# Dissertation

*submitted to the*

Combined Faculties of the Natural Sciences and Mathematics
*of the* Ruperto-Carola-University of Heidelberg, Germany

*for the degree of*

Doctor of Natural Sciences

*Put forward by*

Julian M. Urban

*born in:* Moosburg a.d. Isar, Germany
Oral examination: June 28th, 2022

# Breaking Through Computational Barriers In Lattice QCD With Artificial Intelligence

Referees: Prof. Dr. Jan M. Pawlowski
Prof. Dr. Thomas Gasenzer

## Breaking Through Computational Barriers In Lattice QCD With Artificial Intelligence

Progress in answering some of the most interesting open questions about the nature of reality is currently stalled by hard computational barriers. Research into artificial intelligence may provide solutions to these challenges. In this thesis, the applicability of modern machine learning algorithms to three longstanding problems in lattice quantum chromodynamics is investigated. First, normalizing flow architectures for the generative neural sampling of lattice field theories with dynamical fermions are developed and demonstrated to solve topological freezing in the Schwinger model at criticality. Flows are then applied to the density-of-states approach to complex action problems, showing that the Lee-Yang zeroes of the partition function of a scalar field theory with an imaginary external field can be successfully located. Finally, the problem of extracting real-time physics from imaginary-time data via spectral reconstruction is approached from the perspective of probabilistic inverse theory with Gaussian processes to compute ghost and gluon spectral functions. Future research directions are outlined for the application of the present work to state-of-the-art lattice and phenomenological calculations.

## Rechnerische Hürden der Gitter-QCD mittels künstlicher Intelligenz überwinden

Fortschritte in der Beantwortung einiger der interessantesten Fragen über das Universum werden derzeit von hohen rechnerischen Barrieren verhindert. Die Erforschung künstlicher Intelligenz mag dabei helfen, diese Herausforderungen zu meistern. In dieser Arbeit wird die Anwendbarkeit moderner Algorithmen des maschinellen Lernens auf drei langjährige Probleme der Gitterquantenchromodynamik untersucht. Zunächst werden normalisierende Flüsse zur neuronalen Stichprobenerzeugung von Gitterfeldtheorien mit dynamischen Fermionen entwickelt und zur Bewältigung des topologischen Einfrierens im Schwinger-Modell bei Kritikalität benutzt. Ferner werden Flüsse auf den Zustandsdichtenansatz für Vorzeichenprobleme angewendet. Es wird gezeigt, dass die Lee-Yang-Nullstellen der Zustandssumme einer skalaren Feldtheorie mit einem imaginären externen Feld erfolgreich lokalisiert werden können. Abschließend wird das Problem der Extraktion von Realzeitphysik aus Imaginärzeitdaten durch spektrale Rekonstruktion aus der Perspektive der probabilistischen Inverstheorie mit Gaußprozessen angegangen, um Geist- und Gluonspektralfunktionen zu berechnen. Zukünftige Forschungsrichtungen bezüglich der Anwendung der vorliegenden Arbeit auf moderne Gitterrechnungen und phänomenologische Ansätze werden skizziert.

# Contents

# 1 Introduction

Trying to explain the nature of reality has been part of the human experience for millennia, ever since our primate ancestors had the audacity to develop some level of higher brain function. However, significant progress in this endeavor has only been achieved more recently through the widespread application of the scientific method. In the 20th century, efforts to elucidate the fundamental properties of our universe finally culminated in the formulation of the theories that are still our most successful descriptions of physical phenomena at the largest and smallest scales. On the one hand, we have general relativity as the theory of gravity; on the other hand, the standard model of particle physics, which is the quantum field theory describing all known elementary particles and their electromagnetic, weak, and strong interactions.

After successful experimental confirmation of these theories to extremely high precision, the very end of fundamental theoretical physics seemed to lie within reach. However, as even the most optimistic contemporaries soon realized, unifying all four fundamental interactions into a consistent theoretical framework turned out to be much more difficult than anticipated. Moreover, even just within the individual theories themselves, there are still many open problems and unanswered questions that continue to puzzle physicists today and have opened a myriad of exciting new avenues to explore. Their resolution may not only complete our understanding of the currently known set of elementary particles, but also eventually lead to novel insights that could facilitate a unified description of gravity and quantum theory.

In parallel to these exciting developments in fundamental theoretical physics research, the past two decades have seen the rise of modern artificial intelligence (AI), with many areas of industry and public life now in the process of being utterly revolutionized. While initially the machine learning architectures achieving never-before-seen milestones in various settings—such as image classification or natural language processing—were as powerful as they were mysterious, advances in the understanding of these black boxes have led to their increasing application also in the natural sciences. The massive engineering efforts invested over many years are finally paying off, with meticulously crafted models impressively solving such grand challenges as the protein folding problem.

Perhaps unsurprisingly, AI has found its perfect match in the areas of scientific research that have already heavily relied on numerical approaches, and has become an integral part of many scientists' algorithmic toolbox. Its potential to overcome long-standing problems in these fields and take on a transformative role is rooted in a combination of ingenious model design and increasing levels of available computational power, facilitated by the ongoing deep learning revolution and the associated development of efficient, specialized accelerator hardware.

This thesis is written with the ambition that also some of the most difficult open questions about the nature of reality can be answered through the synergy of physical intuition, AI, and advances in high-performance computing. The present work is based on a series of papers [1–7] that document some of my own humble attempts at furthering our understanding of a tiny piece of the big puzzle.

## 1.1 Motivation

A component of the standard model that has turned out to be particularly difficult to understand is quantum chromodynamics (QCD)—the theory of the strong interaction—mainly because it cannot be treated perturbatively at low energies. One of the most interesting open problems in this context is the QCD phase diagram, specifically in the $T$-$\mu_B$ plane (temperature and baryon chemical potential). For example, it has been speculated that unlike the crossover encountered at low $\mu_B$, at higher baryon densities the phases are separated by a first-order transition that ends in a critical point of second order. A simplified sketch of the phase diagram is shown in Figure 1.1; however, most of the displayed features are conjectural or based on model predictions because ab-initio computations in the relevant region are extremely difficult. Moreover, accessing non-equilibrium properties of the theory exhibits even harder challenges. This only adds to the fact that even in the arguably simplest setting, namely in thermodynamic equilibrium at vanishing chemical potential, extracting experimentally testable predictions from the theory with first-principles calculations already requires enormous computational efforts. Utilizing recent advances in AI research to break through the barriers that prevent further progress in this area is the focus of the present work.

Before we delve further into the nature of these computational barriers, let us begin with a brief survey of what is currently possible with existing methods. In the past decades, two complementary frameworks for the non-perturbative treatment of strongly interacting matter have emerged, namely lattice field theory and functional methods. While both types of approaches play a role in this thesis, our main focus will be on the former. In this framework, path integrals are discretized on a Euclidean spacetime lattice, which enables access to equilibrium properties of the theory under consideration. This is achieved by switching from real to imaginary time via Wick rotation, yielding integrals over field configurations weighted by a Boltzmann factor with the Euclidean lattice action in the exponent. Expectation values are then evaluated stochastically by recasting the problem in terms of statistical sampling: ensembles of field configurations are generated via Markov chain Monte Carlo (MCMC) algorithms that typically explore the configuration space sequentially along some fictitious computer time. Physical observables can then be approximated in terms of ensemble averages. These calculations are by now well enough under control to fully convince us of the theory's correctness in the description of the strong nuclear force. As a result of these developments, lattice QCD has arguably entered the precision era of predicting standard model processes.

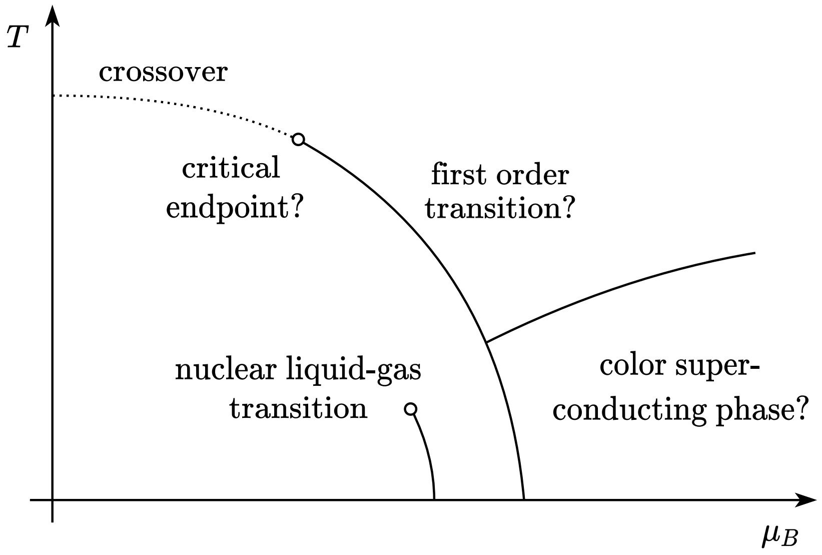


Figure 1.1: Sketch of the conjectured phase diagram of QCD in the $T$-$\mu_B$ plane.

However, further progress in understanding QCD from first principles is hampered by the aforementioned computational barriers. Even without considering finite densities or non-equilibrium processes, approaching the continuum limit of lattice QCD leads to severe slowing down of the commonly employed sampling algorithms. This is because traditional MCMC methods are based on local or diffusive updates. When the parameters of the system are tuned towards criticality, mapping out all relevant regions of configuration space in a reasonable time becomes increasingly difficult, even with the world's most powerful supercomputers. Specifically, state-of-the-art calculations based on the Hybrid/Hamiltonian Monte Carlo (HMC) algorithm are severely affected by so-called topological freezing. Such in-practice violations of ergodicity are highly problematic for the reliability of expectation values, since asymptotic correctness is only guaranteed for ensembles of statistically independent field configurations. To unlock larger volumes in these calculations—required e.g. for many problems in nuclear physics—exponentially improved sampling algorithms are urgently needed. In particular, first-principles calculations for nuclei with high mass numbers would significantly support the interpretations of several current and future intensity-frontier experiments that so far rely on computations in nuclear effective theory. For example, the determination of the neutrino mixing parameters and mass hierarchy from results of the DUNE long-baseline experiment requires axial form factors of argon. Likewise, scalar matrix elements in xenon are relevant for the dark matter direct detection search with XENONnT. Furthermore, in order to work out whether neutrinos are majorana particles, one would like to compute double-beta decay rates of heavy isotopes. In addition, defeating critical slowing down may also improve our understanding of universal properties of condensed matter systems where current calculations face similar problems.

Returning to the QCD phase diagram, further exploration of the yet uncharted waters is additionally prohibited by the infamous sign problem associated with finite chemical potential. Roughly speaking, at non-zero values of $\mu_B$ the imaginary part of the lattice action of the theory is non-vanishing, which leads to a breakdown of standard importance sampling. Naive attempts at circumventing such complex action problems often result in extremely unfavorable signal-to-noise ratios. The required computational effort to reach some fixed precision for physical predictions then grows exponentially with the system size. This renders many ab-initio computations infeasible outside of the region where the ratio $\mu_B/T$ is sufficiently small. Fully charting the phase diagram would lead to a better understanding of the properties of thermal nuclear matter and the physics of neutron stars, as well as cast light on possible exotic phases of quark matter. Apart from finite-density lattice QCD, research into approaches to complex action problems is also relevant in other contexts; e.g. for field theories with topological terms, spin- and mass-imbalanced cold atom systems, or similar problems in condensed matter theory.

Furthermore, as mentioned previously, the imaginary-time computations in Euclidean space described above are generally limited to equilibrium properties of the theory. For the real-time physics of non-equilibrium processes, the associated weighting factor in the path integral is a pure phase. Hence, this is in some sense the most difficult type of complex action problem imaginable. Progress in tackling such problems directly may eventually also help us to achieve genuine real-time computations and gain full access to the physics of dynamical QCD processes, whose theoretical treatment is essential for understanding many current and future experimental results. However, since for now we are restricted to computations in equilibrium, it makes sense to also explore approaches that extract real-time physics indirectly from imaginary-time data. This includes the numerical inversion of the spectral representation of Euclidean correlation functions in order to obtain real-time propagators, which is a heavily ill-conditioned inverse problem in need of regularization. Such calculations are relevant for the description of scattering processes as well as the hadronic resonance spectrum, and can provide first-principles QCD inputs for phenomenological approaches to transport processes in heavy-ion collisions.

In this thesis, I discuss several applications of modern machine learning methods to the computational problems described above, with an aim towards lattice QCD at scale. Specifically, deep neural networks are investigated for the generation and analysis of lattice field configurations. The main contributions here concern the mitigation of critical slowing down and the treatment of complex action problems via generative neural samplers with tractable probabilities. Furthermore, recent developments in probabilistic inverse theory with stochastic processes are utilized for the spectral reconstruction of correlation functions. As already stated in the beginning, in many cases such machine learning algorithms are black boxes if approached naively. Achieving a deeper understanding of their inner workings to facilitate novel insights and theoretical guarantees will be a recurring theme throughout this work.

## 1.2 Publications

The majority of the present work has been published in seven papers, in close collaboration with the many great people listed below. At the time of writing, five papers have been accepted as regular articles in peer-reviewed journals and two are currently being reviewed. Text overlap is indicated accordingly at the beginning of the respective chapters.

### Journal Articles

[1] Jan M. Pawlowski, Julian M. Urban, *Reducing Autocorrelation Times in Lattice Simulations with Generative Adversarial Networks*, Mach.Learn.Sci.Tech. 1 (2020) 045011, arXiv:1811.03533 [hep-lat]

[2] Lukas Kades, Jan M. Pawlowski, Alexander Rothkopf, Manuel Scherzer, Julian M. Urban, Sebastian J. Wetzel, Nicolas Wink, Felix P.G. Ziegler, *Spectral Reconstruction with Deep Neural Networks*, Phys.Rev.D 102 (2020) 9, 096001, arXiv:1905.04305 [physics.comp-ph]

[3] Stefan Blücher, Lukas Kades, Jan M. Pawlowski, Nils Strodthoff, Julian M. Urban, *Towards novel insights in lattice field theory with explainable machine learning*, Phys.Rev.D 101 (2020) 9, 094507, arXiv:2003.01504 [hep-lat]

[4] Michael S. Albergo, Denis Boyda, Kyle Cranmer, Dan C. Hackett, Gurtej Kanwar, Sébastien Racanière, Danilo J. Rezende, Phiala E. Shanahan, Julian M. Urban, *Flow-based sampling for fermionic lattice field theories*, Phys.Rev.D 104 (2021) 11, 114507, arXiv:2106.05934 [hep-lat]

[5] Jan Horak, Jan M. Pawlowski, José Rodríguez-Quintero, Jonas Turnwald, Julian M. Urban, Nicolas Wink, Savvas Zafeiropoulos, *Reconstructing QCD Spectral Functions with Gaussian Processes*, Phys.Rev.D 105 (2022) 3, 036014, arXiv:2107.13464 [hep-ph]

### Preprints

[6] Michael S. Albergo, Denis Boyda, Kyle Cranmer, Dan C. Hackett, Gurtej Kanwar, Sébastien Racanière, Danilo J. Rezende, Fernando Romero-López, Phiala E. Shanahan, Julian M. Urban, *Flow-based sampling in the lattice Schwinger model at criticality*, arXiv:2202.11712 [hep-lat]

[7] Jan M. Pawlowski, Julian M. Urban, *Flow-based density of states for complex actions*, arXiv:2203.01243 [hep-lat]

## 1.3 Outline

- In Chapter 2, the basic concepts behind lattice field theory and MCMC algorithms are briefly reviewed. Based on more general descriptions of the lattice formulations of scalar fields, fermions, and gauge fields, scalar Yukawa theory and the Schwinger model are introduced.

- In Chapter 3, we discuss the computational barriers in lattice QCD that are the primary focus of this work: topological freezing, complex action problems, and accessing real-time physics.

- Chapter 4 introduces the machine learning methods that form the basic building blocks of the architectures and frameworks used throughout this work: deep neural networks, normalizing flows, and Gaussian process regression.

- In Chapter 5, we take a first look at two machine learning applications to lattice field theory. First, a hybrid sampling algorithm employing a generative adversarial network is developed and applied to real, scalar $\phi^4$-theory. Then, it is demonstrated that interpretable AI techniques can be used to extract novel insights from lattice data in the context of scalar Yukawa theory.

- Chapter 6 concerns the development of normalizing flow architectures for the generative sampling of lattice field theories with dynamical fermions. After a proof-of-principle demonstration in scalar Yukawa theory, the chapter culminates in a study of flow-based sampling in the Schwinger model at criticality. It is shown that topological freezing can be successfully mitigated in a situation where standard MCMC algorithms fail to achieve sufficient ergodicity.

- In Chapter 7, the application of normalizing flows to the density-of-states approach to complex action problems is discussed in the context of scalar field theory with an imaginary external field. It is demonstrated that the density of states can be computed directly with this method and that the Lee-Yang zeroes of the partition function can be successfully located.

- In Chapter 8, spectral reconstruction is investigated in the framework of probabilistic inverse theory with Gaussian processes. Ghost and gluon spectral functions are computed from imaginary-time data supplied by lattice and functional computations in QCD and Yang-Mills theory.

- Chapter 9 provides a summary and outlook.

# 2 Lattice field theory

In this chapter, the basic concepts behind lattice field theory are introduced and we discuss the particular fields and models that are relevant for this work. With the exception of dynamical fermion fields—on whose treatment we put particular emphasis—in most places lengthy discussions about technical details are avoided and merely the most important aspects necessary to digest the following chapters are summarized. For pedagogical introductions to the subject where these technical discussions take place, see one of the standard textbooks [8–10]. The following text has some overlap with parts of [1, 3, 4, 6].

## 2.1 Introduction

Lattice field theory is among the most successful methods for regularizing and computing path integral expectation values in quantum field theory. In this approach, path integrals are evaluated numerically by discretizing the fields on a Euclidean spacetime lattice and formulating a stochastic process weighted by the lattice action [11]. The expectation value of some observable $\mathcal{O}$ can then be approximated as

$$\langle \mathcal{O} \rangle = \frac{1}{Z} \int \mathcal{D}\phi \, e^{-S(\phi)} \, \mathcal{O}(\phi) \simeq \frac{1}{N} \sum_{i=1}^{N} \mathcal{O}(\phi_i), \tag{2.1}$$

where $Z$ is the partition function, $S$ is the Euclidean lattice action, and $\{\phi_i\}$ is a set of $N$ samples of the lattice field degrees of freedom distributed as $p(\phi) = \exp[-S(\phi)]/Z$. Statistical uncertainties decrease as $1/\sqrt{N}$ as the estimate converges to the true value. The Euclidean lattice results can then be systematically related to the corresponding continuum Minkowski theory. This procedure enables the investigation of equilibrium properties in the theory of interest, and is a powerful and well-established method to study strongly coupled quantum field theories nonperturbatively. Key areas of application include fundamental interactions, most prominently QCD, as well as problems in condensed matter theory; see [12–19] for recent reviews.

