as a result of training the ML classificator on the groups of data presented in Tab. 4.6. The shapelets of length $N_s = 20$ are imposed on parts of dwell-time samples in their best–matching locations. As one can see in Fig. 4.50, the shapelets which to the largest extent discriminate recordings corresponding to the extreme values of naringenin concentrations (0 μM and 10 μM), represent the extracts of dwell-time sequences characterized by large internal variability. In contrast, shapelets that separate data representing different levels of voltage–activation, seem to describe rapid switchings to long–lasting states as shown in Fig.4.51.

The separation between dwell–time subsequences representing various classes can be visualized in a two–dimensional space $\mathcal{S}_2$. As described in Sect. 3.9.3, to perform the transformation of dwell–time samples to $\mathcal{S}_2$, it is necessary to choose two most representative shapelets s_1 and s_2 and calculate their distances to all samples in the data set $\mathbf{k}$. In this way, two feature vectors $d(s_1, \mathbf{k})$ and $d(s_2, \mathbf{k})$ become coordinates of the dwell-time samples in this new space. We perform this analysis for the two groups of recordings enumerated in Tab. 4.6. The chosen shapelets s_1 and s_2 used in the transformation, which correspond to dwell–time series differing by the values

Potential U_m	$[Nar]$	Acc. KNN	Acc. Shapelet
DIFFERENT NARINGENIN CONCENTRATIONS			
60 mV	10 μM	76%	62%
60 mV	0 μM		
DIFFERENT VALUES OF PIPETTE POTENTIALS			
60 mV	10 μM	66%	82%
20 mV	10 μM		

Table 4.6 The accuracies of the KNN and *shapelet learning* methods presented for two groups of recordings. The first one is characterized by the same value of pipette potential $U_m = 60$ mV and different values of naringenin concentrations $[Nar] = 0$ μM and $[Nar] = 10$ μM. The second group of recordings corresponds to the same value of naringenin concentration $[Nar] = 10$ μM and different values of pipette potentials $U_m = 60$ mV and $U_m = 20$ mV.

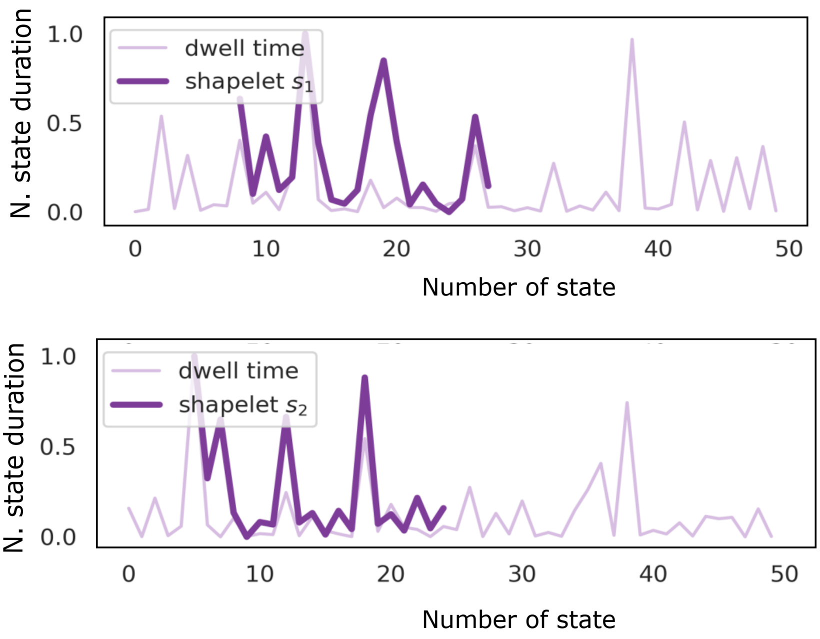


Fig. 4.50 Examples of dwell time series samples (of length $N = 50$) with shapelets (of length $N_s = 20$) imposed on the signals in their best–matching locations. The compared groups of recordings correspond to the same value of pipette potential $U_m = 60$ mV and different values of naringenin concentration $[Nar] = 0$ μM (upper figure) and $[Nar] = 10$ μM (lower figure).

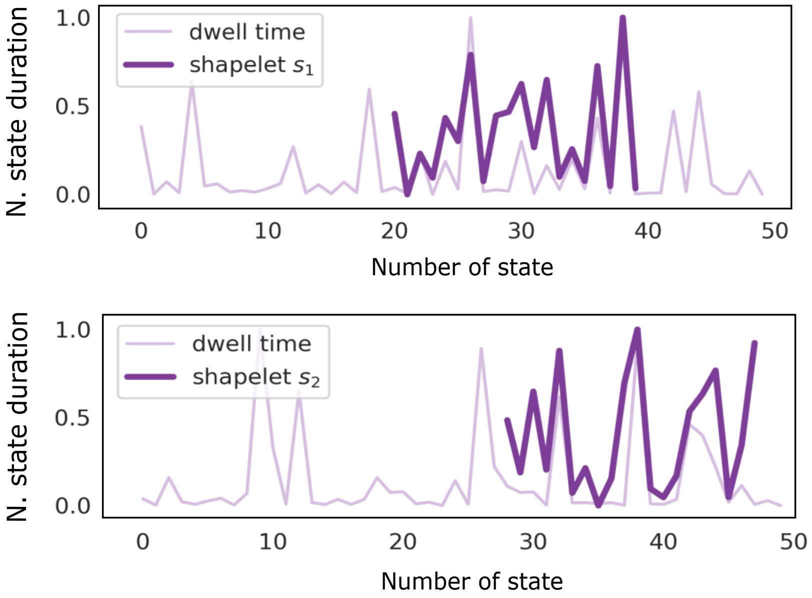


Fig. 4.51 Examples of samples of dwell time series (of length $N = 50$ with) with shapelets (of length $N_s = 20$) imposed on the signals in their best–matching locations.. The compared groups of recordings correspond to the same value of naringenin concentration $[Nar] = 10$ μM and different values of pipette potentials $U_m = 20$ mV (upper figure) and $U_m = 60$ mV (lower figure).

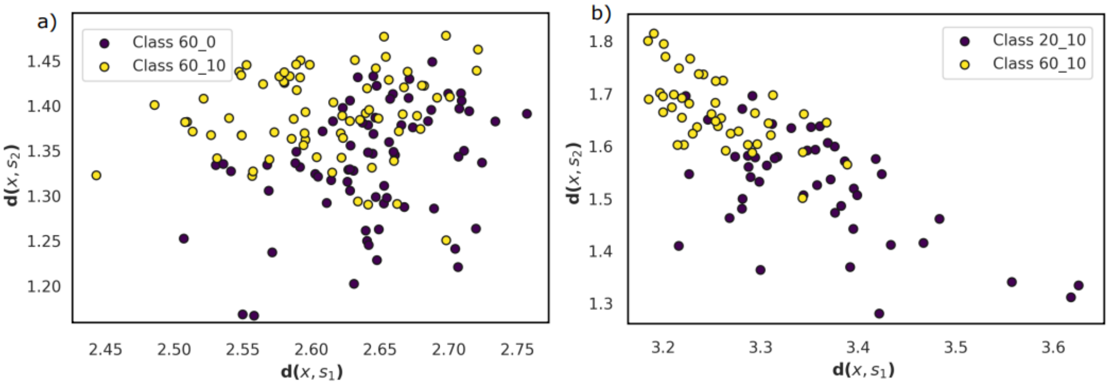


Fig. 4.52 The shapelet-transform representation of the input data describing mitoBK activity. On the left panel, the presented points correspond to data obtained at the fixed value of membrane voltage $U_m = 60$ mV and different values of naringenin concentrations: $[Nar] = 0$ μM (purple dots) and $[Nar] = 10$ μM (yellow dots). On the right panel, points represents the dwell-time samples obtained at the fixed naringenin concentration $[Nar] = 10$ μM and different values of membrane potentials: $U_m = 20$ mV (purple dots) and $U_m = 60$ mV (yellow dots).

of applied voltage and naringenin concentration are presented in Figs. 4.51 and 4.50, respectively. New representations of the analyzed samples are illustrated in Fig. 4.52. One can observe the large overlapping of points corresponding to the set of recordings obtained at different naringenin concentrations. In contrast, the data points representing various levels of membrane potential are more separated. These findings coincide with the classification presented in Tab. 4.6 and suggest that the membrane potential has a greater impact on the dynamics of the local mitoBK channel.

4.4.4 Summary

The results of the studies presented in the previous two sections are of great importance for a more profound understanding of the mitoBK channel's dynamics. We discover that the activation of these ion channels depends on many factors, including the type of cell and the presence of additional external modulators such as naringenin. The underlying differences cannot always be detected by classical methods. Some of them can be, however, identified by the machine learning algorithms such as KNN or shapelet methods. We believe that the novel, alternative methods of the patch–clamp recordings' analysis introduced in the presented research will significantly contribute to the understanding of the ion channel switching dynamics and discovery of new modulation sources free of the off–targets effects.

CHAPTER 5

Conclusions

The studies presented in this Ph.D. thesis constitute the summary of the research conducted in 2016–2022. They focus on the possible applications of machine learning methods which, due to the rapid development of new technologies, have experienced a renaissance. The studies are mainly concerned with the physics of phase transitions. We dedicate it to three Sections containing:

- the neural network's estimation of the critical temperature of the BKT phase transition,
- determination of the order parameter and the nature of phase transitions with the simple unsupervised learning methods,
- discrimination between first–order and second–order phase transitions with the learning by confusion scheme.

In the first Section, we find out that the BKT is difficult to analyze with regard to its topological character and continuous degrees of freedom. For some models, to obtain a sufficiently precise estimation of the critical temperature T_{BKT}, it is necessary to train the neural network on the MC configurations corresponding to temperatures lying near the transition point. It is due to the fact that the configurations obtained at extreme temperatures ($T \gg T_C$) and ($T \ll T_C$) are much more different from those generated for $T \approx T_C$.

In the next Section, we attempt to estimate the order parameter with PCA and K–means techniques. For the three models under study, we find that the distance between clusters obtained after the PCA transformation (and representing the ground states of the given model) correlates well with the order parameter. In addition, we have discovered that the evolution of the 2–dimensional PCA representation of the raw MC configurations with the temperature differs among two types of phase transitions, which can hint at their nature.