Direct sampling schemes for high-dimensional lattice distributions are typically not known. Nevertheless, the distribution $p(\phi)$ can be sampled via MCMC methods with guaranteed asymptotic exactness under certain ergodicity and balance constraints [20]. Here, the 'gold standard' method is the Metropolis-Hastings algorithm [21]. It sequentially explores the configuration space through aperiodic updates ensuring ergodicity, combined with an accept/reject rule: starting from some initial field configuration $\phi$, a new candidate sample $\phi'$ is generated with some a priori

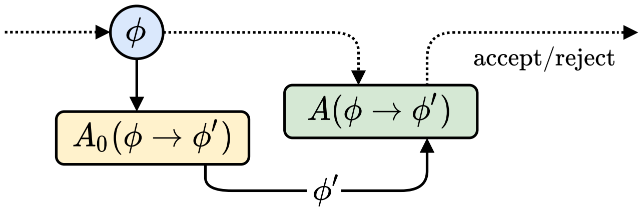


Figure 2.1: Illustration of the Metropolis-Hastings algorithm for MCMC sampling.

selection probability $A_0(\phi \to \phi')$. It is then accepted or rejected according to the acceptance probability

$$A(\phi \to \phi') = \min\left(1,\ \frac{A_0(\phi' \to \phi)\exp(-S(\phi'))}{A_0(\phi \to \phi')\exp(-S(\phi))}\right) \ , \tag{2.2}$$

see Figure 2.1 for an illustration. One usually considers a symmetric selection probability, i.e. $A_0(\phi' \to \phi) = A_0(\phi \to \phi')$. Equation (2.2) then simplifies to

$$A(\phi \to \phi') = \min(1,\ \exp(-\mathit{\Delta} S)) \ , \tag{2.3}$$

where $\mathit{\Delta} S = S(\phi') - S(\phi)$. After 'thermalization', i.e. a sufficient number of updates, the equilibrium distribution of this process is guaranteed to be the desired $p(\phi)$. In order to determine whether a Markov chain has been sufficiently equilibrated, one usually verifies the convergence of a set of representative observables.

Local or diffusive updating methods often exhibit strong correlations between subsequent samples in the Markov chain. Given a chain of $N$ measurements for some real-valued observable $X$, its autocorrelation function is defined as

$$\Gamma_X(\tau) = \frac{1}{N-\tau}\sum_{i=1}^{N-\tau} X_i X_{i+\tau} - \langle X \rangle^2 \ , \tag{2.4}$$

where $\tau$ denotes the number of Markov chain steps separating the pair of measurements considered. Typically, $\Gamma_X(\tau)$ decays exponentially,

$$\Gamma_X(\tau) \sim \exp\left(-\frac{\tau}{\tau_{\text{exp}}}\right) \ . \tag{2.5}$$

Here, $\tau_{\text{exp}}$ denotes the exponential autocorrelation time, which one expects to scale as a power of the correlation length, $\tau_{\text{exp}} \sim \xi^z$. The dynamical critical exponent $z \geq 0$ depends on the type of algorithm used. In the continuum, the correlation length diverges at criticality. On a lattice of finite extent $L$, $\xi$ approaches $\mathcal{O}(L)$.

Autocorrelations can be problematic because the approximation in Equation (2.1) is only valid for ensembles of statistically independent field configurations. Correlated samples introduce an additional systematic error, which one should try to reduce as much as possible. Significant improvements over purely local updates can be achieved with the HMC algorithm [22]. It has been established as the de facto standard method for producing configurations in lattice field theory and is routinely employed in state-of-the-art lattice calculations of QCD and other theories due to its superior volume scaling of the computational cost and other attractive features, see e.g. [8] for an in-depth discussion. The algorithm is based on the numerical treatment of Hamiltonian equations of motion in a fictitious time dimension. Quantum fluctuations are encoded by random sampling of the associated canonical momenta. Given a field configuration and a set of momenta, the Hamiltonian evolution is computed with a symplectic integrator such as the leapfrog method. An additional accept/reject step results in an algorithm satisfying detailed balance, despite the accumulation of numerical errors along the discretized integration trajectory. This allows to take larger steps in configuration space while retaining reasonable acceptance rates. Nevertheless, the HMC algorithm does not solve the problem of critical slowing down, which will be discussed in Section 3.1.

## 2.2 Scalar fields

We first consider the lattice formulation of one of the simplest interacting quantum field theories with bosonic fields, namely real, scalar $\phi^4$-theory. This discussion serves a dual purpose: on the one hand, it illustrates the basics and is instructive for the more involved concepts introduced later. On the other hand, the model considered in this section represents the bosonic part of scalar Yukawa theory, which is important for some of the machine learning applications in this work.

For simplicity, we restrict ourselves to isotropic, symmetric lattices spanning $L$ sites in each dimension with periodic boundary conditions. The lattice action in $d$ dimensions is defined as

$$S(\phi_0) = \sum_{x\in\Lambda} a^d \left[ \frac{1}{2} \sum_{\mu=1}^{d} \frac{(\phi_0(x+a\hat{\mu}) - \phi_0(x))^2}{a^2} + \frac{m_0^2}{2}\phi_0(x)^2 + \frac{g_0}{4!}\phi_0(x)^4 \right] , \tag{2.6}$$

where $\Lambda$ denotes the set of all lattice sites, $a$ is the lattice spacing, $\phi_0, m_0, g_0$ correspond to the bare field, mass, and coupling constant, and $\hat{\mu}$ is a unit vector in $\mu$-direction. We only consider single-component scalar fields for now, however, the generalization to $N$-component fields with an additional, internal $O(N)$ symmetry is straightforward.

The action can be cast into a dimensionless form through the following transformation:

$$\begin{aligned} a^{\frac{d-2}{2}}\phi_0 &= (2\kappa)^{1/2}\phi \\ (am_0)^2 &= \frac{1-2\lambda}{\kappa} - 2d \\ a^{-d+4}\lambda_0 &= \frac{6\lambda}{\kappa^2} \, . \end{aligned} \tag{2.7}$$

Here, $\kappa$ is commonly called the hopping parameter and $\lambda$ now takes the role of the

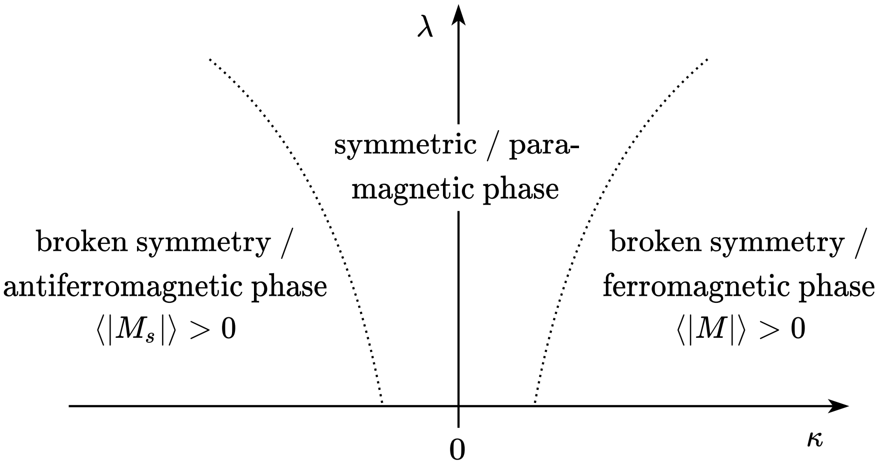

$\lambda$
symmetric / paramagnetic phase
broken symmetry / antiferromagnetic phase
$\langle|M_s|\rangle > 0$
broken symmetry / ferromagnetic phase
$\langle|M|\rangle > 0$
0
$\kappa$

Figure 2.2: Sketch of the phase diagram of real, scalar $\phi^4$-theory in the dimensionless formulation.

coupling constant. Applying this transformation results in

$$S(\phi) = \sum_{x\in\Lambda}\left[-2\kappa\sum_{\mu=1}^{d}\phi(x)\phi(x+\hat{\mu}) + (1-2\lambda)\phi(x)^2 + \lambda\phi(x)^4\right] . \tag{2.8}$$

The theory belongs to the Ising universality class. As such, in $d \geq 2$ it exhibits a phase transition associated with spontaneous breaking of the $Z_2$ symmetry, where field variables will jointly fall into one of two possible orientations. This can be quantified by the magnetization, defined as the average value of the field,

$$M(\phi) = \frac{1}{V}\sum_{x\in\Lambda}\phi(x) , \tag{2.9}$$

where $V = |\Lambda| = L^d$ denotes the dimensionless volume, i.e. the number of lattice sites. Its staggered counterpart is given by

$$M_s = \frac{1}{|\Lambda|}\sum_{x\in\Lambda}(-1)^{x_1+\cdots+x_d}\phi(x) , \tag{2.10}$$

which is relevant for negative $\kappa$. The dimensionless action has the additional staggered symmetry

$$\kappa \mapsto -\kappa \qquad \text{and} \qquad \phi(x) \mapsto (-1)^{x_1+\cdots+x_d}\phi(x), \tag{2.11}$$

which connects both magnetizations. One usually measures the average absolute values $\langle|M|\rangle, \langle|M_s|\rangle$, which provide non-zero order parameters that are large in the (anti-)ferromagnetic phases where the $Z_2$ symmetry is broken, and exponentially suppressed in the paramagnetic (symmetric) phase; see Figure 2.2 for a sketch of the phase diagram of this theory.

In the continuum, the phase transition is characterized by a divergence in the connected two-point susceptibility,

$$\chi_2 = V\left(\langle M^2\rangle - \langle M\rangle^2\right) \ . \tag{2.12}$$

On a finite lattice, one instead observes a peak which becomes narrower as the lattice volume is increased. Furthermore, the connected two-point correlation function of $\phi$ is defined as

$$C_\phi(x,y) = \langle\phi(x)\phi(y)\rangle - \langle\phi(x)\rangle\langle\phi(y)\rangle \ , \tag{2.13}$$

where we fix $\langle\phi(x)\rangle = \langle M\rangle = 0$ analytically. Here, one is often interested in the source-averaged correlator projected to zero momentum, defined as

$$C_\phi(t) = \frac{1}{V}\sum_x\sum_{\vec{y}} C(x, x+(\vec{y},t)) \ . \tag{2.14}$$

## 2.3 Fermions

The simulation of dynamical fermion degrees of freedom in lattice field theory is a highly non-trivial task for many theories of physical interest, both conceptually and computationally. In this section, the main concepts behind formulations of lattice fermions and their numerical implementation are reviewed.

### 2.3.1 Path integrals with fermions

We consider field theories of interacting fermionic and bosonic degrees of freedom discretized on a $d$-dimensional Euclidean hypercubic lattice with periodic boundary conditions. The action of such a theory can be expressed as

$$S(\psi,\bar{\psi},\phi) = S_B(\phi) + S_F(\psi,\bar{\psi},\phi) \ , \tag{2.15}$$

where the subscripts $B$ and $F$ denote the bosonic and fermionic contributions to the action, the discretized boson field variables are collectively denoted by $\phi$, and the discretized fermion field variables are denoted by $\psi, \bar{\psi}$. Here, we assume that the fermionic action is bilinear in $N_f$ flavors of Dirac fermions $\psi_f, \bar{\psi}_f$ and is given by

$$S_F(\psi,\bar{\psi},\phi) = \sum_{f=1}^{N_f} \bar{\psi}_f\, D_f(\phi)\, \psi_f \ , \tag{2.16}$$

where the Dirac operator $D_f(\phi)$ includes the kinetic terms, mass terms, and any coupling to bosonic fields for each fermion flavor $f$. The precise form of $D_f(\phi)$ is determined by the theory of interest and the choice of discretization.

Expectation values of observables $\mathcal{O}$ are computed via path integrals of the form

$$\langle \mathcal{O} \rangle = \frac{1}{Z} \int \mathcal{D}[\phi] \mathcal{D}[\psi, \bar{\psi}] e^{-S_F(\psi,\bar{\psi},\phi)} e^{-S_B(\phi)} \mathcal{O}(\psi, \bar{\psi}, \phi) \ , \tag{2.17}$$

where

$$Z = \int \mathcal{D}[\phi] \mathcal{D}[\psi, \bar{\psi}] \, e^{-S_F(\psi,\bar{\psi},\phi)} e^{-S_B(\phi)} \ , \tag{2.18}$$

and the fermion fields $\psi$ and $\bar{\psi}$ are defined in terms of anti-commuting Grassmann numbers. For bilinear actions of the form given in Equation (2.16), integration over the Grassmann-valued fermion fields can be performed explicitly, giving

$$\int \mathcal{D}[\psi, \bar{\psi}] e^{-S_F(\psi,\bar{\psi},\phi)} = \prod_{f=1}^{N_f} \det D_f(\phi) \ . \tag{2.19}$$

By applying Wick's theorem, the dependence of the observable on the fermions can be integrated out. Path integral expectation values can then be written in terms of purely bosonic degrees of freedom as

$$\langle \mathcal{O} \rangle = \frac{1}{Z} \int \mathcal{D}[\phi] \left[ \prod_{f=1}^{N_f} \det D_f(\phi) \right] e^{-S_B(\phi)} \mathcal{O}(\phi) \ . \tag{2.20}$$

This expectation value may again be estimated via MCMC sampling, by computing an average over a statistical ensemble of configurations $\phi$ sampled from the probability distribution

$$p(\phi) = \frac{1}{Z} e^{-S_B(\phi)} \prod_{f=1}^{N_f} \det D_f(\phi) \ . \tag{2.21}$$

For large lattice volumes, the fermion determinants are treated stochastically via the pseudofermion method introduced in the following section.

### 2.3.2 Pseudofermions

The fermion determinants in Equation (2.21) have to be evaluated for every proposed field configuration in order to determine the acceptance probability. They are not calculated exactly for large volumes because the Dirac matrices $D_f$ are high-dimensional, which makes it difficult to frequently perform a direct evaluation with currently accessible computing platforms. To see this, consider $d$-dimensional field configurations with $L$ sites per spatial dimension and $L_t$ sites in the temporal dimension. Since the total number of fermionic degrees of freedom scales as the total number of lattice sites, $V = L_t L^{d-1}$, each Dirac matrix $D_f$ then has dimensions $\mathcal{O}(V \times V)$. An exact computation of the determinants of such matrices quickly becomes intractable because the cost naively scales like $\mathcal{O}(V^3)$.

Instead, Gaussian integrals over auxiliary bosonic fields—the pseudofermions—can be used to replace the direct evaluation of determinant factors [23], based on the identity

$$\det \mathcal{M} = \frac{1}{Z_{\mathcal{N}}} \int \mathcal{D}[\varphi_R, \varphi_I]\, e^{-\varphi^\dagger \mathcal{M}^{-1} \varphi} \;, \tag{2.22}$$

where the normalization constant $Z_{\mathcal{N}}$ is defined as

$$Z_{\mathcal{N}} = \int \mathcal{D}[\varphi_R, \varphi_I]\, e^{-\varphi^\dagger \varphi} \;. \tag{2.23}$$

Here, $\varphi_R, \varphi_I$ denote the real and imaginary components of the auxiliary complex field $\varphi$, and the matrix $\mathcal{M}$ must be positive-definite. Since the Dirac matrices $D_f$ are typically not positive-definite, one cannot directly apply this identity to each factor of $\det D_f$. However, for fermion flavors $f_1$ and $f_2$ appearing as degenerate pairs, based on $\gamma_5$-hermiticity one can instead use the equality

$$\det D_{f_1} \det D_{f_2} = \det D_{f_1} D_{f_1}^\dagger \;, \tag{2.24}$$

and then apply Equation (2.22) to the positive-definite matrix $\mathcal{M} = D_{f_1} D_{f_1}^\dagger$. For fermion flavors $f$ not included in any degenerate pair, one can apply one-flavor algorithms [24–26] to replace $D_f$ with a positive-definite matrix $\mathcal{M}$ capturing identical dynamics.

Using the pseudofermion approach, a path integral as in Equation (2.20) can thus be rewritten in terms of an action involving the auxiliary pseudofermion fields $\varphi$,

$$\begin{aligned} S(\phi, \varphi) &= S_B(\phi) + S_{PF}(\phi, \varphi) \quad \text{with} \\ S_{PF}(\phi, \varphi) &= \varphi^\dagger \mathcal{M}^{-1}(\phi) \varphi \equiv \sum_{k=1}^{N_{pf}} \varphi_k^\dagger \mathcal{M}_k^{-1}(\phi) \varphi_k \;, \end{aligned} \tag{2.25}$$

after replacing the fermion determinants in the given lattice theory by the determinants of $N_{pf}$ positive-definite matrices $\mathcal{M}_k$ as

$$\prod_{f=1}^{N_f} \det D_f(\phi) = \prod_{k=1}^{N_{pf}} \det \mathcal{M}_k(\phi) \;. \tag{2.26}$$

Each term $\varphi_k^\dagger \mathcal{M}_k^{-1} \varphi_k$ in the pseudofermion action can be efficiently computed using iterative solvers such as the conjugate gradient method. Having formulated the theory using pseudofermions in Equation (2.25), evaluation of the path integral via MCMC can then be performed in this augmented space by sampling from the joint distribution

$$p(\phi, \varphi) = \frac{1}{Z} e^{-S_B(\phi) - S_{PF}(\phi, \varphi)} \;. \tag{2.27}$$

This is commonly done via Gibbs sampling: the boson and pseudofermion variables are evolved in an alternating fashion where one set of variables is kept fixed while the

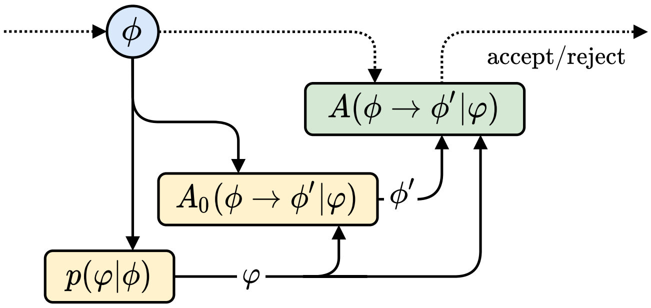


Figure 2.3: Illustration of updates with pseudofermions via Gibbs sampling.

| Name | Probability density |
|---|---|
| Joint[A] | $p(\phi, \varphi) = \frac{1}{Z} \exp(-S_B(\phi) - \varphi^\dagger \left[\mathcal{M}(\phi)\right]^{-1} \varphi)$ |
| $\phi$-marginal | $p(\phi) = \frac{Z_\mathcal{N}}{Z} \exp(-S_B(\phi)) \det \mathcal{M}(\phi)$ |
| $\varphi$-conditional[A,B] | $p(\varphi\|\phi) = \dfrac{1}{Z_\mathcal{N} \det \mathcal{M}(\phi)} \exp(-\varphi^\dagger \left[\mathcal{M}(\phi)\right]^{-1} \varphi)$ |
| $\varphi$-marginal[C] | $p(\varphi) = \frac{1}{Z} \int d\phi \, \exp(-S_B(\phi) - \varphi^\dagger \left[\mathcal{M}(\phi)\right]^{-1} \varphi)$ |
| $\phi$-conditional[A] | $p(\phi\|\varphi) = \dfrac{\exp(-S_B(\phi) - \varphi^\dagger \left[\mathcal{M}(\phi)\right]^{-1} \varphi)}{\int d\phi \, \exp(-S_B(\phi) - \varphi^\dagger \left[\mathcal{M}(\phi)\right]^{-1} \varphi)}$ |

Table 2.1: List of possible distributions derived from the joint target density in Equation (2.27). The normalizing constant $Z$ is given by Equation (2.18) and $Z_\mathcal{N}$ is defined in Equation (2.23). Notes: (A) Only the joint, $\varphi$-conditional, and $\phi$-conditional densities can be tractably computed (up to normalization). (B) The $\varphi$-conditional can be sampled exactly. (C) A closed form for the $\varphi$-marginal density is not generally known (even unnormalized).

other is updated using the respective conditional; see Figure 2.3 for an illustration. However, the joint distribution $p(\phi, \varphi)$ admits several possible marginalizations and decompositions; see Table 2.1. This will be relevant for the different generative modeling approaches introduced in Chapter 6.

While the pseudofermion method renders the treatment of fermion determinants tractable in principle, the joint action may strongly fluctuate in certain limits. This feature can slow down MCMC sampling of the joint distribution and can lead to an unfavorable volume scaling of the associated computational effort, especially when many components of the bosonic field are updated simultaneously. Accordingly, numerous modifications of the pseudofermion formulation have been developed to improve the structure of the action; see e.g. [27–32].

### 2.3.3 Boundary conditions and translational symmetry

In lattice studies of purely bosonic theories, it is common to choose periodic boundary conditions in all directions of the lattice, allowing the incorporation of an exact discrete translational symmetry in lattice actions for such theories. For fermion fields, one needs to impose antiperiodic boundary conditions in the time direction in order to obtain a consistent definition of the trace for the Euclidean partition function. Actions for theories involving fermionic fields are then invariant under simultaneous spatio-temporal translations of $\phi, \psi, \bar{\psi}$ with appropriate boundary conditions applied for each field. To be consistent with the boundary conditions for $\psi$ and $\bar{\psi}$, each Dirac matrix $D_f(\phi)$ must include appropriate signs for any terms coupling fields across the temporal boundary. As a result, these boundary conditions affect the pseudofermion formulation of the theory as well, and the pseudofermion action $S_{PF}(\phi, \varphi)$ is invariant under simultaneous translations of $\phi$ and $\varphi$ with antiperiodic temporal boundary conditions applied to $\varphi$.

In general, the discretization chosen for the Dirac operator determines which particular lattice translations are included in the translational symmetry group. In the staggered formulation [33], for example, the spinor components of each flavor of fermion are distributed over the components of hypercubes with $2^d$ sites each, and the translational symmetry group includes all translations by an even number of sites. Translations by an odd number in particular directions correspond to more complicated internal symmetry transformations that mix spinor degrees of freedom, and must involve sign flips on specific field components to leave the staggered action invariant [34]. These symmetries of the staggered formulation play a role in the staggered-fermion Yukawa model presented in the next section.

## 2.4 Yukawa theory

The simplest theories with dynamical fermions considered in this work are two- and three-dimensional models of a scalar field coupled to fermions via a Yukawa interaction. Specifically, we consider a real, scalar field $\phi$ coupled to one mass-degenerate pair of Kogut-Susskind staggered fermions [33]. This model provides a testbed which features fermionic fields, but without the additional complications brought on by gauge symmetry. The phase diagram of this theory at small values of the Yukawa coupling is similar to the one for pure $\phi^4$-theory shown in Figure 2.2. The effect of introducing fermions then only amounts to a global shift and smearing of the phase boundaries; see Figure 2.4 for a slice of the phase diagram at fixed values of the couplings. For larger values of $g$, the fermions take on an increasingly important role and can give rise to new phases; see [9] for further information. Apart from providing a suitable test case, studying Yukawa interactions is also interesting in its own right, e.g. for Higgs physics [35] or the quark-meson model [36].

For the purely bosonic $S_B(\phi)$, we choose the $\phi^4$-theory action introduced in Section 2.2. The fermionic action $S_F$ is given by the bilinear form in Equation (2.16)

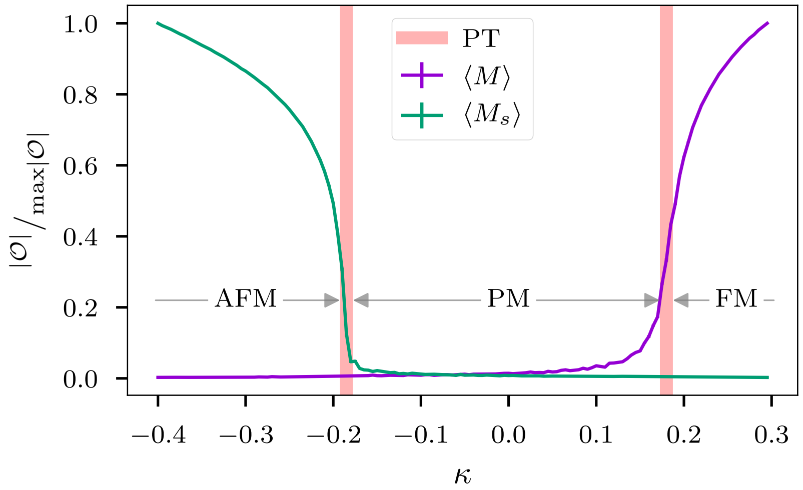


Figure 2.4: Slice of the phase diagram of three-dimensional Yukawa theory for fixed couplings $g = 0.25, \lambda = 1.1$ using normalized values of $\langle M\rangle$ and $\langle M_s\rangle$. Phase transitions separating an antiferromagnetic (AFM), a paramagnetic (PM), and a ferromagnetic (FM) phase are highlighted by the shaded bars. For details about the calculation, see Appendix A.1.

with $N_f = 2$, and both fermion flavors are defined by the discretized Dirac operator

$$\begin{aligned} D_{xy} = \sum_{\mu=1}^{d} \eta_\mu(x) \frac{\delta(x-y+\hat{\mu}) - \delta(x-y-\hat{\mu})}{2} \\ + \delta(x-y)(m_f + g\phi(x)) \ , \end{aligned} \tag{2.28}$$

where $m_f$ is the bare mass of the fermion and $g$ the Yukawa coupling. The staggered factor $\eta_\mu$ is obtained from the Dirac $\gamma$-matrices after the staggered transformation and is defined as

$$\eta_1(x) = 1 \quad \text{and} \quad \eta_l(x) = (-1)^{x_1} \cdots (-1)^{x_{l-1}} \ . \tag{2.29}$$

The Kronecker $\delta$ is defined to have antiperiodic boundary conditions in the time direction (conventionally taken to be $\mu = d$) and periodic boundary conditions in the spatial directions, i.e.

$$\delta(x) = \prod_{\mu=1}^{d} \delta_\mu(x_\mu) \ , \tag{2.30}$$

where

$$\begin{aligned} \delta_{\mu\neq d}(x_\mu) &= \begin{cases} 1 & \text{if } x_\mu = 0, \pm L \\ 0 & \text{otherwise} \end{cases} \\ \text{and} \quad \delta_d(x_d) &= \begin{cases} 1 & \text{if } x_d = 0 \\ -1 & \text{if } x_d = \pm L_t \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{2.31}$$

Fermionic observables can be computed from the matrix elements of the inverse Dirac operator. The chiral condensate of the fermion field is defined as

$$\langle \bar{\psi}\psi \rangle = \left\langle \frac{1}{V} \mathrm{Tr}\, D^{-1} \right\rangle \ , \tag{2.32}$$

and one measures $\langle |\bar{\psi}\psi| \rangle$ as for the magnetization of the scalar field. Using the off-diagonal matrix elements, one can also evaluate the average fermionic two-point correlator in the time direction,

$$C_\psi(t) = \left\langle \psi(y)\, \bar{\psi}(0) \right\rangle = \left\langle D^{-1}_{y,0} \right\rangle \ , \tag{2.33}$$

where $y = (\vec{0}, t)$ with $t$ odd. The particular choices of offsets $y$ select staggered spinor components at the sink $\psi(y)$ that result in a non-zero average correlation function originating from the source $\bar{\psi}(0)$.

## 2.5 Gauge fields

The introduction of gauge fields becomes necessary when one assumes local gauge invariance, a symmetry principle that is essential in the formulation of the standard model of particle physics[1]. Specifically, we demand that a local gauge transformation of the fermion fields on each lattice site of the form

$$\psi(x) \to \Omega(x)\psi(x) \text{ and } \bar{\psi}(x) \to \bar{\psi}(x)\Omega^\dagger(x) \tag{2.34}$$

leaves the action of the theory invariant. Here, the $\Omega(x)$ are arbitrary, independent elements of the chosen gauge group. In this work, only the unitary gauge groups $U(1), SU(N)$ are considered, for which $\Omega^{-1} = \Omega^\dagger$ and in the case of $SU(N)$ we additionally demand $\det \Omega = +1$. This allows for some simplifications, however, other choices of compact Lie groups are also possible. The unitary groups have particular physical relevance, since the standard model is constructed from $U(1) \times SU(2) \times SU(3)$.

Analyzing the effect of a local gauge transformation on the discretized derivative term in the fermionic action,

$$\bar{\psi}(x) \frac{1}{2a} (\psi(x+\hat{\mu}) - \psi(x-\hat{\mu})) \ , \tag{2.35}$$

we find that it is not invariant:

$$\bar{\psi}(x)\psi(x+\hat{\mu}) \to \bar{\psi}(x)\Omega^\dagger(x)\Omega(x+\hat{\mu})\psi(x+\hat{\mu}) \ . \tag{2.36}$$

To restore the invariance of the action under the transformation, an additional field $U_\mu(x)$ must be introduced which transforms as

$$U_\mu(x) \to \Omega(x) U_\mu(x) \Omega^\dagger(x+\hat{\mu}) \ . \tag{2.37}$$

[1] Interestingly, it can also be used to construct general relativity—the theory of gravity—from the perspective of quantum field theory [37].

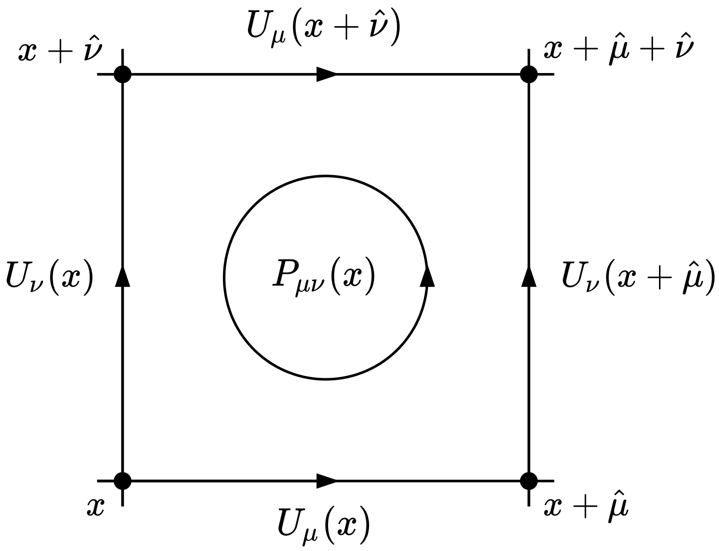


Figure 2.5: Illustration of a plaquette, the smallest possible Wilson loop.

This is the gauge field. Gauge invariance is restored by using

$$\bar\psi(x)U_\mu(x)\psi(x+\hat\mu) \quad \text{instead of} \quad \bar\psi(x)\psi(x+\hat\mu) \tag{2.38}$$

in the derivative term. In contrast to the types of fields discussed so far, $U_\mu(x)$ has a direction: it does not live on the lattice sites themselves, but connects the neighboring sites $x$ and $x+\hat\mu$. For this reason, it is also commonly called a gauge link. Evaluating a gauge link in the opposite direction gives its inverse $U_\mu(x)^\dagger$ and we identify $U_{-\mu}(x) = U_\mu(x-\hat\mu)^\dagger$.

The last step in our construction is to equip the gauge field with its own dynamics, i.e. to find a gauge-invariant action that only depends on $U_\mu(x)$. In its simplest form, such an action may be defined as

$$S_g(U) = -\frac{\beta}{N}\sum_x \sum_{\substack{\mu,\nu \\ \mu<\nu}} \mathrm{Re}\,\mathrm{Tr}\, P_{\mu\nu}(x) \ . \tag{2.39}$$

Here, $\beta$ is the inverse of the squared gauge coupling, and the plaquette $P_{\mu\nu}(x)$ is the smallest possible Wilson loop—a product of links around a $1\times1$ square whose trace is gauge-invariant; see Figure 2.5 for an illustration. It is defined as

$$P_{\mu\nu}(x) = U_\mu(x)U_\nu(x+\hat\mu)U_\mu^\dagger(x+\hat\nu)U_\nu^\dagger(x) \ . \tag{2.40}$$

This action is called the Wilson gauge action and is perhaps the most widely used version, although other forms yielding the same continuum limit have been studied as well. To see how the familiar gauge action in terms of the field strength tensor is recovered in the continuum, the discussion in [8] is recommended.

## 2.6 The Schwinger model

One of the simplest fermionic gauge theories is the $N_f = 2$ Schwinger model, a strongly interacting two-dimensional $U(1)$ gauge theory coupled to two degenerate fermions (conventionally with a charge of $+1$). The theory exhibits similar features to QCD: confinement, spontaneous chiral symmetry breaking due to a chiral condensate, and non-trivial topology [38, 39]. It commonly serves as a toy model for QCD, and is often used for testing new approaches to lattice field theory [40–46], including methods using quantum technologies [47–49]. It has also been used to study properties of quantum field theories [39, 50–59].

Integrating out the fermionic degrees of freedom as described in Section 2.3 yields a lattice action of the form [60–62]

$$S(U) = -\beta \sum_x \operatorname{Re} P_{01}(x) - \log\det D[U]^\dagger D[U] \ , \tag{2.41}$$

where the first term is the gauge action as defined in the previous section, but specifically for the gauge group $U(1)$ in two dimensions. We consider the Wilson discretization [63, 64] of the lattice Dirac operator $D[U]$, given by

$$\begin{aligned} D[U](y,x)^{\beta\alpha} = \delta(y-x)\delta^{\beta\alpha} - \kappa \sum_{\mu=0,1} \Big\{ & [1-\sigma_\mu]^{\beta\alpha} U_\mu(y)\delta(y-x+\hat{\mu}) \\ & + [1+\sigma_\mu]^{\beta\alpha} U_\mu^\dagger(y-\hat{\mu})\delta(y-x-\hat{\mu}) \Big\} \ , \end{aligned} \tag{2.42}$$

where $\sigma_\mu = (\sigma_x, \sigma_y)$, with $\sigma_{x,y}$ denoting the usual Pauli matrices, and $\hat{\mu}$ is again a unit vector in direction $\mu$. The $\delta$-functions are again defined with anti-periodic boundary conditions in the time direction. The bare fermion mass $m_0$ is controlled by the hopping parameter $\kappa = 1/(4 + 2m_0)$ that parametrizes $D$.

$U(1)$ lattice gauge field configurations in two dimensions belong to discrete topological sectors, as quantified by the integer-valued topological charge, commonly defined as [55]

$$Q = \frac{1}{2\pi} \sum_x \theta_P(x) \in \mathbb{Z} \ , \tag{2.43}$$

where the plaquette angles $\theta_P$ are restricted to the principal branch,

$$\theta_P(x) = \operatorname{Im}\log P_{01}(x) \in (-\pi, \pi] \ . \tag{2.44}$$

The associated topological susceptibility is defined as

$$\chi_Q = \frac{1}{V} \langle Q^2 \rangle \ , \tag{2.45}$$

where $V$ is the volume in lattice units, i.e. the total number of sites.

For the action parameters and lattice volume investigated in this work, the topological charge distribution displays frequent UV fluctuations. This issue complicates the unambiguous detection of topological freezing based solely on $Q$ values—akin to the situation in QCD—thereby making it difficult to conclusively demonstrate whether a given sampling algorithm is frozen or not. For the Schwinger model, a more suitable observable that indicates hops between sectors and appears unaffected by this problem is the sign of the real part of the fermion determinant,

$$\sigma = \mathrm{sign}(\mathrm{Re}\ \det D)\ . \tag{2.46}$$

This can be inferred from a lattice analysis of the continuum Atiyah-Singer index theorem [60]. Additionally, the chiral condensate defined in Equation (2.32) is considered for the Schwinger model as well. Its value is also correlated with the topological sectors and is therefore sensitive to freezing.

The Schwinger model is quantum electrodynamics in two spacetime dimensions, but shares many features with QCD, as mentioned above. One may also view QCD as a generalization of the $N_f = 2$ Schwinger model discussed here, namely in four dimensions, with the gauge group $U(1)$ replaced by $SU(3)$, and more dynamical quark flavors. While there are six quarks in the standard model, accurate results at the energy scales of interest can already be obtained with a dynamical treatment of only the three lightest flavors, namely up, down, and strange. Of course, there are many conceptual and practical differences between calculations in the Schwinger model and QCD, and this short comment certainly does not do justice to the great efforts invested over many years into making QCD calculations work at scale. For a detailed treatment of full QCD on the lattice, the interested reader is again referred to one of the standard textbooks [8–10].

# 3 Computational barriers in lattice QCD

After introducing the relevant theoretical and algorithmic concepts in the last chapter, we are now ready to face the computational barriers that are the focus of this work. In Section 3.1, aspects of critical slowing down in lattice calculations are considered, specifically the problem of topological freezing. Subsequently, complex action problems are illustrated in Section 3.2. Finally, we discuss the problem of extracting real-time physics from imaginary-time data via spectral reconstruction in Section 3.3. The following text has some overlap with parts of [2, 4–7].

## 3.1 Topological freezing

The sequential nature of the Markov chain is a potential drawback to the MCMC sampling approach for computing path integrals in lattice field theory. As already mentioned in Chapter 2, known Markov chain update schemes for many theories of interest are local or diffusive, which results in autocorrelations between successive elements of the chain. Naturally, the stronger these autocorrelations become, the more samples must be drawn to achieve a result at fixed statistical precision. Close to criticality—e.g. when approaching the continuum limit of lattice field theories or in order to describe universal properties of condensed matter systems—autocorrelations diverge rapidly for such local or diffusive Markov chains. This issue, referred to as critical slowing down, can render computations prohibitively expensive [65–67]. Autocorrelations may become especially severe if MCMC updates are unlikely to generate transitions between modes that are separated in configuration space. This effect, known as “freezing,” can prevent an effective exploration of the distribution for any practical sample size and amounts to an in-practice violation of ergodicity, which is a necessary condition for the validity of MCMC.

In particular, as explained previously, the HMC algorithm generates samples by continuously evolving the fields through configuration space via Hamiltonian dynamics. Calculations based on HMC for theories whose path integral is separated into distinct topological sectors are often plagued by a pathologically slow mixing of the associated topological charges or winding numbers; see Figure 3.1 for an illustration. In the continuum theory, transformations between field configurations of different topology cannot be achieved by smooth deformations due to the presence of poles. In the lattice formulation, the sectors are instead separated by large potential barriers, which are difficult to overcome with any sampling algorithm that takes small steps in configuration space. As the height of these barriers increases when approaching the continuum limit, tunneling becomes extremely unlikely and achieving ergodicity grows prohibitively expensive.

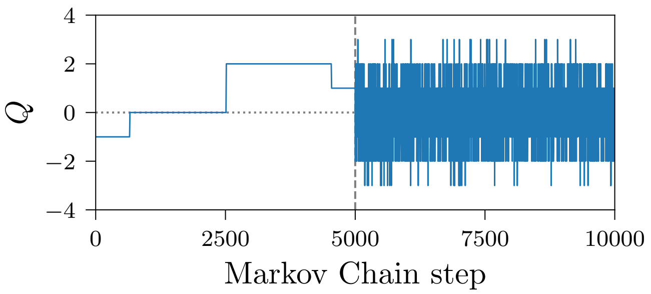


Figure 3.1: Illustration of topological freezing with rare tunneling events (left) vs the desired behavior of a fast mixing topological charge (right).

The existence of discrete topological sectors is a consequence of the commonly employed periodic boundary conditions. Hence, one may argue that the importance of topological effects is negligible for lattices beyond a certain size. However, in practice it is difficult to precisely specify when this point is reached, which makes it hard to guarantee correctness. Generally, the validity of expectation values computed via MCMC methods requires the autocorrelation times of all observables to be much shorter than the total length of the sampled Markov chain. The occurrence of topological freezing indicates insufficient ergodicity at reasonable simulation times even with modern supercomputers, which makes it difficult to rigorously quantify statistical uncertainties. Hence, this may be problematic for results derived from statistical ensembles that are obtained with conventional HMC or similar methods.