In the last Section concerning the physics of phase transition, we analyze the shape of the performance curve obtained with the learning by confusion scheme (LBC) for the same models analyzed already in the previous Section. We conclude that although

LBC deals quite well with the continuous phase transition and is able to determine the critical point precisely, it sometimes fails to correctly identify the moment of a phase transition in the case of discontinuous phase transitions. Such a situation occurs in the Falicov–Kimball model in the weak interaction regime. The performance function forms a plateau spreading over many temperatures in this case. We associate this uncommon behavior with the rate at which the defects of MC configurations grow with the temperature. We discover that their form and ways of proliferation have a decisive impact on the performance curve. We find that although the LBC scheme can not be used to identify the order of the phase transition, it can serve as a tool allowing us to study the finite–size effects of the MC configurations varying between different models of the condensed matter physics.

The remaining part of the thesis presents the application of ML methods to the analysis of signals representing the activity of ion channels. We show that the simple K–Means algorithm with Euclidean distance metrics is able to discriminate between signals corresponding to different cellular lines. Furthermore, we find out that the application of the shapelet method can reveal a stimuli of an ion channel based solely on the shape of the signal.

In conclusion, the dissertation has shown that the methods of artificial intelligence have great potential to become new tools for analyzing the physics of different systems. The main challenge with applying ML methods on a bigger scale in science is the need for their clear interpretability. Therefore, the classical methods of analysis still lead the way. This situation may change in the forthcoming years, during which new ML techniques will surely be developed. The first steps toward the interpretation of the 'black box' have already been taken [175, 176]. They will probably significantly contribute to new discoveries in the field of science.

Annotations

This Ph.D. thesis was based on four articles [70, 163, 177, 178]. Ref. [178] has the form of preprint and is currently under review in *Phys.Rev.E.* It is important to note that during Ph.D. studies, one also published other articles going beyond the topic of this dissertation. They include studies concerning:

- detection of characteristic points in the impedance signal (ICG) monitoring the heart activity [179, 180],
- new discrete symmetries in the extended Standard Model [181, 182, 183, 184],
- decoherence in the open quantum systems [185, 186, 187, 188]

and one review [152] describing impact of the flavonoids on the potassium ion channel activation.

REFERENCES

[1] Christoph G. Salzmann. Advances in the experimental exploration of water's phase diagram. *The Journal of Chemical Physics*, 150(6):060901, 2019.

[2] Lenka Zdeborová. Machine learning: New tool in the box. *Nature Physics*, 13, 02 2017.

[3] Edwin Bedolla-Montiel, Luis Padierna, and Ramon Castaneda-Priego. Machine learning for condensed matter physics. *Journal of Physics Condensed Matter*, 33:053001, 01 2021.

[4] Junwei Liu, Yang Qi, Zi Yang Meng, and Liang Fu. Self-learning monte carlo method. *Phys. Rev. B*, 95:041101, Jan 2017.

[5] John. Elder, ACM Digital Library., Association for Computing Machinery. Special Interest Group on Knowledge Discovery and Data Mining., and Association for Computing Machinery. Special Interest Group on Management of Data. *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2009.

[6] *Essentials of Cell Biology*. Cambridge, MA: NPG Education, 2010.

[7] Yeomans J.M. *Statistical Mechanics of Phase Transitions*. 05 1992.

[8] Nigel Goldenfeld. *Lectures on phase transitions and the renormalization group*. CRC Press, 2018.

[9] P. Papon, Jacques Leblond, and P.H.E. Meijer. *The physics of phase transitions: Concepts and applications*. 01 2006.

[10] H Eugene Stanley and Victor K Wong. Introduction to phase transitions and critical phenomena. *American Journal of Physics*, 40(6):927–928, 1972.

[11] Jean Ginibre. General formulation of griffiths' inequalities. *Communications in mathematical physics*, 16(4):310–328, 1970.

[12] Ernst Ising. Beitrag zur theorie des ferromagnetismus. *Z. Phys.*, 31:253, 1925.

[13] Lars Onsager. Crystal statistics. i. a two-dimensional model with an order-disorder transition. *Phys. Rev.*, 65:117–149, Feb 1944.

[14] Leo P Kadanoff. Scaling laws for ising models near t c. *Physics Physique Fizika*, 2(6):263, 1966.

[15] John Cardy. *Scaling and renormalization in statistical physics*, volume 5. Cambridge university press, 1996.

[16] F. Y. Wu. The potts model. *Rev. Mod. Phys.*, 54:235–268, Jan 1982.

[17] R J Baxter. Potts model at the critical temperature. *Journal of Physics C: Solid State Physics*, 6(23):L445–L448, nov 1973.

[18] M. Blume. Theory of the first-order magnetic phase change in uo_2. *Phys. Rev.*, 141:517–524, Jan 1966.

[19] On the possibility of first-order phase transitions in ising systems of triplet ions with zero-field splitting. *Physica*, 32(5):966 – 988, 1966.

[20] M. Blume, V. J. Emery, and Robert B. Griffiths. Ising model for the λ transition and phase separation in he^3-he^4 mixtures. *Phys. Rev. A*, 4:1071–1077, Sep 1971.

[21] The blume–capel model for spins s=1 and 3/2 in dimensions d=2 and 3. *Physica A: Statistical Mechanics and its Applications*, 507:22 – 66, 2018.

[22] Cláudio DaSilva, A. Caparica, and J. Plascak. Wang-landau monte carlo simulation of the blume-capel model. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 73:036702, 04 2006.

[23] N. D. Mermin and H. Wagner. Absence of ferromagnetism or antiferromagnetism in one- or two-dimensional isotropic heisenberg models. *Phys. Rev. Lett.*, 17:1133–1136, Nov 1966.

[24] V. L. Berezinsky. Destruction of Long-range Order in One-dimensional and Two-dimensional Systems Possessing a Continuous Symmetry Group. II. Quantum Systems. *Sov. Phys. JETP*, 34(3):610, 1972.

[25] J M Kosterlitz and D J Thouless. Ordering, metastability and phase transitions in two-dimensional systems. *Journal of Physics C: Solid State Physics*, 6(7):1181–1203, apr 1973.

[26] J M Kosterlitz. The critical properties of the two-dimensional xy model. *Journal of Physics C: Solid State Physics*, 7(6):1046–1060, mar 1974.

[27] Raghav G Jha. Critical analysis of two-dimensional classical xy model. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(8):083203, 2020.

[28] Brian David Josephson. Possible new effects in superconductive tunnelling. *Physics letters*, 1(7):251–253, 1962.

[29] Laurence Jacobs, Jorge V José, MA Novotny, and AM Goldman. New coherent states in periodic arrays of ultrasmall josephson junctions. *Physical Review B*, 38(7):4562, 1988.

[30] Maciej M. Maśka and Nandini Trivedi. Temperature-driven bcs-bec crossover and cooper-paired metallic phase in coupled boson-fermion systems. *Phys. Rev. B*, 102:144506, Oct 2020.

[31] R. Micnas, J. Ranninger, and S. Robaszkiewicz. Superconductivity in narrow-band systems with local nonretarded attractive interactions. *Rev. Mod. Phys.*, 62:113–171, Jan 1990.

[32] B. M. Kessler, Ç. Ö. Girit, A. Zettl, and V. Bouchiat. Tunable superconducting phase transition in metal-decorated graphene sheets. *Phys. Rev. Lett.*, 104:047001, Jan 2010.

[33] LM Falicov and JC Kimball. Simple model for semiconductor-metal transitions: Sm b 6 and transition-metal oxides. *Physical Review Letters*, 22(19):997, 1969.

[34] Christian Gruber and Daniel Ueltschi. The falicov-kimball model. 2005.

[35] John Hubbard. Electron correlations in narrow energy bands. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 276(1365):238–257, 1963.

[36] Tom Kennedy and Elliott H Lieb. An itinerant electron model with crystalline or magnetic long range order. *Physica A: Statistical Mechanics and its Applications*, 138(1-2):320–358, 1986.

[37] Elliott H Lieb. A model for crystallization: A variation on the hubbard model. *Physica A: Statistical Mechanics and its Applications*, 140(1-2):240–250, 1986.

[38] JK Freericks and LM Falicov. Two-state one-dimensional spinless fermi gas. *Physical review B*, 41(4):2163, 1990.

[39] JK Freericks. Spinless falicov-kimball model (annealed binary alloy) from large to small dimensions. *Physical Review B*, 47(15):9263, 1993.

[40] U Brandt and C Mielsch. Free energy of the falicov-kimball model in large dimensions. *Zeitschrift für Physik B Condensed Matter*, 82(1):37–41, 1991.

[41] Maciej M Maśka and Katarzyna Czajka. Thermodynamics of the two-dimensional falicov-kimball model: A classical monte carlo study. *Physical Review B*, 74(3):035109, 2006.

[42] Murty S. S. Challa, D. P. Landau, and K. Binder. Finite-size effects at temperature-driven first-order transitions. *Phys. Rev. B*, 34:1841–1852, Aug 1986.

[43] Bousquet O. and Rätsch G. *Advanced Lectures on Machine Learning: ML Summer Schools 2003*. Springer Berlin Heidelberg, 2003.

[44] Barry A. Cipra. An introduction to the ising model. *American Mathematical Monthly*, 94:937–959, 1987.

[45] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[46] Kurt Binder. Applications of monte carlo methods to statistical physics. *Reports on Progress in Physics*, 60:487–559, 1997.

[47] Ulli Wolff. Collective monte carlo updating for spin systems. *Phys. Rev. Lett.*, 62:361–364, Jan 1989.

[48] Robert H. Swendsen and Jian-Sheng Wang. Nonuniversal critical dynamics in monte carlo simulations. *Phys. Rev. Lett.*, 58:86–88, Jan 1987.

[49] NV Prokof'Ev, BV Svistunov, and Igor S Tupitsyn. "worm" algorithm in quantum monte carlo simulations. *Physics Letters A*, 238(4-5):253–257, 1998.

[50] Junwei Liu, Huitao Shen, Yang Qi, Zi Yang Meng, and Liang Fu. Self-learning monte carlo method and cumulative update in fermion systems. *Physical Review B*, 95(24):241104, 2017.

[51] Krzanowski W.J. *Principles of multivariate analysis*. Oxford University Press, 2008.

[52] Wenjian Hu, Rajiv R. P. Singh, and Richard T. Scalettar. Discovering phases, phase transitions, and crossovers through unsupervised machine learning: A critical examination. *Phys. Rev. E*, 95:062122, Jun 2017.

[53] Sebastian J. Wetzel. Unsupervised learning of phase transitions: From principal component analysis to variational autoencoders. *Phys. Rev. E*, 96:022140, Aug 2017.

[54] Bishop C. *Pattern Recognition and Machine Learning*. Springer, 2006.

[55] Andriy Burkov. *The hundred-page machine learning book*. 2019. OCLC: 1089445188.

[56] Lei Wang. Discovering phase transitions with unsupervised learning. *Phys. Rev. B*, 94:195105, Nov 2016.

[57] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, 1992.

[58] Pedro Ponte and Roger G. Melko. Kernel methods for interpretable machine learning of order parameters. *Phys. Rev. B*, 96:205146, Nov 2017.

[59] 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.

[60] Ethem Alpaydin. *Introduction to Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, 3 edition, 2014.

[61] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

[62] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[63] Juan Carrasquilla and Roger Melko. Machine learning phases of matter. *Nature Physics*, 13, 05 2016.

[64] John B. Kogut. An introduction to lattice gauge theory and spin systems. *Rev. Mod. Phys.*, 51:659–713, Oct 1979.

[65] Dong-Ling Deng, Xiaopeng Li, and S. Das Sarma. Machine learning topological states. *Phys. Rev. B*, 96:195145, Nov 2017.

[66] Peter Broecker, Juan Carrasquilla, Roger G Melko, and Simon Trebst. Machine learning quantum phases of matter beyond the fermion sign problem. *Scientific reports*, 7(1):1–10, 2017.

[67] Yi Zhang, Roger G. Melko, and Eun-Ah Kim. Machine learning Γ_2 quantum spin liquids with quasiparticle statistics. *Phys. Rev. B*, 96:245119, Dec 2017.

[68] Joaquin F Rodriguez-Nieva and Mathias S Scheurer. Identifying topological order through unsupervised machine learning. *Nature Physics*, 15(8):790–795, 2019.

[69] 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.

[70] Monika Richter-Laskowska, H Khan, N Trivedi, and Maciej M Maśka. A machine learning approach to the berezinskii-kosterlitz-thouless transition in classical and quantum models. *Condensed Matter Physics*, 3, 2018.

[71] Wanzhou Zhang, Jiayu Liu, and Tzu-Chieh Wei. Machine learning of phase transitions in the percolation and in xy model. *Physical Review E*, 99(3), mar 2019.

[72] Kenta Shiina, Hiroyuki Mori, Yutaka Okabe, and Hwee Kuan Lee. Machine-learning studies on spin models. *Scientific reports*, 10(1):1–6, 2020.

[73] 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.

[74] Jordan Venderley, Vedika Khemani, and Eun-Ah Kim. Machine learning out-of-equilibrium phases of matter. *Phys. Rev. Lett.*, 120:257204, Jun 2018.

[75] Pengfei Zhang, Huitao Shen, and Hui Zhai. Machine learning topological invariants with neural networks. *Phys. Rev. Lett.*, 120:066401, Feb 2018.

[76] Xiao Liang, Wen-Yuan Liu, Pei-Ze Lin, Guang-Can Guo, Yong-Sheng Zhang, and Lixin He. Solving frustrated quantum many-particle models with convolutional neural networks. *Physical Review B*, 98(10):104426, 2018.

[77] Daniel Levy, Matthew D Hoffman, and Jascha Sohl-Dickstein. Generalizing hamiltonian monte carlo with neural networks. *arXiv preprint arXiv:1711.09268*, 2017.

[78] Stephen Whitelam, Viktor Selin, Ian Benlolo, Corneel Casert, and Isaac Tamblyn. Training neural networks using metropolis monte carlo and an adaptive variant. *Machine Learning: Science and Technology*, 3(4):045026, 2022.

[79] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[80] Mark A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37(2):233–243, 1991.

[81] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[82] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[83] Wanzhou Zhang, Jiayu Liu, and Tzu-Chieh Wei. Machine learning of phase transitions in the percolation and xy models. *Phys. Rev. E*, 99:032142, Mar 2019.

[84] Li Huang and Lei Wang. Accelerated monte carlo simulations with restricted boltzmann machines. *Phys. Rev. B*, 95:035105, Jan 2017.

[85] Roger G Melko, Giuseppe Carleo, Juan Carrasquilla, and J Ignacio Cirac. Restricted boltzmann machines in quantum physics. *Nature Physics*, 15(9):887–892, 2019.

[86] Giacomo Torlai, Brian Timar, Evert P. L. van Nieuwenburg, Harry Levine, Ahmed Omran, Alexander Keesling, Hannes Bernien, Markus Greiner, Vladan Vuletić, Mikhail D. Lukin, Roger G. Melko, and Manuel Endres. Integrating neural networks with a quantum simulator for state reconstruction. *Phys. Rev. Lett.*, 123:230504, Dec 2019.

[87] Anna Golubeva and Roger G Melko. Pruning a restricted boltzmann machine for quantum state reconstruction. *Physical Review B*, 105(12):125124, 2022.

[88] Giuseppe Carleo and Matthias Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017.

[89] Eamonn Keogh, Kaushik Chakrabarti, Michael Pazzani, and Sharad Mehrotra. Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and information Systems*, 3(3):263–286, 2001.

[90] Jessica Lin, Eamonn Keogh, Li Wei, and Stefano Lonardi. Experiencing sax: a novel symbolic representation of time series. *Data Mining and knowledge discovery*, 15(2):107–144, 2007.

[91] Donald J Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, USA:, 1994.

[92] Kiyoung Yang and Cyrus Shahabi. An efficient k nearest neighbor search for multivariate time series. *Information and Computation*, 205(1):65–98, 2007. Special Issue: TIME 2005.

[93] Wanpracha Art Chaovalitwongse, Ya-Ju Fan, and Rajesh C Sachdeo. On the time series k-nearest neighbor classification of abnormal brain activity. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 37(6):1005–1016, 2007.

[94] Essam Houssein, Moataz Kilany, and Aboul Ella Hassanien. Ecg signals classification: a review. *International Journal of Medical Engineering and Informatics*, 5:376–396, 01 2017.

[95] Zhifeng Xu, Jiyin Cao, Gang Zhang, Xuyong Chen, and Yushun Wu. Active learning accelerated monte-carlo simulation based on the modified k-nearest neighbors algorithm and its application to reliability estimations. *Defence Technology*, 2022.

[96] Josif Grabocka, Nicolas Schilling, Martin Wistuba, and Lars Schmidt-Thieme. Learning time-series shapelets. pages 392–401. Association for Computing Machinery, 2014.

[97] Yi Yang, Qilin Deng, Furao Shen, Jinxi Zhao, and Chaomin Luo. A shapelet learning method for time series classification. In *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 423–430. IEEE, 2016.

[98] Bradley A. P. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30:1145–59, 1997.

[99] Michael E. Fisher, Michael N. Barber, and David Jasnow. Helicity modulus, superfluidity, and scaling in isotropic systems. *Phys. Rev. A*, 8:1111–1124, Aug 1973.

[100] David R. Nelson and J. M. Kosterlitz. Universal jump in the superfluid density of two-dimensional superfluids. *Phys. Rev. Lett.*, 39:1201–1205, Nov 1977.

[101] Norbert Schultka and Efstratios Manousakis. Finite-size scaling in two-dimensional superfluids. *Phys. Rev. B*, 49:12071–12077, May 1994.

[102] Yusuke Tomita and Yutaka Okabe. Probability-changing cluster algorithm for two-dimensional XY and clock models. *Phys. Rev. B*, 65:184405, Apr 2002.

[103] Petter Minnhagen and Mats Wallin. New phase diagram for the two-dimensional coulomb gas. *Phys. Rev. B*, 36:5620–5623, Oct 1987.

[104] Petter Minnhagen. Nonuniversal jumps and the kosterlitz-thouless transition. *Phys. Rev. Lett.*, 54:2351–2354, May 1985.

[105] Martin Hasenbusch. The two-dimensional xy model at the transition temperature: a high-precision monte carlo study. 38(26):5869, jun 2005.

[106] Martin Hasenbusch. The binder cumulant at the kosterlitz–thouless transition. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(08):P08003, aug 2008.

[107] Yukihiro Komura and Yutaka Okabe. Large-scale monte carlo simulation of two-dimensional classical xy model using multiple gpus. *Journal of the Physical Society of Japan*, 81(11):113001, 2012.

[108] Yun-Da Hsieh, Ying-Jer Kao, and Anders W Sandvik. Finite-size scaling method for the berezinskii–kosterlitz–thouless transition. *Journal of Statistical Mechanics: Theory and Experiment*, 2013(09):P09001, 2013.

[109] Akinori Tanaka and Akio Tomiya. Detection of phase transition via convolutional neural networks. *Journal of the Physical Society of Japan*, 86(6):063001, 2017.

[110] E. van Nieuwenburg, Y. Liu, and S. Huber. Learning phase transitions by confusion. *Nature Phys*, 13:435–439, 2017.

[111] Benno S Rem, Niklas Käming, Matthias Tarnowski, Luca Asteria, Nick Fläschner, Christoph Becker, Klaus Sengstock, and Christof Weitenberg. Identifying quantum phase transitions using artificial neural networks on experimental data. *Nature Physics*, 15(9):917–920, 2019.

[112] Patrick Huembeli, Alexandre Dauphin, and Peter Wittek. Identifying quantum phase transitions with adversarial neural networks. *Phys. Rev. B*, 97:134109, Apr 2018.

[113] Wanzhou Zhang, Jiayu Liu, and Tzu-Chieh Wei. Machine learning of phase transitions in the percolation and xy models. *Phys. Rev. E*, 99:032142, Mar 2019.

[114] Yusuke Miyajima, Yusuke Murata, Yasuhiro Tanaka, and Masahito Mochizuki. Machine learning detection of berezinskii-kosterlitz-thouless transitions in q-state clock models. *Phys. Rev. B*, 104:075114, Aug 2021.

[115] Thineswaran Gunasegaran and Yu-N Cheah. Evolutionary cross validation. In *2017 8th International Conference on Information Technology (ICIT)*, pages 89–95, 2017.

[116] Zhong Wang, Xiao-Liang Qi, and Shou-Cheng Zhang. Topological order parameters for interacting topological insulators. *Phys. Rev. Lett.*, 105:256803, Dec 2010.