These challenges have motivated extensive work to replace such local/diffusive MCMC algorithms with other sampling procedures. Specialized Markov chain steps have been developed in a number of specific contexts, including cluster updates [68–76], worm algorithms [77–79], sampling in terms of dual variables [80–82], and event-chain algorithms [83–87]. Though these ideas have been shown to mitigate critical slowing down in some settings, currently they cannot be applied to many theories of interest, including lattice QCD. Solving critical slowing down would unlock much larger physical volumes in such calculations and thereby open up a number of interesting scientific avenues to explore. In particular, it would allow us to study the physics of large nuclei from first principles. Moreover, it may improve our understanding of condensed matter systems where no efficient algorithms are currently known. In this context, a promising ansatz is the use of novel generative machine learning models that can provide statistically independent samples. A first step in this direction based on overrelaxation with generative adversarial networks is detailed in Section 5.1; however, such ideas have since been massively improved upon through architectures with tractable likelihoods. These types of models enable direct importance sampling with guaranteed asymptotic exactness and exhibit various other attractive features. In Section 6.6, such an approach is investigated in the context of the Schwinger model at critiality, demonstrating that topological freezing can be successfully mitigated.

## 3.2 Complex action problems

Moving on to the second of the computational challenges outlined above, in this section we briefly discuss complex action problems. As already mentioned, the notorious sign problem associated with finite baryon chemical potential $\mu_B$ prevents the exploration of the QCD phase diagram in some of the most interesting regions. For a pedagogical introduction to the topic, [88] is recommended. This particular issue is an instance of a fermionic sign problem, versions of which also prohibit studies of many strongly correlated condensed matter systems. Different types of complex action problems can also emerge in various settings, such as with topological terms in the actions of lattice gauge theories (e.g. QCD with a $\theta$-term) or for complex external fields, among others.

To illustrate complex action problems with a simple example, consider the purely real Gaussian integral [89]

$$Z(\lambda) = \int_{-\infty}^{\infty} \mathrm{d}x \, e^{-x^2+\mathrm{i}\lambda x} = \sqrt{\pi} e^{-\frac{\lambda^2}{4}} \; . \tag{3.1}$$

It has the form of a simple partition function with a complex action for non-zero values of $\lambda$. The real part of the integrand is shown in Figure 3.2, illustrating that it is a smooth and positive function for $\lambda = 0$, but oscillates wildly for $\lambda = 50$. Whereas the integral takes the value $\sqrt{\pi}$ for $\lambda = 0$, for $\lambda = 50$ it evaluates to $\sim 6.5 \times 10^{-272}$. Therefore, for a naive stochastic evaluation of this integral, an extremely large number of Monte Carlo samples with contributions of the order $\mathcal{O}(1)$ needs to be averaged over in order to yield a result that is hundreds of orders of magnitude smaller. Obtaining a reliable result with sufficient statistical significance to be distinguished from zero thus represents a highly difficult task. The precise evaluation of integrals of highly oscillatory functions via statistical sampling is an instance of a numerical sign problem or complex action problem. In specific cases, sign problems can be shown to be NP-hard [90]. This means that the complexity of the algorithm scales worse than polynomially with the system size.

For complex actions, standard importance sampling is not applicable due to the breakdown of the usual interpretation of the Boltzmann factor as a positive-definite probability. Formally, this is not a problem, since in principle it is always possible to shift the problematic part of the full weighting factor $\rho(x) \in \mathbb{C}$ into the observable and computing expectation values using a smooth and real weight $\rho_R(x) \in \mathbb{R}$, i.e.

$$\langle \mathcal{O}(x) \rangle = \frac{\int \mathrm{d}x \, \mathcal{O}(x) \rho(x)}{\int \mathrm{d}x \, \rho(x)} = \frac{\int \mathrm{d}x \, \mathcal{O}(x) \frac{\rho(x)}{\rho_R(x)} \rho_R(x)}{\int \mathrm{d}x \, \frac{\rho(x)}{\rho_R(x)} \rho_R(x)} = \frac{\left\langle \mathcal{O}(x) \frac{\rho(x)}{\rho_R(x)} \right\rangle_{x \sim \rho_R(x)}}{\left\langle \frac{\rho(x)}{\rho_R(x)} \right\rangle_{x \sim \rho_R(x)}} \; . \tag{3.2}$$

This procedure is called reweighting, and the denominator of the last expression is usually referred to as the average sign. A common choice for the real-valued weighting factor is $\rho_R \equiv |\rho|$. Using the language of statistical mechanics, the average

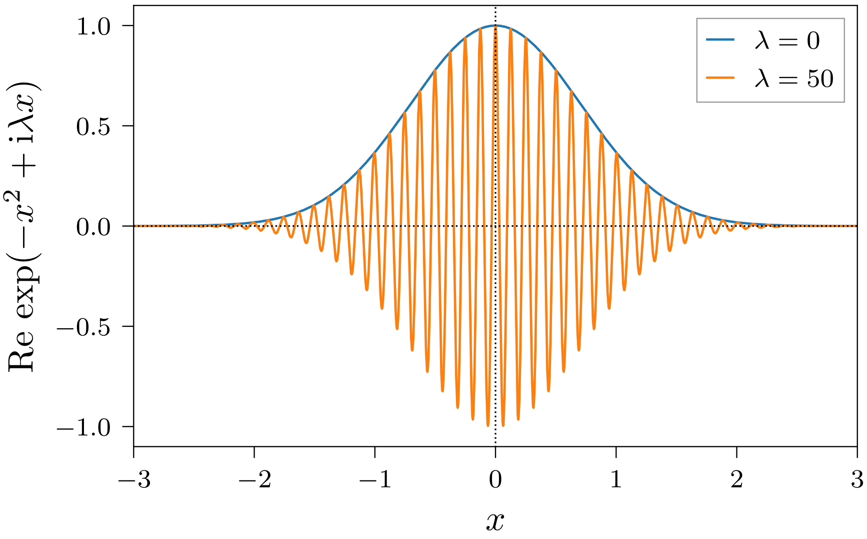


Figure 3.2: Illustration of a complex action problem in a simple Gaussian integral.

sign can be written as

$$\frac{1}{Z_R}\int \mathrm{d}x\,\frac{\rho(x)}{\rho_R(x)}\rho_R(x) = \frac{1}{Z_R}\int \mathrm{d}x\,\rho(x) = \frac{Z}{Z_R} = e^{-\frac{V}{T}\Delta f(T)}\,. \tag{3.3}$$

Here, we have identified $Z = e^{-V/Tf(T)}$, where $f(T)$ denotes the free energy density. Accordingly, one can see that at fixed temperature, the average sign decays exponentially with increasing system size. Since one is usually interested in taking the thermodynamic limit, this becomes problematic: for an average sign close to zero, the cancellation of large contributions requires an extremely large sample size to yield a reliable result. The computational cost then grows exponentially with the volume in order to achieve the same numerical accuracy.

A variety of approaches has been explored to solve complex action problems in many contexts. For QCD and related theories, these include reweighting [91], complex Langevin [92–95], Lefschetz thimbles [96–98], dual formulations [99–104], Taylor expansion around $\mu_B = 0$ [105–107], analytic continuation from imaginary $\mu_B$ [107–109], and the density-of-states (DoS) method [110–112]. It is widely believed that no general solution exists that is applicable to all complex action problems. Instead, different problem settings likely require individual, specialized approaches. In this thesis, the focus is on the DoS method. Specifically, in Chapter 7 it is shown that with a particular generative machine learning architecture—the same approach used to solve critical slowing down in the Schwinger model—the DoS can be computed directly for certain theories with complex actions, thereby circumventing some of the limitations of traditional MCMC methods.

## 3.3 From imaginary to real time

We now turn our attention to the third and final computational barrier considered in this work: the extraction of real-time physics from imaginary-time data via spectral reconstruction, which is a heavily ill-conditioned inverse problem. While many static properties of strongly correlated quantum systems are by now well understood and routinely computed using the combined strength of different non-perturbative approaches—such as lattice calculations and functional methods—a similar understanding of real-time properties is still the subject of ongoing research. Take e.g. the phenomenon of energy and charge transport, which so far has defied a quantitative understanding from first principles. This universal phenomenon is relevant to systems at vastly different energy scales, ranging from ultracold quantum gases to the quark-gluon plasma. There are two limitations preventing the direct application of most non-perturbative approaches to this problem. Firstly, in order to carry out quantitative computations, time has to be analytically continued into the complex plane to Euclidean time. Direct computations in Minkowski spacetime would require evaluating path integral expressions weighted by a pure phase of the form $\exp{(iS)}$, which may be viewed as the most difficult type of complex action problem imaginable. Secondly, explicit computations are either fully numerical or at least involve intermediate numerical steps, thereby complicating the analytic continuation of results from imaginary back to real time.

For QCD in particular, the resolution of many open questions requires the knowledge of time-like observables and hence the computation of real-time correlation functions. Applications include the hadronic resonance spectrum, scattering processes, as well as transport and non-equilibrium phenomena in heavy-ion collisions. For example, the computation of the glueball spectrum via Bethe-Salpeter equations relies on time-like propagators for the gluon and ghost. Likewise, QCD transport coefficients used in hydrodynamic simulations can be computed diagrammatically from the real-time gluon propagator. Furthermore, phenomenological transport models with their underlying assumption of a quasi-particle nature of the gluon can hugely benefit in multiple ways from these quantities. First of all, a reliable computation of the gluon spectral function may offer much-needed support for the quasi-particle assumption of these models, as well as give access to its limitations. Secondly, the gluon spectral function itself can feature as a direct input and pivotal building block in these models. Together with further time-like correlation functions, this offers a path for a systematic quantitative improvement of phenomenological transport approaches towards first-principle transport in QCD.

To make progress in this direction, one needs to undo the analytic continuation of approximately known Euclidean correlation functions. The most relevant examples are two-point functions, the Euclidean propagators. The Källén-Lehmann (KL) spectral representation [113, 114] relates the propagators, be they in Minkowski or Euclidean time, to a single function encoding their physics—the spectral function. If one can extract from the Euclidean two-point correlator its spectral function, the corresponding real-time propagator can be immediately computed.

The KL representation of two-point correlation functions in momentum space reads

$$G(p_0) = \int_0^\infty \frac{\mathrm{d}\omega}{\pi} \frac{\omega\,\rho(\omega)}{\omega^2 + p_0^2} = \int_0^\infty \mathrm{d}\omega\, K(p_0, \omega)\, \rho(\omega)\ , \tag{3.4}$$

with the KL kernel $K(p_0, \omega)$ and $\rho(-\omega) = -\rho(\omega)$. In the vacuum, the spatial momentum dependence of the propagator can be obtained via a Lorentz boost by $p_0^2 \to p^2$ with $p^2 = p_0^2 + \vec{p}^2$, and we write $p$ instead of $p_0$ from now on for notational simplicity. With Equation (3.4), the spectral function is obtained from the retarded propagator via

$$\rho(\omega) = 2\,\mathrm{Im}\, G(-\mathrm{i}(\omega + \mathrm{i}0^+))\ . \tag{3.5}$$

A general spectral function consists of a continuous part $\tilde{\rho}$ and a sum of particle and resonance peaks (proportional to the $\delta$-function and its derivatives). For asymptotic states, $\rho$ is the probability for (multi-)particle excitations to be created from the vacuum in the presence of the corresponding quantum field. Consequently, in this case it is positive semi-definite. For propagators of 'unphysical' fields, such as gauge fields, the spectral representation may still hold. However, the spectral function can then also have negative parts, and the existence of a spectral representation simply constrains the allowed complex structure of correlation functions; see e.g. [115–119].

Importantly, Euclidean correlators obtained from numerical calculations are generally only available in terms of discrete sets of observations $G_i$ at $N_G$ Euclidean momenta $p_i$ with finite precision. Relating the results to the associated Minkowski propagators via Equation (3.5) is problematic; see e.g. [120, 121]. The analytic continuation via $p \to -\mathrm{i}(\omega + \mathrm{i}0^+)$ is formally ill-conditioned, since further assumptions about the complex structure need to be made. Instead, the usual strategy is the numerical inversion of the integral transformation—which, however, is numerically ill-conditioned and needs to be regularized. This quickly becomes evident when one approximates the KL integral by a discrete sum and attempts a naive inversion of the resulting matrix-vector multiplication to compute the spectral function at a discrete set of frequencies $\omega_i$ with spacing $\Delta\omega$, i.e.

$$G_i = \tilde{K}_{ij}\, \rho_j \quad \longrightarrow \quad \rho_i = (\tilde{K}^{-1})_{ij}\, G_j\ , \tag{3.6}$$

where $\tilde{K}_{ij} = \Delta\omega K(p_i, \omega_j)$. For common ranges of $p_i$ and $\omega_i$, the matrix $\tilde{K}$ exhibits a pathologically large condition number due to the presence of small eigenvalues. Since these eigenvalues become extremely large in the inverse matrix, even tiny fluctuations in the data points $G_i$ are greatly exacerbated. Hence, any attempt to compute an approximation $\hat{\rho}$ by naive inversion fails spectacularly at the magnitudes of statistical errors typically achieved in numerical calculations of Euclidean correlation functions.

The problem is illustrated in Figure 3.3 at the example of a Breit-Wigner distribution as the spectral function, defined as

$$\rho_B(\omega) = \frac{4A\Gamma\omega}{4\Gamma^2\omega^2 + (M^2 + \Gamma^2 - \omega^2)^2}\ , \tag{3.7}$$

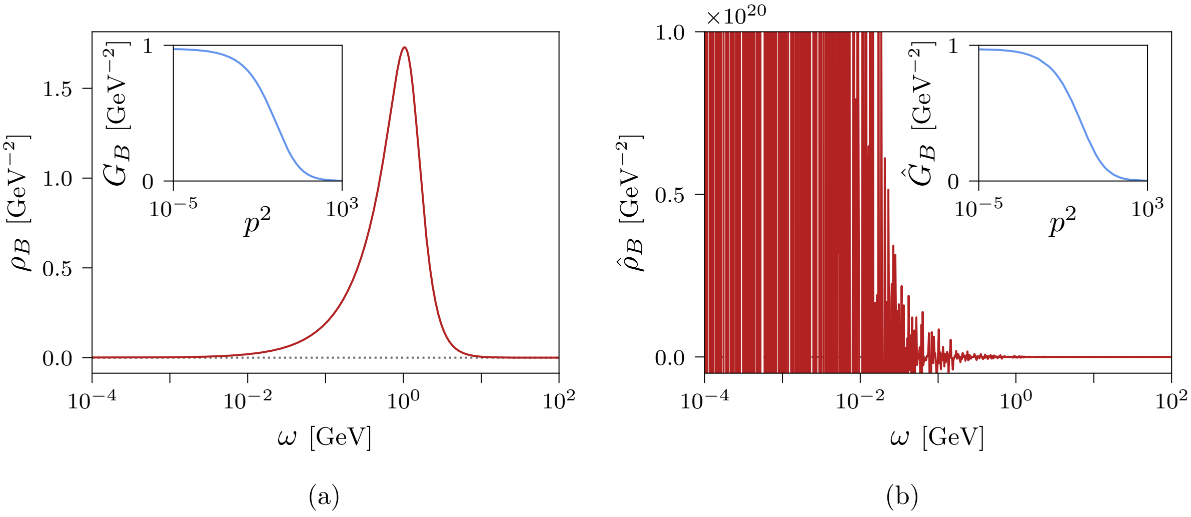


Figure 3.3: Illustration of an ill-conditioned reconstruction. (a) Breit-Wigner spectral function and associated propagator. (b) Spectral function obtained from the propagator with small additive noise using a naive matrix inversion. The result is completely polluted by strong fluctuations.

using the parameters $A = 1.6$, $M = 1$, $\Gamma = 0.8$. Equation (3.4) can be solved analytically for this spectral function to yield the propagator

$$G_B(p) = \frac{A}{(p+\Gamma)^2 + M^2} \, . \tag{3.8}$$

Applying Equation (3.6) to compute an approximation $\hat{\rho}_B$ from $\hat{G}_B = G_B + \epsilon$, using additive Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma)$ with $\sigma = 10^{-4}$, results in fluctuations tens of orders of magnitude larger than the true solution.

A variety of approaches has been explored to tackle this issue, such as the maximum entropy method [122–124], Bayesian inference techniques [125, 126], suitable expansions in functional spaces [117, 120, 121, 127, 128], Padé-type approximants [129, 130], Tikhonov regularization [131–133], neural networks [134–137], and kernel ridge regression [138, 139]. Alternative approaches based on the existence of complex conjugate poles have also been considered, see e.g. [129, 140–147]. In Chapter 8, a novel approach to spectral reconstruction from the perspective of probabilistic inverse theory with Gaussian processes is investigated.

# 4 Machine learning

In our quest towards breaking through the computational barriers discussed in Chapter 3, we turn to the application of machine learning techniques. These methods have proven capable of efficiently identifying high-level features in a broad range of data types—in many cases, such as speech or image recognition, with spectacular success. They are also increasingly applied to a variety of problems in the natural sciences. Deep neural networks in particular have demonstrated unprecedented levels of prediction and generalization performance in scientific tasks. Accordingly, there is also growing interest in the lattice community to harness the capabilities of these algorithms. Applications include predictive objectives, such as detecting phase transitions from lattice configurations or predicting action parameters, as well as generative modeling for the development of novel sampling algorithms. For a pedagogical introduction to machine learning for physicists, the interested reader is referred to [148]. For reviews on applications in physics, see [149–151].

This chapter introduces the relevant frameworks that are employed in the present work, namely deep neural networks in Section 4.1, normalizing flows in Section 4.2, and Gaussian process regression in Section 4.3. The following text has some overlap with parts of [5–7].

## 4.1 Deep neural networks

In this section, the principles behind deep neural networks are sketched, starting with the multi-layer perceptron (MLP) as an illustrative example and subsequently introducing convolutional neural networks (CNN). Afterwards, the meaning of learning is explained and some aspects of classification and regression tasks are discussed. For a comprehensive textbook treatment of these deep learning concepts, [152] is recommended.

### 4.1.1 Multi-layer perceptrons

Simply put, artificial neural networks are just parametrized functions. They can exhibit high complexity, but are usually built from simple elements inspired by the basic computational units of biological brains, namely neurons and synapses. Each neuron performs a simple arithmetic operation on its input, which can be one or a collection of numbers, from which a single output is calculated. More specifically, consider a collection of neurons labeled by the index $j$ connected by synapses to one receiving neuron $i$. Let $x_j \in \mathbb{R}$ be the output of neuron $j$. The associated synapse is given a weight $w_{ij}$ which acts as a multiplicative factor on $x_j$. Neuron $i$ collects

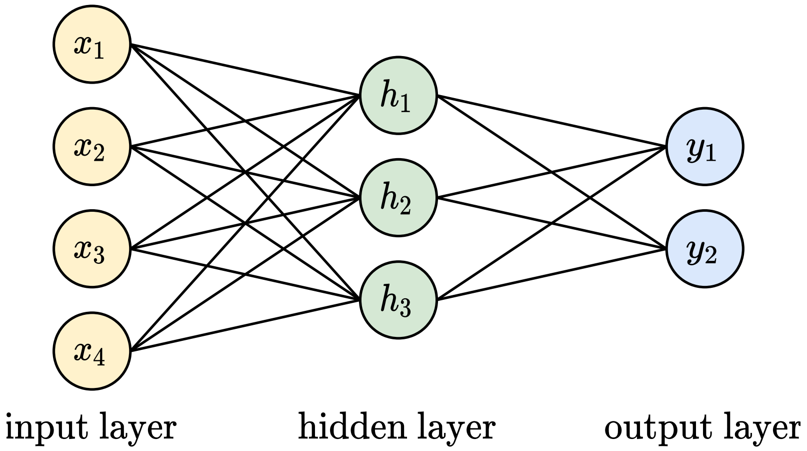


Figure 4.1: Illustration of a standard MLP with one hidden layer.

all outputs in a sum and optionally adds a bias $b_i$. Finally, the result is handed to a non-linear activation function $A(\cdot)$. Taken together, the output $x_i$ is calculated as

$$x_i = A\left(\sum_j w_{ij} x_j + b_i\right) . \tag{4.1}$$

This allows for the construction of in principle arbitrarily structured neural networks by composing many such units in various ways. Here, we restrict ourselves to the case where neurons are organized in layers, with each neuron connected only to neurons of the immediate previous and next layer. Networks of this type are called "shallow" if they consist of only an input and output layer, and "deep" if there also are intermediate ("hidden") layers. This is the basic structure of the MLP, shown schematically in Figure 4.1. Essentially, a MLP performs repeated matrix-vector multiplications with additional non-linearities in between. Without further restrictions on the connectivity, such networks and layers are called fully-connected. The parameters of a MLP are its weights and biases, while e.g. the number of layers and neurons per layer are usually called hyperparameters. In such a network, information can only be processed in one direction, i.e. the transformation is not invertible per se and contains no loops. Architectures of this type are called feedforward networks.

Two of the most important aspects of MLPs are the non-linearity of activations and the utilization of hidden layers, which together allow to distinguish data that are not linearly separable. Without non-linear activation functions, a network consisting of several layers can always be reformulated in terms of a shallow network since all operations are linear transformations. Hence, networks may only be considered "deep" if they exhibit both hidden layers and non-linearities between them, which has been one of the main pillars of deep learning's success. Even though the basic operation performed by a single neuron is just weighted addition, it has been demonstrated that deep neural networks are able to learn extremely complicated transformations. In fact, one can formally prove that under certain conditions, any continuous function can be approximated by a deep neural network with a finite number of neurons [153]. Such proofs are called universal approximation theorems.

In practice, many different types of activation functions are employed. One of the most common choices, which is also used throughout this work, is the rectified linear unit (ReLU) [154] and its variants. It is defined as

$$\mathrm{ReLU}(x) = \max(0, x)\ . \tag{4.2}$$

A popular smooth alternative to this activation with similar properties is given by the SoftPlus function, defined as

$$\mathrm{SoftPlus}(x) = \log(1 + \exp(x))\ . \tag{4.3}$$

Other frequently employed activations include e.g. sigmoid-type functions like tanh or the logistic function, which are used to constrain neuron outputs to fixed intervals.

### 4.1.2 Convolutional neural networks

This section briefly introduces CNNs [155]. They are inspired by the biological structure of the animal visual cortex, where patches of a largely homogeneous distribution of neurons only fire in response to activity in restricted clusters of the visual field. Similarly, convolutional layers in artificial neural networks enforce locality and translational symmetry, and their primary area of application has been in the field of computer vision. From the computational viewpoint, one of the CNN's most important features is its amenability to parallelization on graphical processing units, which can render its evaluation significantly more efficient than the dense matrix-vector multiplications in the fully-connected layers of a MLP. The advent of the CNN has seen tremendous impact in the machine learning community. Its use first led to a significant improvement of earlier performance benchmarks of neural networks in various image classification challenges. Subsequently, it has been established as one of the most important building blocks of modern deep learning architectures; see [156] for a recent review.

In convolutional layers, the same arithmetic operation is applied to different patches of input data on a regular grid, thereby mapping it to an output of the same dimensions. Mathematically, this is still just a matrix-vector multiplication as in fully-connected layers, but with additional constraints enforcing greater sparsity and redundancy in the weight matrix. Consider the one-dimensional case which operates on input vectors $x \in \mathbb{R}^n$. We define the convolution kernel or filter as a set of weights $w \in \mathbb{R}^{m \leq n}$. Furthermore, we introduce a stride parameter $s$ which determines the gap between consecutive applications of the convolution operation. $x$ is mapped to a vector $y$ by

$$y_i = \sum_{j=1}^{m} w_j \cdot x_{s \cdot i + j} + b\ , \tag{4.4}$$

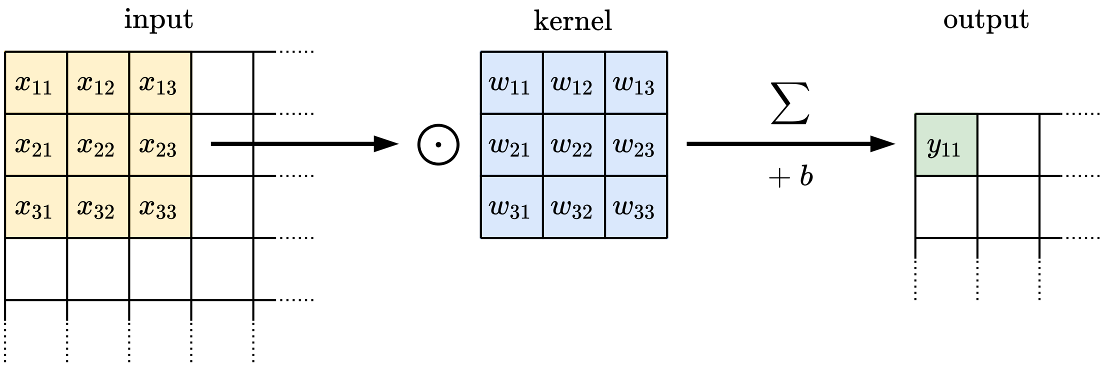


Figure 4.2: Illustration of a convolutional layer for two-dimensional input data.

where $b$ again denotes an optional bias. The above definition straightforwardly generalizes to higher dimensions; an illustration for two-dimensional input data is shown in Figure 4.2. Usually, more than one pair of kernel and bias is applied in the same layer, leading to several outputs. These are called channels, following the naming convention of color channels in images. Accordingly, input data can also have multiple channels already, which may be mixed into a single output channel by applying different kernels and summing all contributions.

An essential advantage of convolutional over fully-connected layers is that the degree to which information is locally mixed can be precisely controlled by the choice of the kernel size. This is important whenever a notion of spatial or temporal locality is present in the input, such as in images or time series data. By taking such prior information about the data into account, the learning of meaningful representations is encouraged. Considering e.g. the task of object recognition, the first layer might learn to identify edges, which may then be clustered into simple shapes, and finally internal representations of whole objects. Learning abstract representations is often additionally facilitated through the use of so-called pooling layers. They reduce the information content by filtering the outputs of several neurons into a single input for the next layer, essentially creating an information bottleneck. Common choices are average pooling, which calculates the mean, as well as max pooling, which singles out the largest value.

### 4.1.3 Learning as an optimization problem

Having discussed the meaning of “deep”, we now move on to define “learning”. Generally speaking, neural networks are made to approximate some desired function by optimizing their weights and biases. This is usually achieved by iteratively minimizing an appropriate loss function $L$ with respect to the model’s parameters using some form of gradient descent. In their most basic implementation, discrete gradient updates take the form

$$w_i' = w_i - \gamma \frac{\partial}{\partial w_i} L(w_i) \ . \tag{4.5}$$

The step size hyperparameter $\gamma$ is commonly called the learning rate. The optimization procedure consists of repeatedly calculating the loss gradients via automatic differentiation [157] and applying a gradient update, which is called training. The gradients are calculated successively from the last to the first layer by applying the chain rule. This procedure is called backpropagation.

Traditionally, optimization was performed on the whole available dataset at once. However, this is computationally expensive and often completely infeasible with the increasingly large datasets available today. Instead, the data is now usually split into randomized batches, and the batch size is considered a hyperparameter. Since the gradients of the loss function computed only on a subset of the data are an approximation of the actual gradient from the whole dataset, this optimization procedure is commonly called stochastic gradient descent. Many improved versions of gradient descent have been developed, such as the popular Adam optimizer [158], which is also used throughout this work. It adds momentum terms to Equation (4.5) and computes individual learning rates $\gamma_i$ for all weights $w_i$ from some base learning rate $\gamma$. It is not clear a priori what are the optimal values for $\gamma$ and other hyperparameters that lead to the best performance for a given problem setting, and they must generally be determined empirically.

In the following, two supervised learning scenarios are illustrated for which neural networks are commonly employed, starting with non-linear regression. Here, the optimization objective is to approximate some desired function that maps input data to $n$-dimensional vectors of real numbers. To achieve this, $n$ neurons without activation functions are used in the last layer of a feedforward network. Denoting their outputs after a forward pass as $y \in \mathbb{R}^n$ and the associated ground truth labels as $\hat{y}$, a common choice for a suitable loss function is the mean squared error, defined as

$$L_{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{4.6}$$

for a single input-output pair. Hence, using $L_{MSE}$ for optimization corresponds to minimizing the Euclidean distance between predictions and ground truth values. If the trained network generalizes well, it should be able to correctly predict labels in close proximity to the ground truth for previously unseen data.

Another common task is the classification of input data into distinct categories. Assuming $n$ different classes, one can use similar networks as for the non-linear regression task, but with an additional SoftMax activation function in the last layer, defined as

$$s_i(y) = \frac{e^{y_i}}{\sum_{j=1}^{n} e^{y_j}} \quad \text{with} \quad \sum_{i=1}^{n} s_i = 1 \ . \tag{4.7}$$

Since the sum of all outputs is always exactly 1, they can be interpreted as probabilities for each class. The ground truth labels are so-called one-hot vectors $\hat{y}$, which are simply $n$-dimensional vectors with a single entry of 1 corresponding to the associated class of a sample and and all other components set to 0. The corresponding

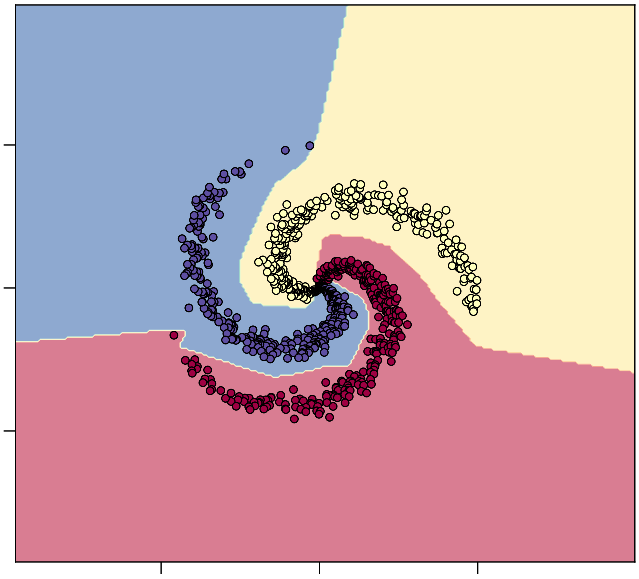


Figure 4.3: Classification of spiral arms using a MLP. Dot colors represent class labels, background colors show the associated prediction landscape.

loss function is the cross entropy, defined as

$$L_{CE} = -\sum_{i=1}^{n} \hat{y}_i \cdot \log(s_i(y)) \ . \tag{4.8}$$

For the case of $n = 2$ (binary classification), it is possible to reduce the number of neurons in the last layer to a single output, using 0 and 1 as the class labels. The activation function is then just the sigmoid-shaped logistic function, of which SoftMax is a generalization. It is defined as

$$\sigma(y) = \frac{1}{1 + e^{-m(y-y_0)}} \ , \tag{4.9}$$

where $y_0$ corresponds to the curve's midpoint and $m$ determines its logistic growth rate or steepness. The loss then simplifies to the binary cross entropy, defined as

$$L_{BCE} = -(\hat{y} \cdot \log(\sigma(y)) + (1 - \hat{y}) \cdot \log(1 - \sigma(y)) \ . \tag{4.10}$$

If the trained network generalizes well, it should assign high probability to the correct class for previously unseen data.

In order to illustrate both the capabilities and limitations of simple deep neural networks, a MLP with two hidden layers is trained to classify spiral arms; see Figure 4.3. This dataset cannot be linearly separated, but the network still manages to correctly identify which arm a given point belongs to by learning reasonable decision boundaries. Nevertheless, the depicted prediction landscape also shows that its performance would break down when trying to classify longer arms than it has seen during training: beyond the most extreme points of the training dataset, the decision boundaries start to propagate linearly away from the center. The issue illustrates one of the pitfalls of many deep learning architectures, namely their inability to properly extrapolate predictions unless explicitly designed to do so. This goes to show that deep learning is not magic, and we have to adjust our expectations of what it can achieve. In most cases, neural networks will try to find the least-effort solution to a given problem if they are not deliberately nudged towards something better.[2] For the particular case of classifying spiral arms, one may potentially achieve this by implementing some form of rotational symmetry into the network, similar to how convolutional layers implement translational symmetry. We will return to this powerful idea in later chapters when deep learning architectures are designed to respect the symmetries of lattice field theories.

Moreover, from the perspective of traditional statistical modeling and optimization theory, it may seem highly counterintuitive that neural networks with many parameters are able to generalize to previously unseen data at all. Naively, it appears that models are in fact completely overparametrized, especially when considering some of the current state-of-the-art architectures whose parameter counts go into the billions [159]. While these models are incredibly successful empirically, theoretical understanding of their success is still the subject of ongoing research [160]. Nevertheless, overfitting does indeed occur when models are trained naively. The simplest strategy to prevent this from happening is to just stop the training as soon as the prediction error on a test dataset is observed to reach a minimum. Other techniques to combat overfitting include e.g. Dropout [161] and regularization by weight decay [162].

## 4.2 Normalizing flows

Beyond their usefulness as predictive models, feedforward neural networks may also be used as trainable components in more complicated transformations, such as normalizing flows used for generative modeling. Flows are a class of probabilistic models for which both efficient sampling and density estimation are made possible using a change-of-variables formula [163–167]. Provably exact sampling that corrects for deviations between the model and target distributions can be obtained with independence Metropolis [168] or reweighting. These may be applied a posteriori, enabling embarrassingly parallel sampling that can provide practical advantages over standard MCMC algorithms.

Normalizing flows are attractive because—among several other interesting features—they have the potential to solve critical slowing down, for reasons that will become clear later. In the context of lattice calculations, they have been successfully applied to model scalar field theory [169–173] as well as $U(1)$ and $SU(N)$ gauge theories [174–178]. In Chapter 6, normalizing flows are developed to model lattice field

[2] Perhaps artificial neural networks are not so different from human brains after all.

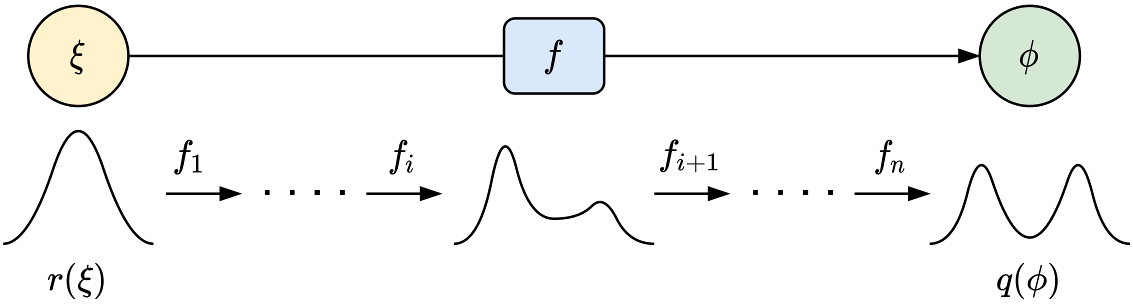


Figure 4.4: Illustration of a normalizing flow composed of $n$ invertible flow layers.

theories with dynamical fermions. Regarding their application to complex action problems, flows have been studied in the context of contour deformations [179, 180]. In Chapter 7, flows are applied to the density-of-states approach. For an in-depth introduction to normalizing flows for lattice field theory with further implementation details and explanations, see [181]. Other machine learning approaches studied in this context are the restricted Boltzmann machine [182, 183], autoregressive networks [184–188], the self-learning Monte Carlo method [189–195], and adversarial learning [196–198]; see also Section 5.1.

Starting with a prior distribution over a continuous space $\mathcal{X}$ with a known probability density $r(\xi)$, an invertible transport map ("flow") $f : \mathcal{X} \to \mathcal{X}, \xi \mapsto \phi$ can be used to redistribute samples $\xi$ under $r(\xi)$ to samples $\phi = f(\xi)$ under a new density $q(\phi)$. We only require that the map be diffeomorphic, i.e. that both $f$ and its inverse are differentiable. The resulting density $q(\phi)$ is fixed by the choice of prior distribution and map. It can be evaluated explicitly as

$$q(\phi) = r(\xi) \left| \det \left( \frac{\partial f}{\partial \xi} \right) \right|^{-1} , \tag{4.11}$$

where $\xi = f^{-1}(\phi)$ and $\det(\partial f / \partial \xi)$ is the Jacobian determinant of $f$. Because the density after the transformation can be computed explicitly, flows provide a mechanism for both sampling and density estimation. Similar to the standard deep neural networks discussed previously, a flow $f$ is often constructed by composing a number of flow layers $f_i$; see Figure 4.4 for an illustration. However, in contrast to the layers of feedforward networks, individual flow layers must be invertible by construction.

By choosing a sufficiently expressive parametrization of $f$, the space of associated transformations (corresponding to a large variational family of model densities $q$) can be explored through numerical optimization in order to find an instance that best approximates some target density $p$. In particular, the parameters of $f$ may be optimized by performing stochastic gradient descent on a measure of the discrepancy between the two densities $q$ and $p$, i.e. an appropriate loss function. A common choice is the Kullback-Leibler divergence [199], which is a measure of the relative entropy

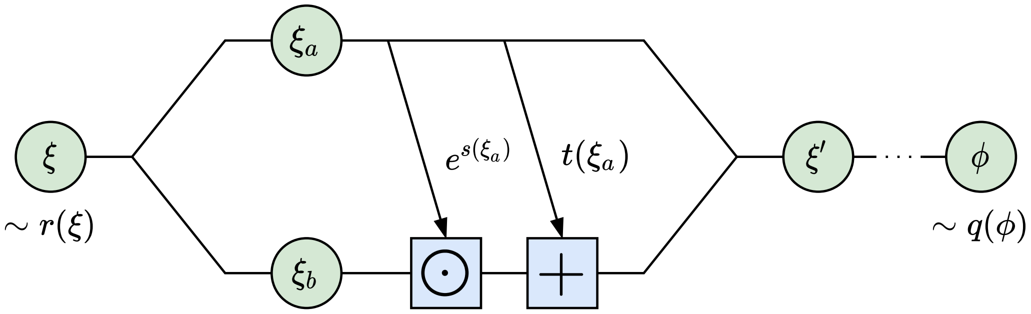


Figure 4.5: Illustration of the affine coupling layer defined in Equation (4.15). The blue boxes depict element-wise operations.

between distributions. It is defined as

$$\begin{aligned} D_{\mathrm{KL}}(q||p) &= \int \mathcal{D}\phi\, q(\phi)(\log q(\phi) - \log p(\phi)) \\ &= \big\langle \log q(\phi) - \log p(\phi) \big\rangle_{\phi\sim q(\phi)} \geq 0 \end{aligned} \tag{4.12}$$

and takes the minimum value $D_{\mathrm{KL}}(q||p) = 0$ iff $q = p$. With expectation values measured using samples from the model distribution $q$, $D_{\mathrm{KL}}(q||p)$ can be stochastically estimated without requiring samples from the target distribution $p$.