[117] C. L. Kane and E. J. Mele. Z_2 topological order and the quantum spin hall effect. *Phys. Rev. Lett.*, 95:146802, Sep 2005.

[118] G. Parisi. Infinite number of order parameters for spin-glasses. *Phys. Rev. Lett.*, 43:1754–1756, Dec 1979.

[119] Hirohito Kiwata. Deriving the order parameters of a spin-glass model using principal component analysis. *Physical Review E*, 99(6):063304, 2019.

[120] Ce Wang and Hui Zhai. Machine learning of frustrated classical spin models. i. principal component analysis. *Physical Review B*, 96(14):144432, 2017.

[121] Wenjian Hu, Rajiv R. P. Singh, and Richard T. Scalettar. Discovering phases, phase transitions, and crossovers through unsupervised machine learning: A critical examination. *Phys. Rev. E*, 95:062122, Jun 2017.

[122] J.A. Plascak and P.H.L. Martins. Probability distribution function of the order parameter: Mixing fields and universality. *Computer Physics Communications*, 184(2):259–269, 2013.

[123] Katharina Vollmayr, Joseph D Reger, Manfred Scheucher, and Kurt Binder. Finite size effects at thermally-driven first order phase transitions: A phenomenological theory of the order parameter distribution. *Zeitschrift für Physik B Condensed Matter*, 91(1):113–125, 1993.

[124] JD Reger, RN Bhatt, and AP Young. Monte carlo study of the order-parameter distribution in the four-dimensional ising spin glass. *Physical review letters*, 64(16):1859, 1990.

[125] Vladimir Privman. *Finite size scaling and numerical simulation of statistical systems*. World Scientific, 1990.

[126] Kurt Binder. Finite size scaling analysis of ising model block distribution functions. *Zeitschrift für Physik B Condensed Matter*, 43(2):119–140, 1981.

[127] C.-D. Li, D.-R. Tan, and F.-J. Jiang. Applications of neural networks to the studies of phase transitions of two-dimensional potts models. *Annals of Physics*, 391:312–331, 2018.

[128] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[129] Ketan Rajshekhar Shahapure and Charles Nicholas. Cluster quality analysis using silhouette score. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 747–748. IEEE, 2020.

[130] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.

[131] Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pages 410–420, 2007.

[132] D. P. Landau. Finite-size behavior of the ising square lattice. *Phys. Rev. B*, 13:2997–3011, Apr 1976.

[133] Michael E. Fisher and Michael N. Barber. Scaling theory for finite-size effects in the critical region. *Phys. Rev. Lett.*, 28:1516–1519, Jun 1972.

[134] Arthur E. Ferdinand and Michael E. Fisher. Bounded and inhomogeneous ising models. i. specific-heat anomaly of a finite lattice. *Phys. Rev.*, 185:832–846, Sep 1969.

[135] G. V. Bhanot and S. Sanielevici. Binder-challa-landau cumulant and lattice gauge theories: The order of the deconfinement transition. *Phys. Rev. D*, 40:3454–3462, Nov 1989.

[136] M. S. S. Challa, D. P. Landau, and K. Binder. Monte carlo studies of finite-size effects at first-order transitions. *Phase Transitions*, 24-26(1):343–369, 1990.

[137] P. Peczak and D. P. Landau. Monte carlo study of finite-size effects at a weakly first-order phase transition. *Phys. Rev. B*, 39:11932–11942, Jun 1989.

[138] K. Binder and D. P. Landau. Finite-size scaling at first-order phase transitions. *Phys. Rev. B*, 30:1477–1485, Aug 1984.

[139] K. Binder. Critical properties from monte carlo coarse graining and renormalization. *Phys. Rev. Lett.*, 47:693–696, Aug 1981.

[140] 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.

[141] I. Corte, S. Acevedo, M. Arlego, and C.A. Lamas. Exploring neural network training strategies to determine phase transitions in frustrated magnetic models. *Computational Materials Science*, 198:110702, 2021.

[142] Rui Wang, Yu-Gang Ma, R. Wada, Lie-Wen Chen, Wan-Bing He, Huan-Ling Liu, and Kai-Jia Sun. Nuclear liquid-gas phase transition with machine learning. *Phys. Rev. Research*, 2:043202, Nov 2020.

[143] Song Sub Lee and Beom Jun Kim. Confusion scheme in machine learning detects double phase transitions and quasi-long-range order. *Phys. Rev. E*, 99:043308, Apr 2019.

[144] Julian Arnold and Frank Schäfer. Replacing neural networks by optimal analytical predictors for the detection of phase transitions, 2022.

[145] Elmer V. H. Doggen, Frank Schindler, Konstantin S. Tikhonov, Alexander D. Mirlin, Titus Neupert, Dmitry G. Polyakov, and Igor V. Gornyi. Many-body localization and delocalization in large quantum chains. *Phys. Rev. B*, 98:174202, Nov 2018.

[146] Daria Zvyagintseva, Helgi Sigurdsson, Valerii K Kozin, Ivan Iorsh, Ivan A Shelykh, Vladimir Ulyantsev, and Oleksandr Kyriienko. Machine learning of phase transitions in nonlinear polariton lattices. *Communications Physics*, 5(1):1–10, 2022.

[147] NA Koritsky, AK Fedorov, SV Solov'yov, and AK Zvezdin. Learning phase transitions in the ferrimagnetic gdfeco alloy. In *2020 Science and Artificial Intelligence conference (SAI ence)*, pages 33–36. IEEE, 2020.

[148] 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.

[149] Bertil Hille. *Ion channels of excitable membranes*. Sunderland, Mass: Sinauer, 2001.

[150] Michael Ackerman and David Clapham. Ion channels — basic science and clinical disease. *The New England journal of medicine*, 336:1575–86, 06 1997.

[151] Heike Wulff, Neil A Castle, and Luis A Pardo. Voltage-gated potassium channels as therapeutic targets. *Nature reviews Drug discovery*, 8(12):982–1001, 2009.

[152] Monika Richter-Laskowska, Paulina Trybek, Domenico Vittorio Delfino, and Agata Wawrzkiewicz-Jałowiecka. Flavonoids as modulators of potassium channels. *International Journal of Molecular Sciences*, 24(2):1311, 2023.

[153] Ildiko Szabo and Mario Zoratti. Mitochondrial channels: ion fluxes and more. *Physiological reviews*, 94(2):519–608, 2014.

[154] Enrique Balderas, Jin Zhang, Enrico Stefani, and Ligia Toro. Mitochondrial bkca channel. *Frontiers in physiology*, 6:104, 2015.

[155] Adam Szewczyk, Wieslawa Jarmuszkiewicz, and Wolfram S Kunz. Mitochondrial potassium channels. *IUBMB life*, 61(2):134–143, 2009.

[156] Sandra Frankenreiter, Piotr Bednarczyk, Angelina Kniess, Nadja I Bork, Julia Straubinger, Piotr Koprowski, Antoni Wrzosek, Eva Mohr, Angela Logan, Michael P Murphy, et al. cgmp-elevating compounds and ischemic conditioning provide cardioprotection against ischemia and reperfusion injury via cardiomyocyte-specific bk channels. *Circulation*, 136(24):2337–2355, 2017.

[157] Anna Kicinska, Rafał Kampa, Jan Daniluk, Aleksandra Sek, Wieslawa Jarmuszkiewicz, Adam Szewczyk, and Piotr Bednarczyk. Regulation of the mitochondrial bkca channel by the citrus flavonoid naringenin as a potential means of preventing cell damage. *Molecules*, 25:3010, 06 2020.

[158] Reinhold Penner. A practical guide to patch clamping. In *Single-channel recording*, pages 3–30. Springer, 1995.

[159] Numan Celik, Fiona O'Brien, Sean Brennan, Richard D Rainbow, Caroline Dart, Yalin Zheng, Frans Coenen, and Richard Barrett-Jolley. Deep-channel uses deep neural networks to detect single-molecule events from patch-clamp data. *Communications biology*, 3(1):1–10, 2020.

[160] Ke Han, Miao Wang, Lei Zhang, Ying Wang, Mian Guo, Ming Zhao, Qian Zhao, Yu Zhang, Nianyin Zeng, and Chunyu Wang. Predicting ion channels genes and their types with machine learning techniques. *Frontiers in Genetics*, 10:399, 2019.

[161] Gustavo F Contreras, Alan Neely, Osvaldo Alvarez, Carlos Gonzalez, and Ramon Latorre. Modulation of bk channel voltage gating by different auxiliary β subunits. *Proceedings of the National Academy of Sciences*, 109(46):18991–18996, 2012.

[162] Q Li and J Yan. Modulation of bk channel function by auxiliary beta and gamma subunits. *International review of neurobiology*, 128:51–90, 2016.

[163] Monika Richter-Laskowska, Paulina Trybek, Piotr Bednarczyk, and Agata Wawrzkiewicz-Jałowiecka. Application of machine-learning methods to recognize mitobk channels from different cell types based on the experimental patch-clamp results. *International Journal of Molecular Sciences*, 22(2):840, 2021.

[164] Detlef Siemen, Christos Loupatatzis, Jiri Borecky, Erich Gulbins, and Florian Lang. Ca2+-activated k channel of the bk-type in the inner mitochondrial membrane of a human glioma cell line. *Biochemical and biophysical research communications*, 257(2):549–554, 1999.

[165] Agnieszka Walewska, Bogusz Kulawiak, Adam Szewczyk, and Piotr Koprowski. Mechanosensitivity of mitochondrial large-conductance calcium-activated potassium channels. *Biochimica et Biophysica Acta (BBA)-Bioenergetics*, 1859(9):797–805, 2018.

[166] Bartłomiej Augustynek, Piotr Koprowski, Daria Rotko, Wolfram S Kunz, Adam Szewczyk, and Bogusz Kulawiak. Mitochondrial bk channel openers cgs7181 and cgs7184 exhibit cytotoxic properties. *International Journal of Molecular Sciences*, 19(2):353, 2018.

[167] Anna Kicinska, Rafał P Kampa, Jan Daniluk, Aleksandra Sek, Wieslawa Jarmuszkiewicz, Adam Szewczyk, and Piotr Bednarczyk. Regulation of the mitochondrial bkca channel by the citrus flavonoid naringenin as a potential means of preventing cell damage. *Molecules*, 25(13):3010, 2020.

[168] Wenhong Xu, Yongge Liu, Sheng Wang, Todd McDonald, Jennifer E Van Eyk, Agnieszka Sidor, and Brian O'Rourke. Cytoprotective role of ca2+-activated k+ channels in the cardiac inner mitochondrial membrane. *Science*, 298(5595):1029–1033, 2002.

[169] Ana L González-Cota, Carmen Santana-Calvo, Rocío Servín-Vences, Gerardo Orta, and Enrique Balderas. Regulatory mechanisms of mitochondrial bkca channels. *Channels*, 15(1):424–437, 2021.

[170] Adam Szewczyk, Anna Kajma, Dominika Malinska, Antoni Wrzosek, Piotr Bednarczyk, Barbara Zabłocka, and Krzysztof Dołowy. Pharmacology of mitochondrial potassium channels: dark side of the field. *FEBS letters*, 584(10):2063–2069, 2010.

[171] Hung-Te Hsu, Yu-Ting Tseng, Yi-Ching Lo, and Sheng-Nan Wu. Ability of naringenin, a bioflavonoid, to activate m-type potassium current in motor neuron-like cells and to increase bkca-channel activity in hek293t cells transfected with α-hslo subunit. *BMC neuroscience*, 15(1):1–16, 2014.

[172] Simona Saponara, Lara Testai, D Iozzi, E Martinotti, Alma Martelli, S Chericoni, G Sgaragli, Fabio Fusi, and Vincenzo Calderone. (+/-)-naringenin as large conductance ca2+-activated k+ (bkca) channel opener in vascular smooth muscle cells. *British journal of pharmacology*, 149(8):1013–1021, 2006.

[173] Szymon Mercik, Karina Weron, and Zuzanna Siwy. Statistical analysis of ionic current fluctuations in membrane channels. *Phys. Rev. E*, 60:7343–7348, Dec 1999.

[174] Rushi Longadge and Snehalata Dongre. Class imbalance problem in data mining review. *arXiv preprint arXiv:1305.1707*, 2013.

[175] Anna Dawid, Patrick Huembeli, Michal Tomza, Maciej Lewenstein, and Alexandre Dauphin. 22(11):115001, nov 2020.

[176] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information, 2017.

[177] Monika Richter-Laskowska, Paulina Trybek, Piotr Bednarczyk, and Agata Wawrzkiewicz-Jałowiecka. To what extent naringenin binding and membrane depolarization shape mitobk channel gating—a machine learning approach. *PLOS Computational Biology*, 18(7):e1010315, 2022.

[178] Monika Richter-Laskowska, Marcin Kurpas, and Maciej Maśka. A learning by confusion approach to characterize phase transitions. *arXiv preprint arXiv:2206.15114*, 2022.

[179] Paulina Trybek, Ewelina Sobotnicka, Agata Wawrzkiewicz-Jałowiecka, Łukasz Machura, Daniel Feige, Aleksander Sobotnicki, and Monika Richter-Laskowska. A new method of identifying characteristic points in the impedance cardiography signal based on empirical mode decomposition. *Sensors*, 23(2):675, 2023.

[180] Ilona Karpiel, Monika Richter-Laskowska, Daniel Feige, Adam Gacek, and Aleksander Sobotnicki. An effective method of detecting characteristic points of impedance cardiogram verified in the clinical pilot study. *Sensors*, 22(24):9872, 2022.

[181] Bartosz Dziewit, Jacek Holeczek, Monika Richter, Sebastian Zajac, and Marek Zrałek. The discrete family symmetry as a possible solution to the flavour problem. *Physics of Atomic Nuclei*, 80:747–751, 2017.

[182] Bartosz Dziewit, Jacek Holeczek, Monika Richter, Sebastian Zajac, and Marek Zralek. Texture zeros in neutrino mass matrix. *Physics of Atomic Nuclei*, 80:353–357, 2017.

[183] Bartosz Dziewit, Jacek Holeczek, Monika Richter, Sebastian Zajac, and Marek Zrałek. The flavour problem and the family symmetry beyond the standard model. *Acta Physica Polonica Series B*, 46(11):2399, 2015.

[184] Piotr Chaber, Bartosz Dziewit, Jacek Holeczek, Monika Richter, Marek Zrałek, and Sebastian Zajac. Lepton masses and mixing in a two-higgs-doublet model. *Physical Review D*, 98(5):055007, 2018.

[185] Bartosz Dziewit, Monika Richter, and Jerzy Dajka. Distinguishing quantum states using time-traveling qubits in the presence of thermal environments. *Physical Review A*, 95(3):032136, 2017.

[186] Monika Richter, Bartosz Dziewit, and Jerzy Dajka. The quantum cheshire cat effect in the presence of decoherence. *Advances in Mathematical Physics*, 2018, 2018.

[187] Monika Richter-Laskowska, Marcin Łobejko, and Jerzy Dajka. Quantum contextuality of a single neutrino under interactions with matter. *New Journal of Physics*, 20(6):063040, 2018.

[188] Monika Richter, Bartosz Dziewit, and Jerzy Dajka. Leggett-garg k 3 quantity discriminates between dirac and majorana neutrinos. *Physical Review D*, 96(7):076008, 2017.

[189] Federico Girosi, Michael Jones, and Tomaso Poggio. Regularization Theory and Neural Networks Architectures. *Neural Computation*, 7(2):219–269, 03 1995.

[190] Francois Chollet et al. Keras, 2015.

[191] Andrew Ng. Feature selection, l 1 vs. l 2 regularization, and rotational invariance. *Proceedings of the Twenty-First International Conference on Machine Learning*, 09 2004.

[192] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[193] Jin Huang and C.X. Ling. Using auc and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 17(3):299–310, 2005.

LIST OF PUBLICATIONS

Publications being the basis of the dissertation

1. Richter–Laskowska, M., Trybek, P., Bednarczyk P. and Wawrzkiewicz–Jałowiecka A. 2022. To what extent naringenin binding and membrane depolarization shape mitoBK channel gating – a machine learning approach. PLOS Computational Biology, 18(7), e1010315.

2. Richter–Laskowska, M., Trybek, P., Bednarczyk P. and Wawrzkiewicz–Jałowiecka A. 2021. Application of Machine-Learning Methods to Recognize mitoBK Channels from Different Cell Types Based on the Experimental Patch-Clamp Results. International Journal of Molecular Sciences, 22(840)

3. Richter–Laskowska M., Khan H., Trivedi N. and Maśka M. 2018. A machine learning approach to the Berezinskii-Kosterlitz-Thouless transition in classical and quantum models. Condens. Matter Phys. vol. 21, No. 3, 33602

4. Richter–Laskowska, M., Kurpas, M. and Maśka M. 2022. A learning by confusion approach to characterize phase transitions, arXiv:2206.15114 (preprint)

Remaining publications

1. Richter-Laskowska, M., Trybek, P., Delfino, D. V., Wawrzkiewicz-Jał owiecka, A. (2023). Flavonoids as Modulators of Potassium Channels. International Journal of Molecular Sciences, 24(2), 1311.

2. Trybek, P., Sobotnicka, E., Wawrzkiewicz-Jałowiecka, A., Machura, Ł., Feige, D., Sobotnicki, A., and Richter-Laskowska, M. (2023). A New Method of Identifying Characteristic Points in the Impedance Cardiography Signal Based on Empirical Mode Decomposition. Sensors, 23(2), 675.

3. Karpiel, I., Richter-Laskowska, M., Feige, D., Gacek, A., and Sobotnicki, A. (2022). An Effective Method of Detecting Characteristic Points of Impedance Cardiogram Verified in the Clinical Pilot Study. Sensors, 22(24), 9872.

4. Dziewit B., Holeczek J., Richter M., Zajac S. and Zrałek M. Lepton masses and mixing in a two-Higgs-doublet model. 2018. Phys. Rev. D 98, 055007

5. Richter–Laskowska M., Łobejko M., Dajka J. 2018. Quantum contextuality of a single neutrino under interactions with matter. New J. Phys. 20 063040

6. Richter M., Dziewit B., Dajka J. 2018. The Quantum Cheshire Cat Effect in the Presence of Decoherence. Advances in Mathematical Physics 2018(307):1-8

7. Richter M., Dziewit B., Dajka J. 2017. Leggett-Garg K 3 quantity discriminates between Dirac and Majorana neutrinos. Phys. Rev. D 96, 076008

8. Dziewit B., Richter M., Dajka J. 2017. Distinguishing quantum states using time-traveling qubits in the presence of thermal environments. Phys. Rev. A 95, 032136

9. Dziewit B., Holeczek J., Richter M., Zajac S. and Zralek M. 2016. Texture zeros in neutrino mass matrix. Physics of Atomic Nuclei 80(2)

10. Dziewit B., Holeczek J., Richter M., Zajac S. and Zralek M. 2016. The discrete family symmetry as a possible solution to the flavour problem. Physics of Atomic Nuclei 80(4)

11. Dziewit B., Holeczek J., Richter M., Zajac S. and Zralek M. 2015. The Flavour Problem and the Family Symmetry Beyond the Standard Model. Acta Physica Polonica Series B 46(11):2399

Appendices

Appendix A

The estimation of the critical temperature with the neural network – computational details

Here, we present details regarding the neural network computations used to estimate the critical temperature in the Berezinskii–Kosterlitz–Thouless phase transition described in Sec. 4.

For all models under study, the dimension of generated MC configurations is set to 16×16. Each MC sample is subsequently transformed into a vector of sine and cosine functions (Fig. A.1). Therefore, the input of our simple feed–forward neural network is of dimension $2 \times 16 \times 16 = 512$. It is then followed by six fully–connected layers. The first five of them, composed of 512, 192, 64, 16, and 16 neurons, are activated with the rectified linear unit (ReLU) activation function. The last neuron activated by the sigmoid function returns a value between 0 and 1. It ensures a straightforward interpretation of the output as the probability of a configuration belonging to the high–temperature phase.