For targets of the form $p(\phi) = e^{-S(\phi)}/Z$, the Kullback-Leibler divergence can be evaluated as

$$D_{\mathrm{KL}}(q||p) = \big\langle \log q(\phi) + S(\phi) \big\rangle_{\phi\sim q(\phi)} + \log Z \ . \tag{4.13}$$

When the normalization $Z$ is not known a priori—e.g. for virtually all interesting lattice field theories—$D_{\mathrm{KL}}$ can only be estimated up to the constant $\log Z$. However, this does not affect optimization and one may freely use $(D_{\mathrm{KL}} - \log Z)$ as the loss function for training, which then provides a bound on $\log Z$. Writing the partition function as

$$Z = \int \mathcal{D}\phi\, q(\phi) \frac{e^{-S(\phi)}}{q(\phi)} = \big\langle e^{-S(\phi) - \log q(\phi)} \big\rangle_{\phi\sim q(\phi)} \ , \tag{4.14}$$

it follows that any model giving good agreement to the target distribution necessarily provides a precise, unbiased estimate of $Z$ through model samples alone.

A common building block for the construction of invertible flow transformations is the affine coupling layer. In each such layer, the input $\xi$ is split into two equal-sized subsets $\xi_a, \xi_b$ which are transformed according to

$$\begin{aligned} \xi'_a &= \xi_a \\ \xi'_b &= \xi_b \odot e^{s(\xi_a)} + t(\xi_a) \ , \end{aligned} \tag{4.15}$$

i.e. $\xi_a$ remains unchanged ("frozen") while $\xi_b$ is updated ("active"); see Figure 4.5 for an illustration. Here, the symbol $\odot$ denotes element-wise multiplication. Each affine coupling layer is trivially invertible and has a triangular Jacobian matrix,

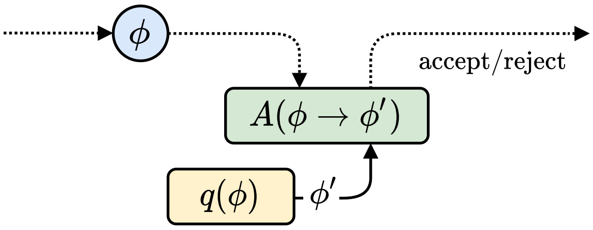


Figure 4.6: Illustration of independence Metropolis sampling for normalizing flows.

thereby making the computation of its determinant and thus the model density $q$ tractable. The context functions $s, t$ that take the frozen variables as inputs and are used to update the active ones can be arbitrary functions. As mentioned above, one may parametrize them with feedforward neural networks, which are then optimized during training. Importantly, they are not required to be invertible themselves, thereby providing much freedom in choosing a particular parametrization. Expressive transformations are built by chaining together many such affine coupling layers with alternating frozen and active subsets.

The simplest and most straightforward approach to utilize a trained flow for sampling from the target distribution is via the aforementioned independence Metropolis algorithm [168]. Starting from some initial configuration $\phi$, the probability to accept a statistically independent proposal $\phi'$ generated by the flow model is defined as

$$\begin{aligned} A(\phi \to \phi') &= \min\left(1, \frac{e^{-S(\phi')}}{e^{-S(\phi)}} \frac{q(\phi)}{q(\phi')}\right) \\ &= \min\left(1, \frac{e^{-S(\phi') - \log q(\phi')}}{e^{-S(\phi) - \log q(\phi)}}\right) \quad , \end{aligned} \tag{4.16}$$

see Figure 4.6 for an illustration. It is instructive to compare Equation (4.16) with Equations (2.2) and (2.3) as well as Figure 4.6 with Figure 2.1.

From the second line of Equation (4.16), one can immediately see why flow-based sampling has the potential to solve critical slowing down, simply because it allows large steps to be taken in configuration space. With conventional MCMC algorithms, the large action differences associated with such steps render transitions to greater action values extremely unlikely because the associated acceptance probabilities are strongly suppressed. With flow-based sampling, these differences can be compensated by the model probabilities, provided that the target distribution is modeled sufficiently well such that $S(\phi) + \log q(\phi)$ is approximately constant as a function of $\phi$. Hence, reaching high acceptance rates is not anymore a question of distance in configuration space, but rather a question of model quality. By investing the bulk of the computational effort into learning and evaluating a high-quality flow instead of making small updates to a Markov chain, a significant advantage over traditional sampling algorithms may be achieved. Furthermore, the flow-based approach allows embarrassingly parallel sampling of the model distribution. An asymptotically exact Markov chain can then be constructed in a trivial post-processing step.

## 4.3 Gaussian process regression

This section serves as a brief introduction to Gaussian process regression (GPR). Starting from early developments in the context of geostatistics in the 1950s [200], today GPR is widely employed in a variety of settings for the probabilistic modeling of functions from a finite number of measurements; see [201, 202] for recent reviews. For a modern, comprehensive textbook treatment of the topic, see [203]. For a pedagogical introduction with simple code examples, [204] is highly recommended. The notation used in Section 3.3 is adopted for consistency, however, the general formalism presented here is also applicable outside of the specific context of spectral reconstruction. In contrast to the topics of the previous sections, the method described here is not a deep learning algorithm. However, there has been considerable work in the machine learning community connecting both subjects.

GPR is discussed here for the case where direct observations are available for the function to be modeled. We assume our knowledge of a function $\rho(\omega)$ to be encoded in a GP with mean and covariance functions $\mu(\omega), C(\omega, \omega')$, denoted by

$$\rho(\omega) \sim \mathcal{GP}\left(\mu(\omega), C(\omega, \omega')\right) \ . \tag{4.17}$$

The covariance is assumed to be symmetric, i.e. $C(\omega, \omega') = C(\omega', \omega)$. As per the definition of a GP, any finite set of function evaluations at $N$ sample points $\omega_i$ follows a multivariate normal distribution,

$$\begin{pmatrix} \rho(\omega_1) \\ \vdots \\ \rho(\omega_N) \end{pmatrix} \sim \mathcal{N}\left( \begin{pmatrix} \mu(\omega_1) \\ \vdots \\ \mu(\omega_N) \end{pmatrix}, \begin{pmatrix} C(\omega_1, \omega_1) & \dots & C(\omega_1, \omega_N) \\ \vdots & \ddots & \vdots \\ C(\omega_N, \omega_1) & \dots & C(\omega_N, \omega_N) \end{pmatrix} \right) \ . \tag{4.18}$$

Similarly, we can write down the joint distribution of a set of observations $\hat{\rho}_i$ at points $\hat{\omega}_i$ and the value of $\rho$ at an arbitrary point $\omega$ as

$$\begin{pmatrix} \rho(\omega) \\ \hat{\boldsymbol{\rho}} \end{pmatrix} \sim \mathcal{N}\left( \begin{pmatrix} \mu(\omega) \\ \hat{\boldsymbol{\mu}} \end{pmatrix}, \begin{pmatrix} C(\omega, \omega') & \hat{\mathbf{C}}^T(\omega) \\ \hat{\mathbf{C}}(\omega') & \hat{\mathbf{C}} + \sigma_n^2 \cdot \mathbf{1} \end{pmatrix} \right) \ , \tag{4.19}$$

where boldface type denotes vector and matrix quantities. Here, we have defined $\hat{\boldsymbol{\mu}} \equiv \mu(\hat{\omega}_i)$, $\hat{\mathbf{C}}_i(\omega) \equiv C(\hat{\omega}_i, \omega)$, and $\hat{\mathbf{C}}_{ij} \equiv C(\hat{\omega}_i, \hat{\omega}_j)$. $\sigma_n^2$ quantifies the point-wise variance of additional measurement noise which may be present in the observations $\hat{\boldsymbol{\rho}}$. Due to the inherent analytic tractability of the normal distribution, one can derive the posterior distribution of function values conditioned on observations as

$$\begin{aligned} \rho(\omega)|\hat{\boldsymbol{\rho}} \sim \mathcal{N}\Big( & \mu(\omega) + \hat{\mathbf{C}}^T(\omega) \left(\hat{\mathbf{C}} + \sigma_n^2 \cdot \mathbf{1}\right)^{-1} (\hat{\boldsymbol{\rho}} - \hat{\boldsymbol{\mu}}) \ , \\ & C(\omega, \omega') - \hat{\mathbf{C}}^T(\omega) \left(\hat{\mathbf{C}} + \sigma_n^2 \cdot \mathbf{1}\right)^{-1} \hat{\mathbf{C}}(\omega') \Big) \ . \end{aligned} \tag{4.20}$$

In order for GPs to be useful for modeling, the covariance $C(\omega, \omega')$ is commonly defined via a so-called kernel function. One may encode any prior beliefs about the shape of plausible solutions for a given problem by choosing an appropriate form, which is then usually parametrized by a small number of hyperparameters. For an introduction to constructing GP kernels of various types as well as strategies to apply and combine them, see the kernel cookbook [205]. The mean function $\mu(\omega)$ is often set to zero, since its contribution can be fully absorbed by the kernel. Typically, the latter is the sole focus of the optimization procedure. However, a custom mean function may still be useful in certain situations in order to incorporate prior beliefs about the functional form of the expected solution. This can improve the calculation by providing a better starting point for the prediction.

The mean and covariance of the GP posterior often depend strongly on the particular values of the kernel's hyperparameters. An optimal choice for these hyperparameters, denoted here by $\hat{\boldsymbol{\alpha}}$, may be obtained by maximizing the associated likelihood,

$$\begin{aligned} p(\hat{\boldsymbol{\rho}}|\boldsymbol{\alpha}) = &\left((2\pi)^N \det\left(\hat{\mathbf{C}}_\alpha + \sigma_n^2 \cdot \mathbf{1}\right)\right)^{-\frac{1}{2}} \cdot \\ &\exp\left(-\frac{1}{2}(\hat{\boldsymbol{\rho}} - \hat{\boldsymbol{\mu}})^T \left(\hat{\mathbf{C}}_{\boldsymbol{\alpha}} + \sigma_n^2 \cdot \mathbf{1}\right)^{-1} (\hat{\boldsymbol{\rho}} - \hat{\boldsymbol{\mu}})\right) \ , \end{aligned} \tag{4.21}$$

where we have written $\hat{\mathbf{C}}_{\hat{\boldsymbol{\alpha}}}$ to emphasize the dependence on the hyperparameters. Instead of directly maximizing $p(\hat{\boldsymbol{\rho}}|\boldsymbol{\alpha})$ as a function of $\hat{\boldsymbol{\alpha}}$, one conventionally minimizes the negative log likelihood (NLL),

$$\begin{aligned} -\log p(\hat{\mathbf{f}}|\boldsymbol{\alpha}) = &\frac{1}{2}(\hat{\boldsymbol{\rho}} - \hat{\boldsymbol{\mu}})^T \left(\hat{\mathbf{C}}_{\boldsymbol{\alpha}} + \sigma_n^2 \cdot \mathbf{1}\right)^{-1} (\hat{\boldsymbol{\rho}} - \hat{\boldsymbol{\mu}}) \\ &+ \frac{1}{2} \log\det\left(\hat{\mathbf{C}}_\alpha + \sigma_n^2 \cdot \mathbf{1}\right) + \frac{N}{2} \log 2\pi \ . \end{aligned} \tag{4.22}$$

However, since simply finding and employing the maximum likelihood configuration of hyperparameters may ignore relevant additional structures in the posterior distribution, one can also integrate out $\hat{\boldsymbol{\alpha}}$ using suitable hyperpriors to account for some variability.

A frequently used kernel parametrization is the radial basis function (RBF) kernel, also called squared exponential. It is defined as

$$C(\omega, \omega') = \sigma_C^2 \exp(-\frac{(\omega - \omega')^2}{2l^2}) \ , \tag{4.23}$$

where $\sigma_C$ encodes the overall magnitude and $l$ is a generic length scale controlling the frequency of fluctuations. This kernel has been established as the standard choice for many applications due to a number of attractive features, such as universality [206] and every function in its prior being infinitely differentiable. It is suitable for the prediction of smooth functions without any discontinuities in the first few derivatives. The RBF kernel is also used for the results on spectral reconstruction presented in this work. An example for function prediction with GPR using this kernel is shown in Figure 4.7. Other frequently used parametrizations include the rational quadratic kernel, locally or globally periodic kernels, and the Matérn covariance [207].

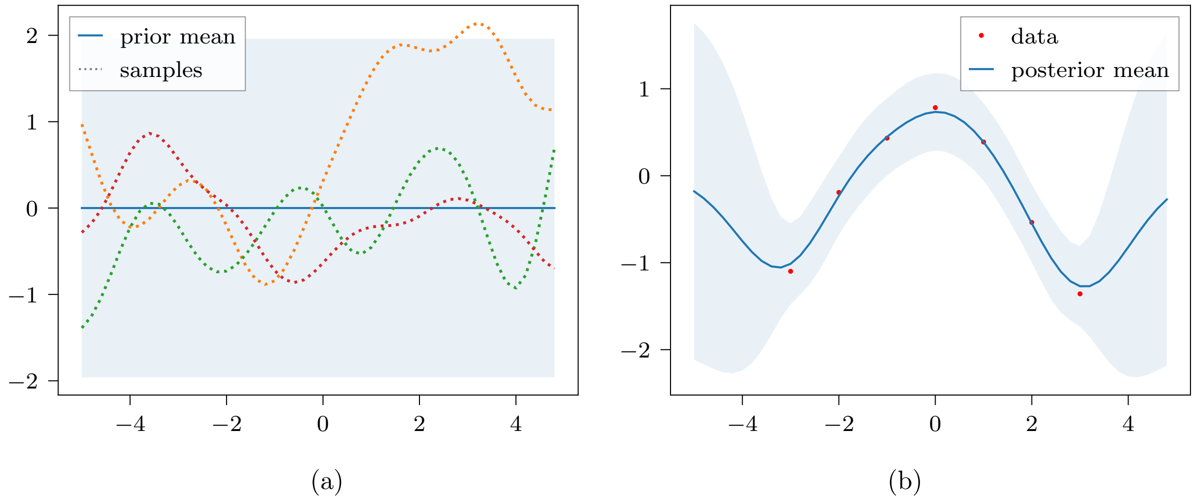


Figure 4.7: (a) Flat GP prior with three sample functions. (b) GP posterior mean and standard deviation for a small number of data points with finite uncertainty.

Based on the above formulation of GPR for the modeling of direct observations $\hat{\boldsymbol{\rho}}$ at points $\hat{\boldsymbol{\omega}}$, one can also derive closed-form expressions for inference from indirect observations $\hat{\mathbf{G}}$ at points $\hat{\mathbf{p}}$ that are generated from $\rho(\omega)$ by a linear forward process. This is possible because linear transformations preserve Gaussian statistics. Hence, based on this insight one may predict solutions to linear inverse problems. The modified procedure involves all terms related to the observations that depend on the discrete set of points $\hat{\boldsymbol{\omega}}$, which are promoted back to the continuous domain and subsequently integrated out to yield the nodes $\hat{\mathbf{p}}$ instead. In this work, the linear forward process under consideration is of course the KL integral defined in Equation (3.4). The regularized inversion of this integral transformation using GPR is discussed in further detail and applied to the computation of ghost and gluon spectral functions in Chapter 8.

# 5 AI for lattice field theory: a first look

In this chapter, we take a first look at the use of modern machine learning techniques in lattice field theory, with some of the computational barriers of Chapter 3 in mind. The following two applications are presented:

- In Section 5.1, we discuss a first attempt at devising a generative neural sampling algorithm for lattice calculations;
- In Section 5.2, interpretable deep learning is investigated for the extraction of unknown relevant observables from lattice data.

These works represent some of the earliest applications of machine learning in this domain. Although many of the associated ideas and components have since been overhauled and specialized much further, the conceptual developments therein continue to guide and inspire current research efforts. Hence, it is worthwhile to revisit these early ideas for harnessing the power of AI to solve the major computational problems that we face.

## 5.1 Neural sampling with GANs

Following a brief introduction in Section 5.1.1, generative adversarial networks (GAN) are discussed in Section 5.1.2. The proposed sampling algorithm is developed in Section 5.1.3. Numerical results are presented in Section 5.1.5 and a rough comparison of the computational cost is provided in Section 5.1.6. We conclude with a summary and outlook in Section 5.1.7. The contents of this section have been published in [1] together with Jan M. Pawlowski.

### 5.1.1 Introduction

In order to tackle critical slowing down in lattice calculations, promising new approaches based on generative machine learning methods are currently being explored. An interesting candidate for this purpose is the GAN [208], which has received much attention in the machine learning community. By construction, the generated samples are statistically independent. Hence, there are in principle no autocorrelations if they are arranged in a Markov chain. This makes GANs attractive for a first look at designing a more efficient sampling approach based on machine learning. However, simply replacing a MCMC algorithm with a GAN is problematic for several reasons. Most importantly, the learned distribution typically shows non-negligible deviations from the desired target and the model probabilites cannot be tractably computed.

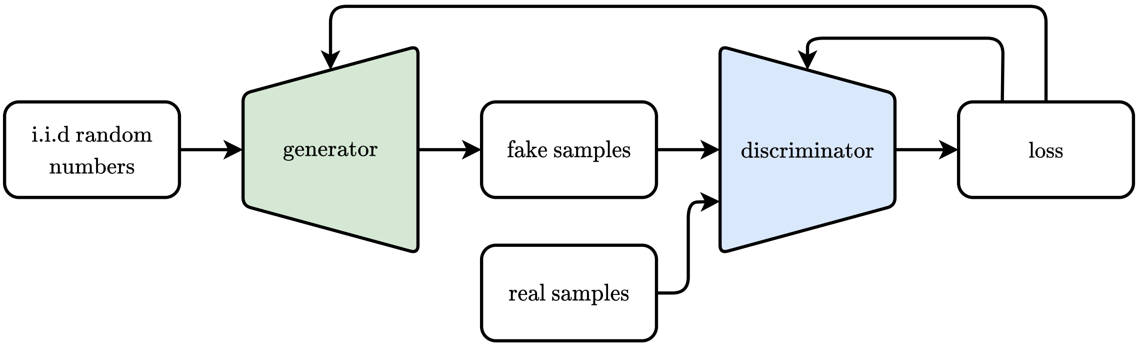


Figure 5.1: Schematic of a GAN's components and data flow. Random numbers are passed to the generator to produce fake samples. The discriminator learns to distinguish between real and fake. Training is performed by backpropagating loss gradients through both networks and updating weights in an alternating fashion.

This casts some doubt on the reliability of such an approach. Moreover, even if the approximation by the network exhibits high precision, one cannot simply assume that sampling from it is sufficiently ergodic. Implementing these key properties in a conclusive manner is essential for accurate and reliable lattice computations.

In a first attempt to achieve this, a hybrid algorithm is proposed where the GAN is implemented as an overrelaxation step in combination with conventional HMC. This approach effectively breaks the Markov chain for observables unrelated to the action, thereby potentially leading to a significant reduction in the associated autocorrelation times. However, it is unclear whether asymptotic exactness can be guaranteed with this method. Already at this point, it should be emphasized that this is not an issue for the more sophisticated flow-based sampling algorithms discussed in later chapters. For the purpose of this appetizer, we shall simply ignore these potential problems and just press on.

The approach is demonstrated in the context of real, scalar $\phi^4$-theory in two dimensions. Aside from the aforementioned statistical concerns, the results show that novel sampling algorithms based on machine learning are feasible, and motivate further research into the matter.