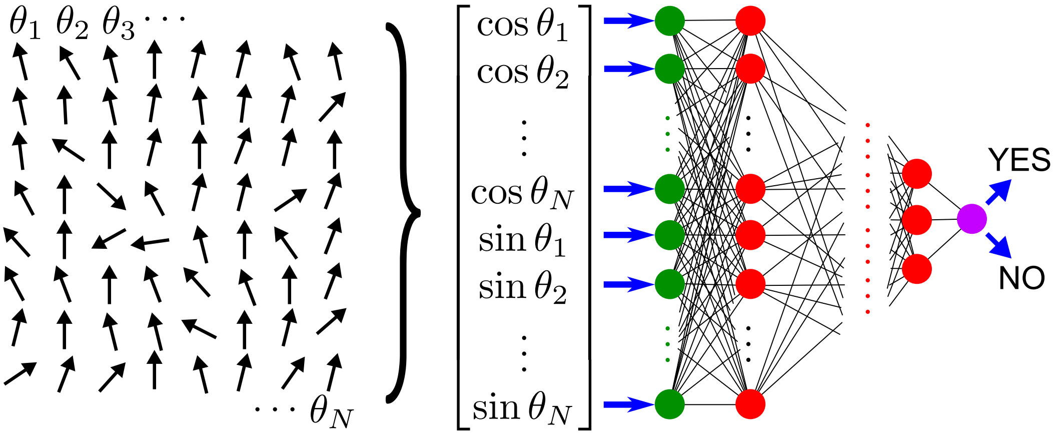


Fig. A.1 The scheme of preprocessing stage performed before feeding the raw spin configurations into a neural network. The configurations represented by the angles θ_i are transformed to the vector of trigonometric functions and such are analyzed by an algorithm deciding whether given configuration belongs to the high–temperature phase (YES) or to the low–temperature phase (NO).

The neural network is trained with the ADAM (*A method of stochastic*

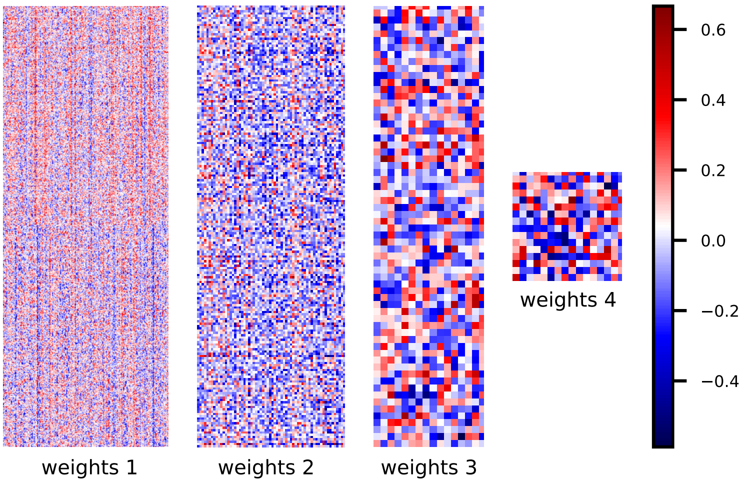


Fig. A.2 Illustration of weights learned during the process of training. The white, blue, and red colors correspond to zero, negative and positive weights, respectively. 'weights 1' connect the first hidden layer of dimension 512 with the next one of dimension 256. Similarly, 'weights 2' links the layer of size 256 to that of dimension 64, etc.

optimization) algorithm [61], which aims to minimize the loss function of the form:

$$L = -\sum_i \left[y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)\right] + \lambda \sum_i w_i^2, \tag{A.1}$$

where y_i denote the data labels and $\hat{y}_i$ are predictions of a neural network. The first part of Eq. A.1 stands simply for the binary cross–entropy function, while the second part of L is the L2 regularization term [189], where w_i are the weights used during training. λ stands for the regularization constant, which must be tuned with trial and error. In our case, the addition of this term is quite beneficial – it significantly improves the predictions' accuracy, makes the results more stable, and reduces overfitting.

Note that such a neural network structure gives us a good balance between the number of training epochs and the time needed for the algorithm's convergence. Despite a large number of neurons, most of them are activated during the training, contributing to the final output. Their exemplary values are shown in Fig. A.2.

All computations were performed with use of the `Python` language supported by the interface for the artificial neural network called `Keras` [190].

Appendix B

Calculation of $M(T'_C)$ quantity for the discontinuous phase transition

In this appendix, we present the calculations leading to the expression for the quantity $M(T'_C) = \int_{T_{min}}^{T_{max}} f(T'_C, T)dT$ obtained in Eq. (4.19). T_1 and T_2 indicate the boundary temperatures between which two distinct phases coexist. T_C and T'_C correspond to the actual and assumed critical temperatures, respectively.

This analysis is performed for several cases:

a) $\mathbf{T'_C = T_C}$

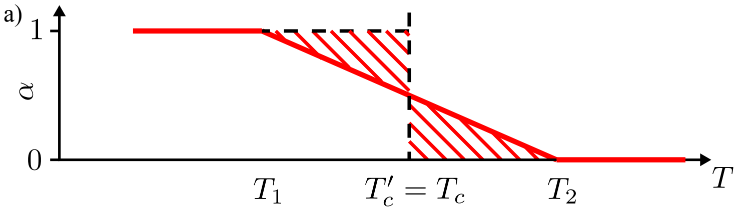


Fig. B.1 The proportion of mislabeled data α as the function of temperature assuming $T'_C = T_C$.

In this case, the integral $M(T'_C)$ can be calculated as the sum of two areas highlighted in Fig. B.3. Since the critical temperature T_C lies right in the middle between temperatures T_1 and T_2, these areas are identical. It is, therefore, sufficient to calculate one of them and multiply by two the overall result at the end. Denoting by $T_2 - T_1 = \Delta$ we obtain:

$$M(T'_C) = (T_C - T_1) \cdot \frac{\alpha(T'_C)}{2} = \frac{\Delta}{2} \cdot \frac{1}{2} = \frac{1}{4}\Delta,$$

where $\alpha(T'_C)$ was calculated from Eq. (4.18).

b) $\mathbf{T_1 < T'_C < T_C}$

Similarly to the previous case, the quantity $M(T'_C)$ can be calculated as the sum of two areas. However, this time they are not the same and must be evaluated

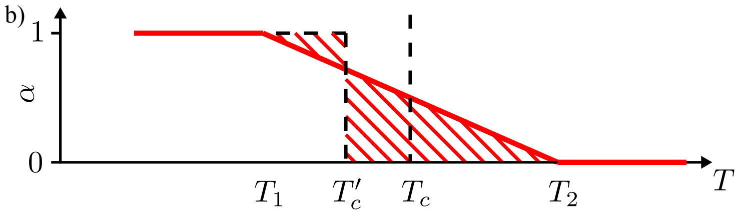


Fig. B.2 The proportion of mislabeled data α as the function of temperature assuming $T_1 < T'_C < T_C$.

separately. The value of $M(T'_C)$ then reads:

$$M(T'_C) = \frac{1}{2}\left[1-\alpha(T'_C)\right](T'_C - T_1) + \frac{1}{2}\alpha(T'_C)(T_2 - T'_C) = \frac{1}{2}\left[1-\alpha(T'_C)\right]\left(\frac{1}{2}\Delta - \tau\right) + \frac{1}{2}\alpha(T'_C)\left(\frac{1}{2}\Delta + \tau\right) = \frac{1}{4}\Delta + \alpha(T'_C)\tau - \frac{1}{2}\tau, \tag{B.1}$$

where $T'_C - T_C = \tau$. Making use of Eq. (4.18) and assuming $T_2 = T_C + \frac{1}{2}\Delta$ we get the following expression for $\alpha(T'_C)$:

$$\alpha(T'_C) = \frac{T_2 - T'_C}{T_2 - T_1} = \frac{T_2 - T'_C}{\Delta} = \frac{T_C + \frac{1}{2}\Delta - T'_C}{\Delta} = \frac{\tau + \frac{1}{2}\Delta}{\Delta}. \tag{B.2}$$

By inserting Eq. (B.2) into Eq. (B.1) we obtain:

$$M(T'_C) = \frac{1}{4}\Delta + \frac{\tau + \frac{1}{2}\Delta}{\Delta}\tau - \frac{1}{2}\tau = \frac{1}{4}\Delta + \frac{\tau^2}{2\Delta} = \frac{1}{4}\Delta + \frac{1}{\Delta}(T_C - T'_C)^2, \tag{B.3}$$

in agreement with Eq. (4.19) for $|T'_C - T_C| < \frac{1}{2}\Delta$.

c) $\mathbf{T'_C < T_1}$

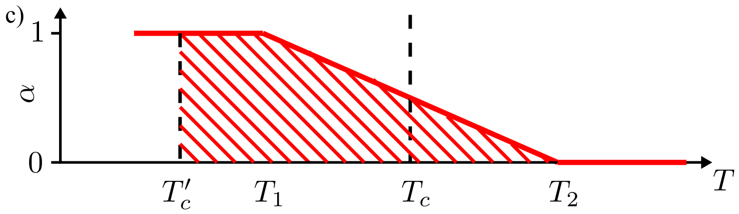


Fig. B.3 The proportion of mislabeled data α as the function of temperature assuming $T'_C < T_1$.

In this case the quantity $M(T'_C)$ can be calculated as the area of trapezium:

$$M(T'_C) = \frac{T_2 - T'_C + T_1 - T'_C}{2} = \frac{T_2 + T_1}{2} - T'_C = T_C - T'_C. \tag{B.4}$$

Appendix C

Computational details – a learning by confusion approach to characterize phase transitions

C.1 Monte Carlo simulations

The Monte Carlo configurations are generated using the following:

1. Classical Metropolis algorithm in the case of the BC model,
2. Wolff cluster method [47] in the case of the qP model,
3. Modified version of the classical Metropolis algorithm for the FK model.

Since the first two methods are standard and well described in the literature, we will focus only on the case of the FK model.

The simulation of the FK model is carried out in analogy to the simulation performed for an Ising model characterized by the following partition function $\mathcal{Z}$:

$$\mathcal{Z} = \sum_i e^{-\beta E_i}, \tag{C.1}$$

where the sum is taken over all possible states of the system. Since the number of all possible configurations is enormous, one cannot evaluate the partition function explicitly. One method to overcome this problem is to generate a Markov chain of states. In the Ising model, the proposition of a new state f is generated from the previous one i by a flip of the spin on a randomly chosen lattice site in the initial configuration. We then consider the energy difference ΔE between i and f:

$$\Delta E = E_f - E_i. \tag{C.2}$$

The new state is accepted with the transition probability :

$$P_{if} = e^{-\beta \Delta E}, \tag{C.3}$$

where $\beta = \frac{1}{k_B T}$.