### 5.1.2 Generative adversarial networks

GANs belong to a class of unsupervised, generative machine learning methods based on deep neural networks. The characteristic feature that distinguishes them from many other architectures is the utilization of game theory principles for their training. They consist of two consecutive feedforward neural networks, namely the generator $G$ and discriminator $D$; see Figure 5.1 for a schematic. The generator computes samples $G(\xi)$ from random inputs $\xi$ drawn from a multi-variate prior distribution $r(\xi)$. The discriminator receives these generator outputs as well as samples $\phi$ from a training dataset sampled from the target distribution $p(\phi)$. $D$ is a binary classifier that is trained to distinguish between 'real' and 'fake' samples. Its last layer consists of a single neuron with a sigmoid activation. The binary cross-entropy defined in Equation (4.10) is used as the loss function. Training corresponds to minimizing the loss separately for $D$ and $G$ using opposite ground truth labels, respectively. This is achieved by computing gradients via backpropagation through both networks and applying gradient descent in an alternating fashion. In intuitive terms, the optimization objective for the discriminator is to maximize its classification accuracy, while the generator is trained to produce samples that cause false positive predictions in the discriminator. The two networks play a zero-sum non-cooperative game, and the model is said to converge when they reach so-called Nash equilibrium. If successful, the generator approximates the true target distribution $p(\phi)$.

Since $r(\xi)$ is commonly chosen to be a simple multi-variate uniform or normal distribution from which one can easily obtain i.i.d. samples, candidate configurations drawn from $G$ are statistically independent by construction. This can provide a potential advantage over traditional MCMC algorithms. However, in practice one encounters deviations of varying severity between the model distribution of the generator and the true target distribution. Furthermore, GANs may not be sufficiently ergodic in order to perform reliable lattice calculations. This is especially problematic in the case of so-called mode collapse, where the generator learns to produce only one or a very small number of samples largely independent of its prior. Insufficient variation among the GAN output is not punished by the discriminator and can only be checked a posteriori. While a number of improved approaches to deal with such issues has been proposed in the machine learning literature [209], one may question whether GANs can indeed satisfy stringent ergodicity requirements. Interestingly, it was shown that they can act as reliable pseudo-random number generators, outperforming several standard, non-cryptographic algorithms [210].

### 5.1.3 Sampling algorithm

Using GANs to perform lattice calculations requires implementing a suitable selection procedure to generate new samples in the Markov chain. Naively, one could just try to use the Metropolis accept/reject step defined in Equation (2.3), evaluating only the action differences $\Delta S$. However—even without worrying about violations of asymptotic correctness due to ignoring the model probabilities of the samples—this would not work in practice, simply because candidate configurations are accepted either automatically if $\Delta S \leq 0$ or with probability $\exp(-\Delta S)$ if $\Delta S > 0$. Accordingly, such an algorithm only performs well if changes in the action are not too large. Usually, MCMC updates can be tuned to avoid this problem and achieve reasonable acceptance rates. For configurations from a GAN, large positive and negative values for $\Delta S$ would be very common, since subsequent samples are uncorrelated. Hence, this algorithm would effectively freeze at the lower end of the available action distribution after a short time. Jumping to larger actions would be exponentially suppressed, leading to a vanishing acceptance rate.

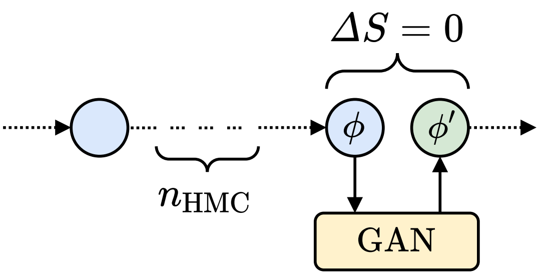


Figure 5.2: Illustration of the hybrid algorithm using a GAN overrelaxation step.

In order to avoid the aforementioned issues, one possibility is to use the GAN to implement an overrelaxation step, which can then be combined with any action-based importance sampling algorithm. Overrelaxation was originally designed for lattice calculations of $SU(2)$ and $SU(3)$ Yang-Mills theory by exploiting symmetries of the action [211]. It is based on the fact that a candidate configuration is automatically accepted in the Metropolis step if $\Delta S = 0$, under the condition that the a priori selection probability $A_0$ is symmetric. This is achieved by performing certain transformations of the gauge links that leave $S$ unchanged. By itself, overrelaxation is therefore not ergodic, since it operates on hypersurfaces of constant action. Ergodicity is achieved by combining it with a standard MCMC algorithm. In this manner, autocorrelations can be reduced substantially while still asymptotically approaching the correct distribution of the theory.

The method employed here for selecting suitable candidate configurations is based on [212], where GANs are proposed as an ansatz to the more general task of solving inverse problems. The approach is implemented with the following procedure (see Figure 5.2 for an illustration):

1. Take a number of HMC steps $n_{\text{HMC}}$ to obtain a configuration $\phi$;

2. Pre-sampling step: sample from the GAN until a configuration $G(\xi)$ is found that fulfills $|\Delta S| = |S[G(\xi)] - S[\phi]| \leq \epsilon$. $\epsilon$ can be optimized to minimize the total time required by the procedure;

3. Gradient flow step: perform a gradient descent of the associated latent variable $\xi$ using $\Delta S^2$ as the loss, i.e. $\xi' = \text{argmin}_{\xi}(S[G(\xi)] - S[\phi])^2$. $\phi' = G(\xi')$ is the new configuration after the gradient flow.

In this manner, $S[\phi]$ and $S[\phi']$ can ideally be matched arbitrarily well, down to the available floating point precision. The action values can then be considered effectively equal for all intents and purposes. In principle, gradient descent can be performed for any randomly drawn $\xi$ without the need for the second step. The additional pre-sampling simply ensures that the distance in the latent space between the initial $\xi$ and the final $\xi'$ is already small a priori, which can speed up the gradient flow and avoids the risk of getting stuck in a local minimum of the loss landscape.

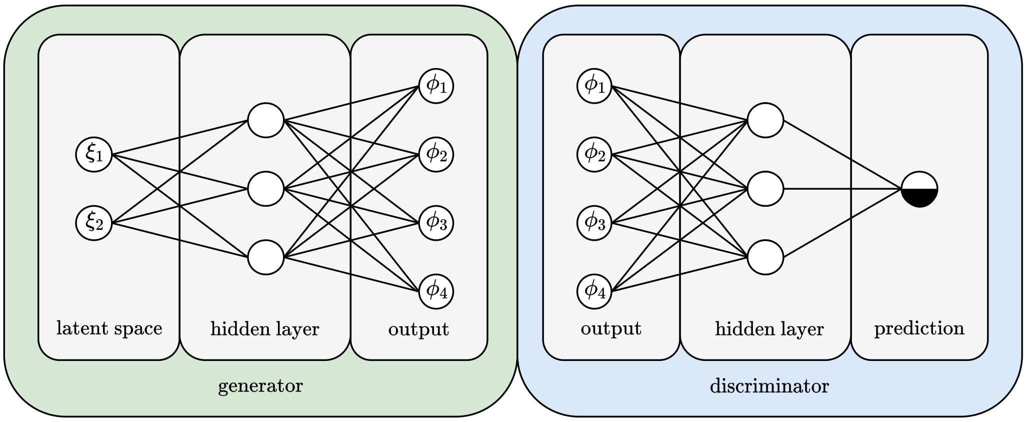


Figure 5.3: Illustration of the vanilla GAN architecture with fully-connected layers.

The optimal choice of $\epsilon$, the gradient descent step size $\gamma$, as well as the sampling batch size, can be determined through a hyperparameter optimization.

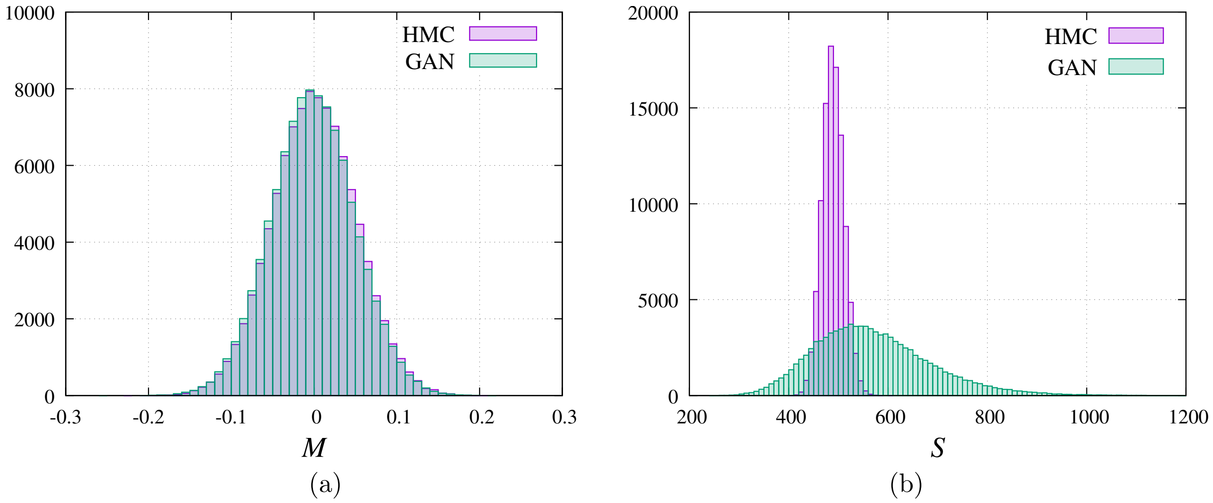
The potential improvement of the proposed hybrid algorithm over standard MCMC methods is that it can effectively break the Markov chain for observables other than the action while preserving some essential properties. By implementing the overrelaxation step in combination with standard HMC, it is also not necessary for the GAN to approximate the target distribution to an exceedingly high precision. Another important aspect is the applicability to a much wider variety of theories compared to conventional overrelaxation, since the algorithm does not depend on specific symmetries of the action.

### 5.1.4 Model details

To put these ideas to the test, a GAN with fully-connected layers is trained on field configurations from real, scalar $\phi^4$-theory as defined in Section 2.2 on a $32 \times 32$ lattice, using 1000 samples generated in the symmetric phase at $\kappa = 0.21$ and $\lambda = 0.022$. For context, note that the phase transition occurs at roughly $\kappa \approx 0.27$ (with fixed $\lambda = 0.022$) [213]. The generator has an input layer with 256 neurons and one hidden layer of size 512 is chosen for both the generator and the discriminator; see Figure 5.3 for an illustration of this architecture. The generator's last layer has no activation function, thereby allowing values $G(\xi) \in \mathbb{R}^{32\times 32}$. As mentioned above, the discriminator's output neuron features a sigmoid activation to allow binary classification. For all other layers, the ReLU activation is used.

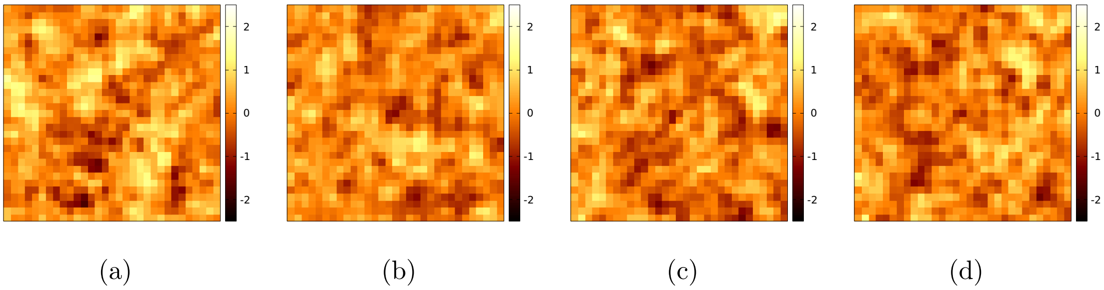
### 5.1.5 Results

The distributions of $M$ and $S$ computed with configurations sampled independently from the GAN are compared to the HMC baseline in Figure 5.4. The distribution of magnetization values already matches the baseline remarkably well. However, the distribution of action values is considerably broader, indicating that the GAN has



(a) (b)

Figure 5.4: Comparison of magnetization (a) and action (b) distributions with $10^5$ samples generated using HMC and the GAN.

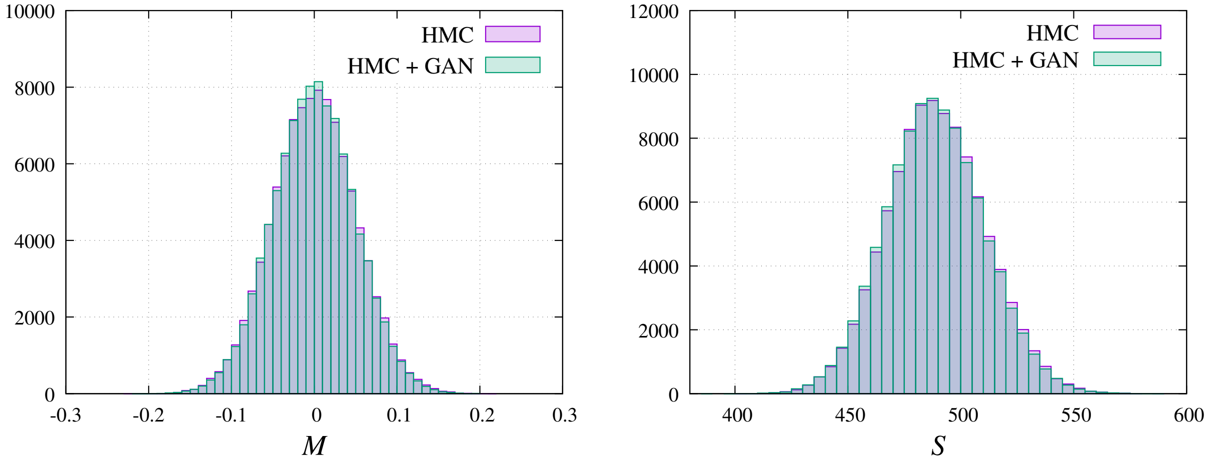


(a) (b) (c) (d)

Figure 5.5: (a) Field configuration sampled with HMC. (b-d) Samples with the same value of the action as (a), generated with the GAN overrelaxation step.

not managed to fully capture all relevant features of the theory. This difference is compensated by the proposed hybrid algorithm, as discussed in the following.

First, the GAN's ability to reproduce every desired action value is verified with field configurations which were not part of the training dataset. Here, $\epsilon = 1$ is used for the pre-sampling step and a step size of $\gamma = 10^{-5}$ for the gradient descent. In the subsequent runs, the GAN is always able to produce samples with matching actions. Figure 5.5 shows a sample from HMC and three corresponding proposals generated with the overrelaxation method. The distributions of $M$ and $S$ obtained with the modified sampling algorithm using $n_{\mathrm{HMC}} = 3$ are consistent with results from the HMC baseline, see Figure 5.6. In particular, the difference between the action distributions that was visible in Figure 5.4 has disappeared completely. This is expected, since in the hybrid algorithm the actions are first generated with HMC, and the subsequent overrelaxation step leaves $S$ invariant.



(a)

(b)

Figure 5.6: Comparison of magnetization (a) and action (b) distributions with $10^5$ samples generated using HMC and the proposed hybrid algorithm.

### 5.1.6 Computational cost

In order to facilitate a rough comparison of the computational cost, both the HMC update and the GAN overrelaxation step are implemented using the same software framework, namely the deep learning library PyTorch [214]. All steps of the computation except for the recording and monitoring functionality are performed on an Nvidia GeForce GTX 1070. The average time required to accept one HMC trajectory is measured to be 42 ms.

For the GAN overrelaxation, the time needed depends strongly on the aforementioned hyperparameters. In the pre-sampling step, one needs to consider the behavior with respect to the batch size and $\epsilon$. Larger batches require more time, but are more likely to contain suitable samples for small values of $\epsilon$. On the contrary, if $\epsilon$ is larger, it is increasingly likely for any given sample to satisfy the criterion, and small batch sizes are sufficient. In contrast, the gradient flow step is always completed faster for smaller $\epsilon$. It also generally depends on the discrete step size $\gamma$, but different choices of this parameter are found to have only a weak effect on the overall time. An optimal trade-off determined via a hyperparameter scan is found to be at $\epsilon = 10^{-2}$ with a batch size of $10^3$ and $\gamma = 10^{-6}$. With these values, the sampling and gradient flow step require on average 53 ms and 64 ms, respectively, yielding a combined time of 117 ms. While this is almost three times the duration of evaluating one HMC trajectory, it must be contrasted against the potential advantages by breaking autocorrelations of observables in the Markov chain. An in-depth study of the achievable gains is beyond the scope of this first attempt, however, some potential improvements to the proposed hybrid algorithm are discussed in the following summary.

### 5.1.7 Summary and outlook

Deep neural networks were investigated for the purpose of improving lattice calculations by generating independent field configurations. A simple hybrid algorithm was constructed based on a combination of standard HMC and an overrelaxation step implemented with a GAN. Distributions of basic observables generated by this algorithm were observed to be consistent with the HMC baseline.

The time benchmarks and hyperparameter optimization suggest that the computational cost of the method could be further reduced by a simple modification to the pre-sampling step. Here, most configurations are ignored until one with $|\varDelta S| < \epsilon$ is found. This repeated generation of samples which are immediately discarded again is inefficient. Instead, one should maintain a reservoir of configurations from the GAN, together with their associated actions. By repopulating this reservoir periodically, it can be guaranteed that appropriate samples are always available for the gradient flow step. With a conditional GAN [215], it may also be possible to reduce $\varDelta S$ a priori by using the target value of the action as the conditional parameter. Training a conditional GAN at different action parameters may also allow extrapolation to other regions of the phase diagram where no training data has been generated [198].

In summary, this work provides a first example of a novel sampling algorithm for lattice calculations based on machine learning. The results presented in this section motivate further research into generative neural samplers based on more sophisticated architectures, which will be the subject of Chapters 6 and 7.

## 5.2 Novel insights from interpretable AI

In this section, the extraction of relevant observables from lattice data via interpretable AI is investigated. After a brief introduction in Section 5.2.1, we discuss interpretable machine learning techniques in general and for this work specifically in Section 5.2.2. Results are presented in Section 5.2.3 and we conclude in Section 5.2.4. Further information and implementation details can be found in Appendix A. The contents of this section have been published in [3] together with Stefan Blücher, Lukas Kades, Jan M. Pawlowski, and Nils Strodthoff.

### 5.2.1 Introduction

It is well known that the occurrence of numerical sign problems in lattice calculations is formulation-dependent. In many cases, expressing the theory with different degrees of freedom removes the problem completely. However, no such alternative formulation has been found for lattice QCD at finite chemical potential. In this context, it may be instructive to search for so far unidentified structures in the data. Finding previously unknown observables that are characteristic for the theory under study may then inform the construction of a novel formulation.

One ansatz for the identification of relevant observables is through representation learning, i.e. by training a machine learning architecture with a pretext task. The rationale behind this approach is that the model learns to recognize patterns which can be leveraged to construct observables from low-level features. However, solving a given task does by itself not lead to novel physical insights, since the inner structure of the algorithm typically remains opaque. This issue can at least partially be resolved by the use of interpretable AI techniques, which have recently attracted considerable interest in the machine learning community. In this work, we focus on layer-wise relevance propagation (LRP) [216]. It is one of several popular post-hoc attribution methods that propagate the prediction back to the input domain, thereby highlighting features that influence the algorithm towards/against a particular classification decision.

This approach is tested in the context of three-dimensional scalar Yukawa theory with two mass-degenerate flavors of staggered fermions as discussed in Section 2.4, using the dimensionless formulation of the action for the scalar field as defined in Equation (2.8). Inference of an action parameter is considered as a pretext task in order to identify relevant observables. In a first step, it is demonstrated that this is at least partially possible by training an MLP on a set of standard observables. Here, it is shown that the relevance of features in different phases, as determined by LRP, agrees with physical expectations. The results are benchmarked against a similar method based on random forests. Subsequently, it is demonstrated that the action parameter can be inferred directly from field configurations using a CNN. LRP is employed to identify relevant filters and discuss how these align with physical knowledge. This also allows the construction of a novel observable that appears to be a distinctive feature of the paramagnetic phase.