The FK model involves, apart from classical also, itinerant quantum fermions. It implies that the above–described procedure cannot be straightforwardly applied to this model. The partition function is in this case more complicated and reads as:

$$\mathcal{Z} = \sum_{C} \mathrm{Tr}_{e} e^{-\beta[\mathcal{H}(C)-\mu N]}, \tag{C.4}$$

where C denotes all possible configurations, the trace is taken over the fermionic degrees of freedom, $\beta = \frac{1}{k_B T}$, μ is the chemical potential, and N denotes the occupation number. We can simplify the form of Eq. (C.4) by rewriting it as:

$$\mathcal{Z} = \sum_{C} \mathrm{Tr}_{e} e^{-\beta[\mathcal{H}(C)-\mu N]} = \sum_{C} e^{-\beta\mathcal{F}} = \sum_{C}\sum_{i} e^{-\beta E_i}, \tag{C.5}$$

where $\mathcal{F}$ stands for the free energy, while E_i denote all possible system's energies. Denoting by ε_i eigenenergies of the system and by M the overall number of possible energy levels occupied by N particles, the energies E_i can be expressed as:

$$E_i = \sum_{i=1}^{M} n_i(\varepsilon_i - \mu), \tag{C.6}$$

where n_i is the number of particles corresponding to the eigenenergy ε_i. Then Eq. (C.5) transforms to:

$$\mathcal{Z} = \sum_{C}\sum_{n_i} e^{-\beta\sum_{i=1}^{M} n_i(\varepsilon_i-\mu)} = \sum_{C}\prod_{i=1}^{M}\sum_{n_i} e^{-\beta n_i(\varepsilon_i-\mu)}. \tag{C.7}$$

Making use of the Pauli exclusion principle ($n_i \in \{0,1\}$), we present the Eq. (C.7) in the form:

$$\mathcal{Z} = \sum_{C}\prod_{i=1}^{M}\left[1 + e^{-\beta(\varepsilon_i-\mu)}\right]. \tag{C.8}$$

Then, making use of Eq. (C.8) it is straightforward to show that:

$$\mathcal{F} = -\frac{1}{\beta}\sum_{i=1}^{M}\ln\left[1 + e^{-\beta(\varepsilon_i-\mu)}\right]. \tag{C.9}$$

Having form of the free energy we can evaluate the partition function $\mathcal{Z} = \sum_C e^{-\beta\mathcal{F}}$ which is similar to that presented for the Ising model (Eq. C.3). Then a new state f in the FK system proposed by a change in the position of one ion in the preceding configuration i is accepted with probability $P_{if} = e^{-\beta\mathcal{F}}$. The scheme illustrating this technique is illustrated in Fig. C.1.

The main advantage of this method is that it is simple and easy to implement. Moreover, it can be generalized also for other models involving classical and quantum

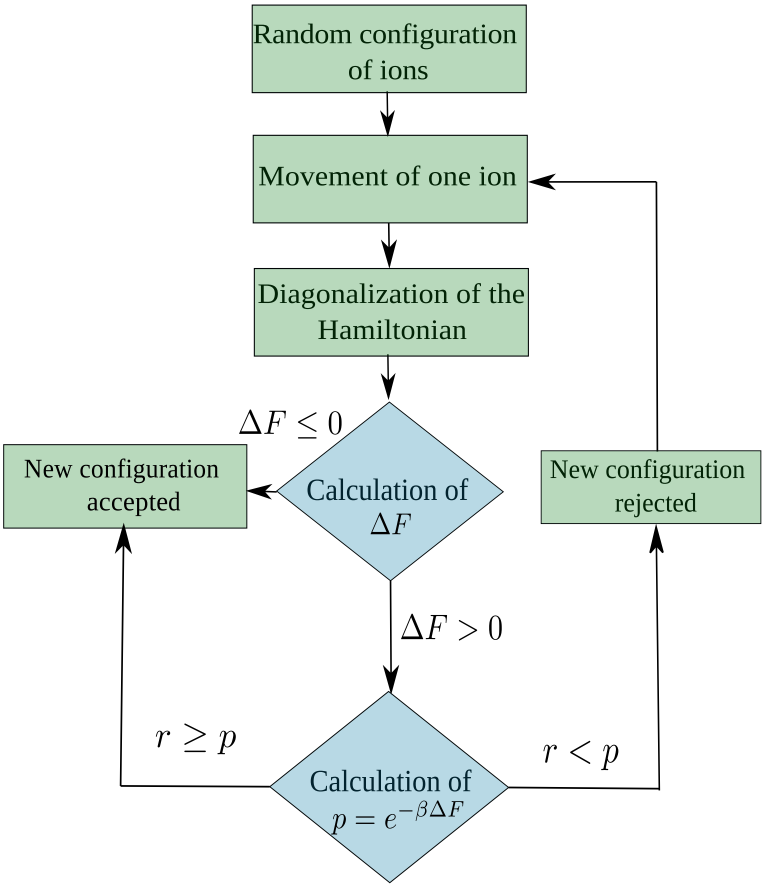

Random configuration of ions
Movement of one ion
Diagonalization of the Hamiltonian
$\Delta F \leq 0$
New configuration accepted
Calculation of ΔF
New configuration rejected
$\Delta F > 0$
$r \geq p$
Calculation of $p = e^{-\beta \Delta F}$
$r < p$

Fig. C.1 The scheme of a modified version of the Metropolis algorithm applied to systems with classical and fermionic degrees of freedom presented in the example of the Falicov–Kimball model. We start from the random configuration of ions and subsequently change the position of one arbitrarily selected ion in the lattice. The Hamiltonian is diagonalized afterward. This step allows calculating the free energy difference ΔF between the previous and new states. If $\Delta F \leq 0$, the new configuration is accepted and added to the Markov chain. If $\Delta F > 0$ one calculates the probability $e^{-\beta\Delta F}$ and compare it with a random number $r \in [0, 1]$. If $r >= p$, the proposed configuration is accepted. Otherwise, we reject the configuration, propose a new state, and repeat the whole procedure.

degrees of freedom. One of the examples of such a system is the fermion–phase model introduced in Sec. 2.5. Here, the main difference in the simulations (apart from another form of the Hamiltonian) lies in the way a new state is proposed – instead of movement of an ion, we change a spin direction on one lattice site.

Despite its assets, the method generates slow MC simulations and requires enormous computational resources. It is due to the necessity of diagonalization of the Hamiltonian in each MC step. Therefore, the lattice size L for which we can perform simulations in a reasonable amount of time is strictly limited.

C.2 Training of a neural network

The Monte Carlo configurations corresponding to all three microscopic models studied in Sec. 4.3 (i.e., FK, qP and BC) allow only discrete values on their sites (either spin or occupation number). It enables the preprocessing of the input dataset in such a way as to obtain a more general representation of the configurations associated with different models.

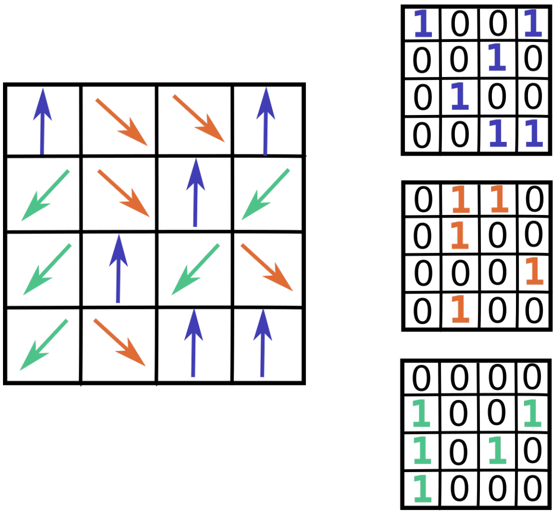


Fig. C.2 The transformation of raw input spin configurations. Each MC configuration of size $L \times L$ is transformed to q (number of possible spin orientations) new configurations in a way illustrated in the above picture.

Within the proposed scheme, the raw FK configurations are fed directly into the first layer of a neural network. The qP and BC Monte Carlo samples are modified in the way depicted in Fig. C.2 – every input configuration of dimension $L \times L$ is transformed to n configurations of the same size. The number n corresponds to q in the case of qP model and is always equal to 3 for the BC model (there are three possible values of spin: $-1, 0, 1$). Note that such a transformation extends the number of input channels in a neural network.

The neural network used in the numerical study of the LBC scheme (Sec.4.3) is composed of one convolutional layer with 64 filters of size 2×2. It is followed by the max pool layer, which reduces twice the dimensionality of feature maps obtained from convolution. The neurons are then flattened and connected with a dense layer. It is subsequently followed by the sigmoid function $\sigma(x)$. Both convolutional and first fully–connected layers are activated with the ReLU activation function.

We train the neural network with the Adam optimization algorithm [61] with a learning rate $\eta = 0.001$. Additionally, to avoid the overfitting problem, we apply to all layers L2 regularization term [191] along with the early–stopping method [189]. The chosen batch size is set to 256. This choice allows for speed computations that do not

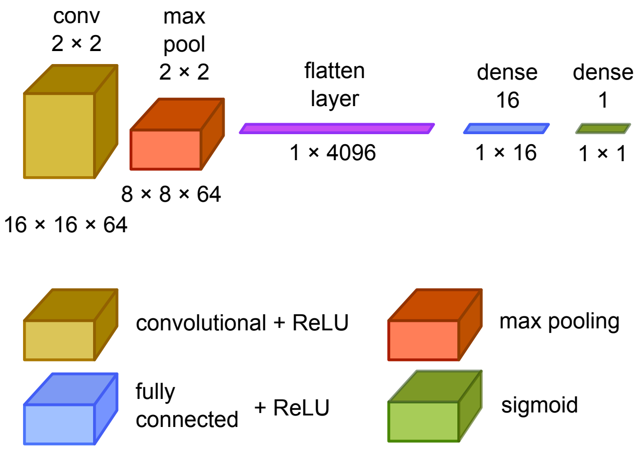


Fig. C.3 The architecture of a neural network used in the numerical study of the LBC scheme. It consists of one convolutional layer followed by the two–dimensional max–pooling layer and two fully–connected layers at the end. The convolutional layer and the first fully connected layer are activated with the ReLU activation function. The last layer is activated by the sigmoid.

excessively overload the GPU.

For every model, the training is performed on configurations corresponding to 100 consecutive temperatures. For each temperature, we generate 10000 statistically independent states with Monte Carlo (MC) simulations. Such obtained data is then split into training and testing datasets in proportion 80% and 20%, respectively.

All neural network computations and some data prepossessing steps were performed with the use of Python modules: `Keras` [190] and `Scikit-Learn` [192].

C.3 Justification of the usage of the AUC–ROC metrics

In the proposed variant of the LBC scheme, we evaluate the NN performance with the AUC–ROC metrics instead of the more popular accuracy. This step requires some justification. The method was originally designed to find the critical value of the parameter that separates two distinct phases [110]. Here, the motivation for applying the LBC scheme to our data is slightly different – we aim to determine the character of a phase transition. For this reason, our predictions should be more accurate. They should also contain information about the mean values of the outputs obtained from the last layer of a neural network and its distribution. Moreover, during the evaluation of the LBC scheme, we need to deal with the imbalanced dataset. Based on the objective criteria of *statistical consistency* and *discriminancy*, it has been proven that the AUC–ROC is better suited as a performance metric in such a case [193].

The power of AUC–ROC metrics can be illustrated in the following example. Let us assume that we have to deal with the binary classification problem, i.e., we need to find a model which can discriminate between two classes. Let us then say that two samples in our dataset are classified positively, but the output of a neural network for one sample indicates the value 0.53, while for the second one, 0.9. It is clear that it is much easier for an algorithm to classify the second sample. However, this information is not provided in the accuracy.

Regarding the research problem presented in Sec. 4.3, usage of the AUC–ROC metrics allows one to grasp significant differences between continuous and discontinuous phase transitions. Such discrepancies are illustrated in Figs. C.4 and C.5 on the example of the FK model.

As one can see, in the case of discontinuous phase transition (Fig. C.4), the output values are perfectly separated for $T_C' < T_C$ and $T_C' \approx T_C$. When $T_C' > T_C$, we observe that the distribution of outputs is more flattened for the high–temperature phase. The situation is quite different for the continuous phase transition (Fig. C.5), for which the distribution is similar in all three cases.

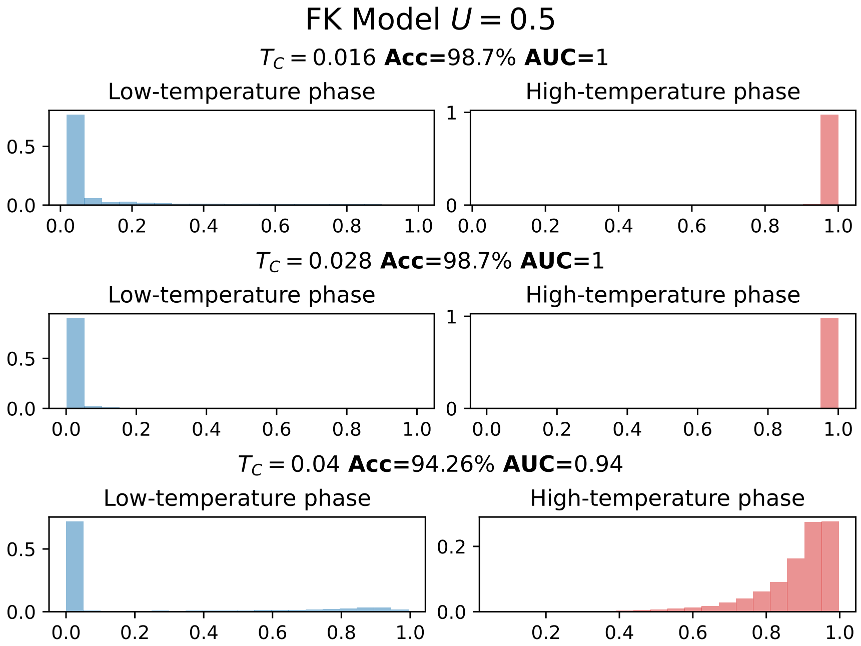


Fig. C.4 The neural network output's distributions obtained as a result of the evaluation of the LBC scheme for three different values of T'_C: $T'_C < T_C$, $T'_C \approx T_C$, $T'_C > T_C$ in the case of discontinuous phase transition.

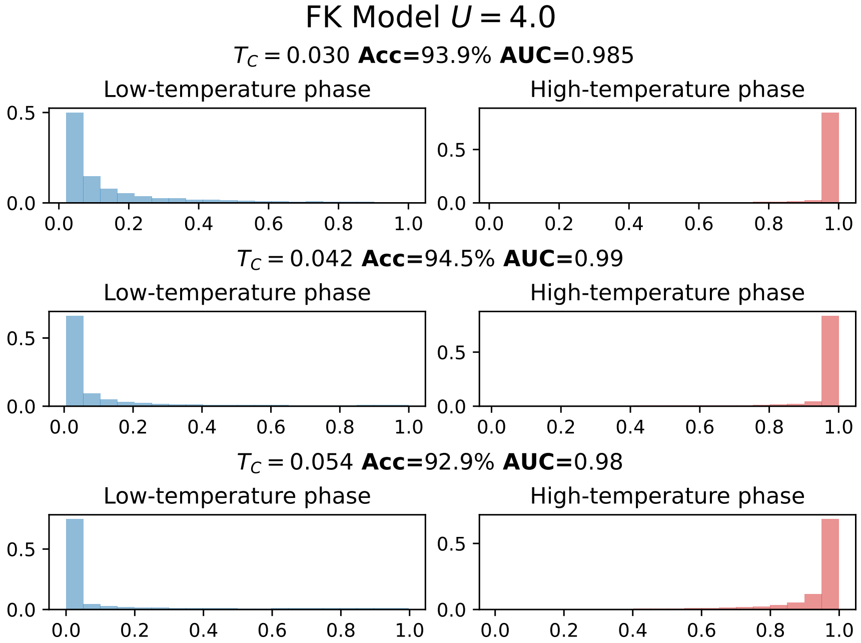


Fig. C.5 The neural network output's distributions obtained as a result of the evaluation of the LBC scheme for three different values of T'_C: $T'_C < T_C$, $T'_C \approx T_C$, $T'_C > T_C$ in the case of continuous phase transition.

Appendix D

Finite–size scaling of the LBC scheme for the FK model

The studies presented in Sec. 4.3 apply only to one lattice size $L = 16$. To confirm that the differences in the LBC curve obtained for models undergoing first– and second–order phase transition, are maintained in the thermodynamic limit, we performed additional simulations for $L \in \{10, 12, 20, 26\}$. The results are illustrated in Fig. D.1.

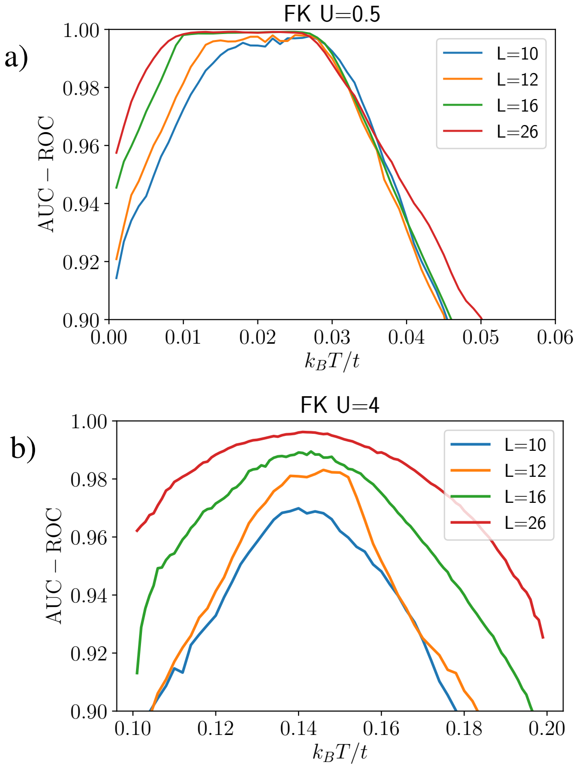


Fig. D.1 The AUC–ROC curves versus temperature for the FK model $U = 0.5$ (a) and $U = 4$ (b) and different lattice sizes L.

TThe main observation coming from that picture is that the plateau emerging in the AUC–ROC curve becomes wider with increasing lattice size (Fig. D.1a)). Moreover, the maximum performance of the algorithm converges to one for larger systems exhibiting a continuous phase transition (Fig. D.1b)). Finite–size effects are responsible for such a behavior of the AUC–ROC function. When the size of a system increases, peaks in specific heat and magnetic susceptibility become narrower, and so does the

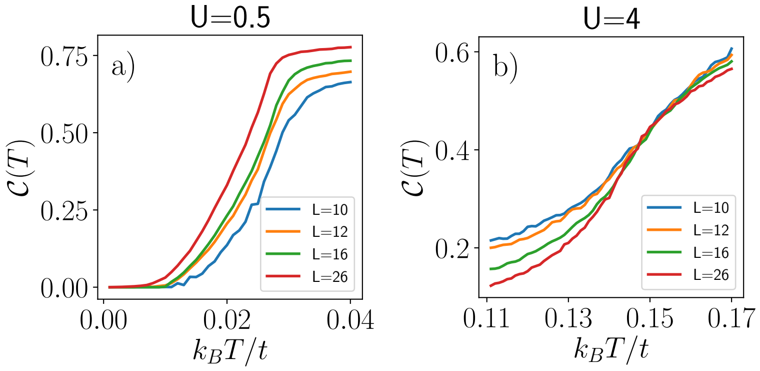


Fig. D.2 The concentration of defects $\mathcal{C}(T)$ as the function of temperature for the FK model simulated on different lattice sizes L and characterized by different potentials $U = 0.5$ (a) and $U = 4$ (b).

region of critical fluctuations. As a result, the function of the defect concentration $\mathcal{C}(T)$ becomes steeper around T_C. Consequently, it is easier for the algorithm to discriminate between configurations and the accuracy of neural network predictions increases.

To verify whether the change in the neural network performance that occurs for larger systems is associated with the alternations in the $\mathcal{C}(T)$ characteristics, we depict Fig. D.2. One can see a clear correspondence of these results to those presented in Fig. D.1. For $U = 0.5$ the region of the fastest growth in $\mathcal{C}(T)$ is shifted towards lower temperatures, leading to a widening of the plateau. For $U = 4$ $\mathcal{C}(T)$ becomes steeper around T_c, which increases the neural network performance.