### 5.2.2 Interpretable AI

Simple methods from statistics and machine learning often lack the capability to model complex data, whereas sophisticated algorithms typically tend to be less transparent. A commonly used algorithm is principal component analysis (PCA). It has been successfully applied to the extraction of several (already known) order parameters for various systems [217–219]. However, its linearity prohibits the identification of more complicated quantities, e.g. Wilson loops in gauge theories [220]. Hence, we require tools capable of modeling non-linear features, such as deep neural networks. They allow for a more comprehensive treatment of complex systems, which has been demonstrated e.g. for fermionic theories in [221, 222]. The approach also enables new ways to locate phase transitions in a semi-supervised manner, such as learning by confusion and similar techniques [223, 224]. For lattice QCD, action parameters can be extracted from field configurations [225].

Overall, deep learning tools seem particularly well-suited to grasp relevant information about lattice field configurations in a completely data-driven approach, by learning abstract internal representations of relevant features. However, their lack of transparency is frequently a major drawback of using such methods, which prohibits access to and comprehension of these representations. A unified understanding of how and what these architectures learn, and why it seems to work so well in a wide range of applications, is still pending. To better understand the processes behind phase classification with neural networks in lattice models, multiple proposals have been made, such as pruning [220, 226, 227], utilizing (kernel) support vector machines [228, 229], and saliency maps [230].

Moreover, in the broader scope of machine learning research, there has been growing interest in interpretability approaches, most of them focusing on post-hoc explanations for trained models. So-called attribution methods typically assign a relevance score to each of the input features, thereby showing what the classifier was particularly sensitive to, or what influenced the algorithm towards/against a classification decision. In the domain of image recognition, such attribution maps are typically visualized as heatmaps overlaying the input image. The development of attribution algorithms is a very active field of research in the machine learning community. Therefore, the interested reader is referred to dedicated research articles for more in-depth treatments [231, 232]. Interpretable AI in general is also an important topic outside of purely academic research, in particular for safety-critical applications like self-driving cars, medical diagnosis, and crime prevention. In order to promote trust in AI technologies among the general population, an improved understanding of the inner workings of these machines is essential.

Very broadly, the most important types of such local interpretability methods can be categorized as: 1. Gradient-based, such as saliency maps [233] obtained by computing the derivative of a particular class score with respect to the input features or integrated gradients [234]. 2. Decomposition-based, such as layer-wise relevance propagation (LRP) [216] or DeepLift [235]. 3. Perturbation-based, as in [236], investigating the change in class scores when occluding parts of the input.

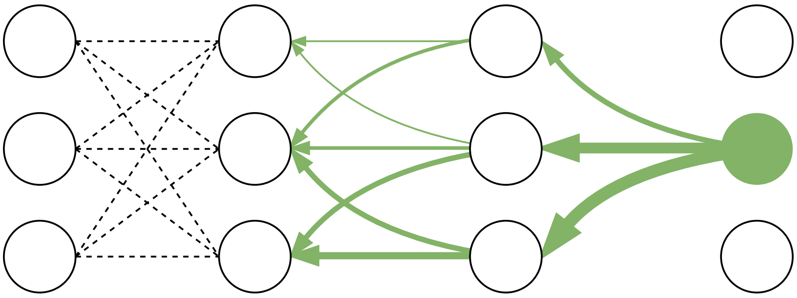


Figure 5.7: Illustration of LRP through the last two layers of a classification network that predicts one-hot vectors. Relevance is indicated by arrow width. The conservation law requires the sum of widths to remain constant during backpropagation.

In this work, we focus on LRP, a particular variant of decomposition-based attribution methods, which has been successfully applied to other problems in physics and chemistry, e.g. in the context of atomistic systems [237]. Nevertheless, it should be stressed that qualitative findings are expected to agree for all decomposition- and gradient-based methods [238]. The general idea of LRP is to start from a relevance assignment in the output layer and subsequently propagate this relevance back to the input using certain propagation rules; see Figure 5.7 for an illustration and Appendix A.2 for details. The method thereby assigns a relevance score to each neuron, where positive (negative) entries influence the classifier towards (against) a particular decision.

### 5.2.3 Results

In this section, numerical results for three-dimensional Yukawa theory are presented. A MLP and a CNN are trained to infer the associated hopping parameter $\kappa$ from a set of known observables (Approach A), as well as solely from the raw field configurations (Approach B), akin to [225]. In the first case, without providing any prior knowledge of the phase boundaries, LRP manages to reveal the underlying phase structure and returns a phase-dependent importance hierarchy of the observables in accordance with physical expert knowledge. In the second case, by calculating the relevances of the learned filters, one can associate each of them with one of the physical phases and thereby extract the known order parameters. Moreover, it facilitates the construction of a novel observable that characterizes the symmetric phase. Both variants of this strategy are sketched in Figure 5.8. Since the action parameter prediction is an ill-conditioned inverse problem, the optimization objective is formulated in terms of maximum likelihood estimation. Assuming a Gaussian distribution with fixed variance, this objective reduces to minimizing the MSE, which we use as the loss function in the following. In addition, weight regularization is applied, see Appendix A.4 for details.

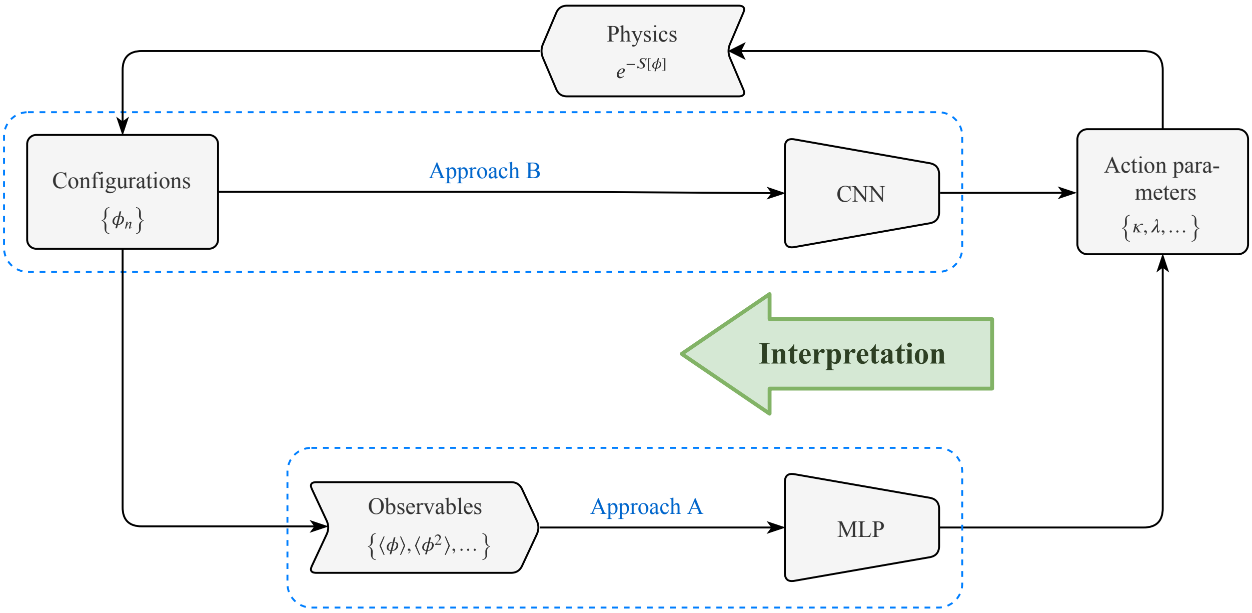


Figure 5.8: Diagram of the strategy to learn meaningful structures from lattice data by analyzing networks trained for action parameter inference. Field configurations used for training are either preprocessed into observables for the MLP (Approach A) or directly used for a CNN (Approach B). Obtaining accurate predictions for the parameters indicates approximate cycle consistency in the above diagram, which supports the notion that the networks have successfully identified characteristic features. These can then be extracted in a subsequent interpretation step using LRP.

### Importance hierarchies of known observables

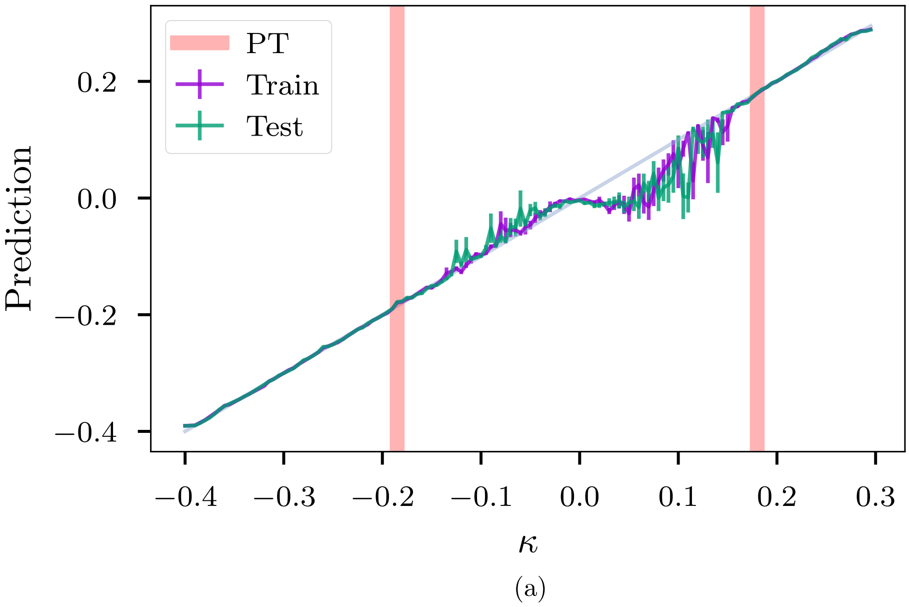
In Section 2.4 a set of standard observables was introduced, consisting of the normal and staggered magnetization as well as the two-point correlation function.[3] It seems reasonable to assume that much of the relevant information characterizing the phase structure and dynamics of the theory is encoded in these quantities. To check this, an ordered dataset of measurements of these quantities is created at various, evenly spaced values of $\kappa$; see Appendix A.1 for details on the dataset. It is used to perform a regression analysis with a MLP; see Appendix A.4 for details on the specific architecture. The method is compared against a random forest regressor as a baseline, which is a standard method based on the optimization of decision trees [239]; see Appendix A.3 for details. The results for both approaches, shown in Figures 5.9 and 5.10, are discussed below.

A qualitatively similar accuracy is observed on the training and test data in the broken ferromagnetic (FM) and antiferromagnetic (AFM) phases. This is expected, since we know from Figure 2.4 that always one of the two types of magnetizations is strictly monotonic in the respective phase and can therefore determine $\kappa$ uniquely. However, both approaches yield only mediocre performance in the symmetric, paramagnetic (PM) phase. Here, both magnetizations tend to zero and therefore do not contain much relevant information. Moreover, the two-point correlator exhibits approximately symmetric properties around $\kappa = 0$. Therefore, it also does not provide a unique mapping. This issue is resembled in the prediction for both methods. The random forest yields a symmetric discrepancy around $\kappa = 0$. In comparison, the MLP shows an improved performance for $\kappa < 0$, albeit at the price of a larger variance for $\kappa > 0$. At this point, one can already see that the chosen set of observables suffices to characterize the theory only in the broken phases, whereas in the symmetric phase, additional information appears to be necessary.

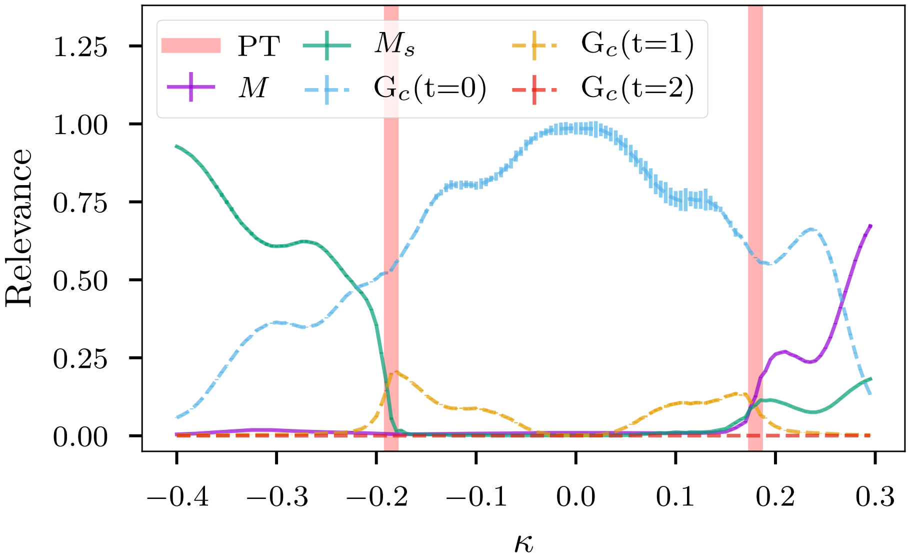
Before we embark on the search for the missing piece, let us first examine the results further to verify that the learned decision rules conform to the physical interpretation given above. We begin with the relevances as determined by LRP, shown in Figure 5.9b, and later compare to the random forest benchmark below. As expected, $M$ and $M_s$ are relevant in the FM and AFM phases, respectively. There, considerable relevance is also assigned to the correlator. However, the contribution appears to diminish when going deeper into the broken phases. Its comparably large relevance in the PM phase shows that it contains most of the information used for the noisy prediction. As described above, the mediocre performance here indicates that although the network seems to find some weak signals to characterize the PM phase, the chosen set of observables is not optimal.

The interpretation sketched above is further supported by the results obtained through random forest regression. Analogously to the previously introduced relevance for LRP, one can determine nominal contributions of input features to the prediction and hence a measure of local feature importance, which is shown in

[3] A slightly modified definition of the time-sliced correlator is used in order to remove lattice artifacts from the data, see Appendix A.1.

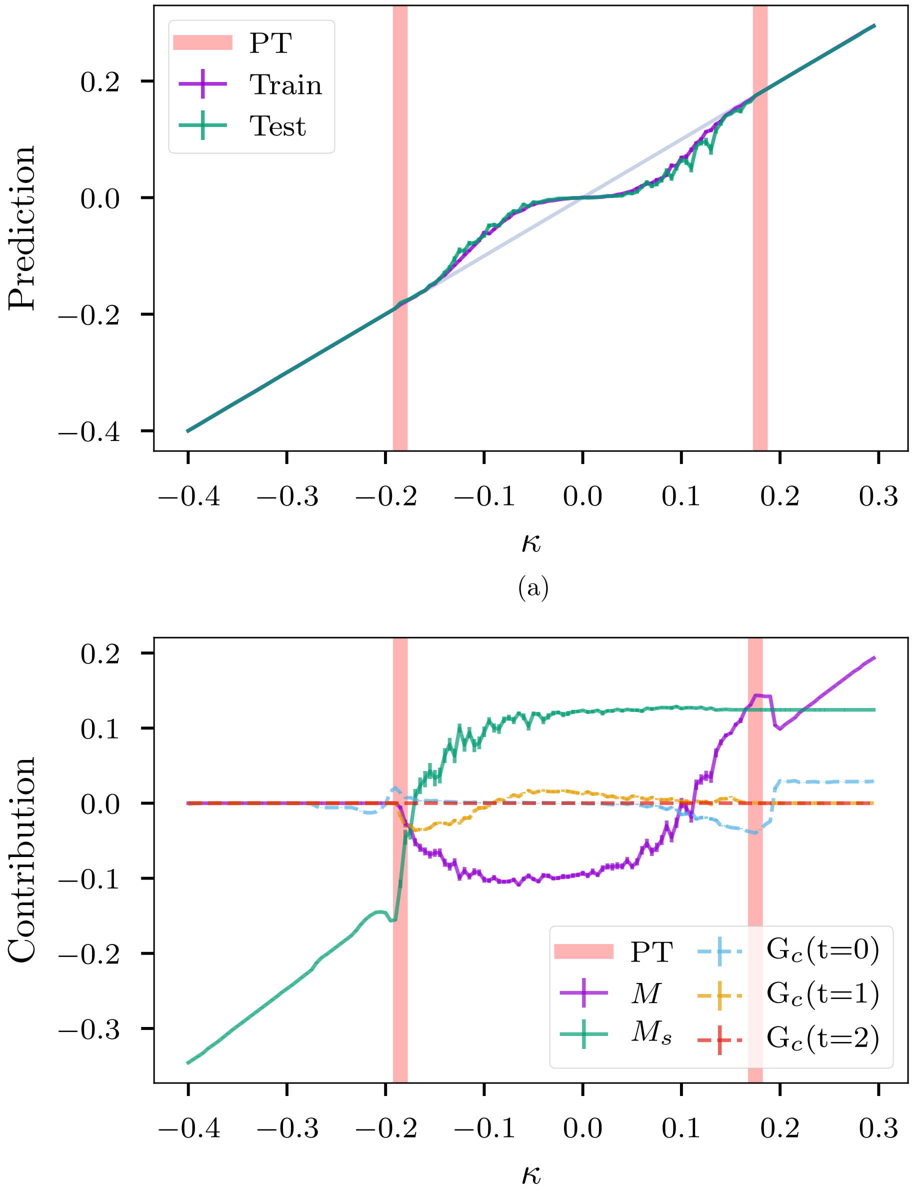


(a)



(b)

Figure 5.9: Results for the MLP: Predictions (a) and normalized LRP relevances of individual features (b). Error bars here and throughout this section are obtained with the statistical jackknife method.



(a)

(b)

Figure 5.10: Benchmark results for the random forest: Predictions (a) and nominal contributions of individual features (b).

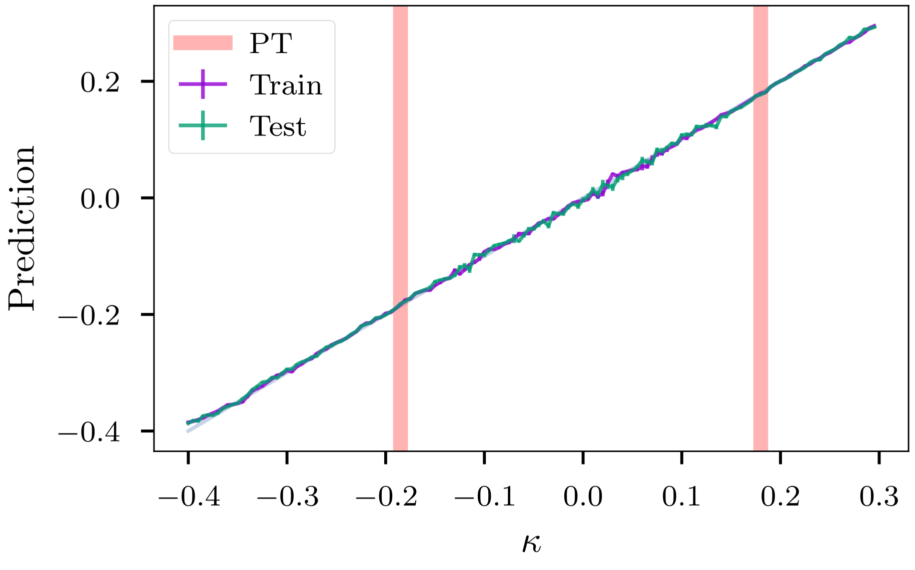
Figure 5.10b; see Appendix A.3 for details. In the broken phases, the respective contributions of $M$ and $M_s$ demonstrate a linear dependence on $\kappa$. Again, this clearly indicates that these quantities characterize the associated phases. For the PM phase, the situation appears more challenging, since no such clear dependence is observed for any of the observables. The non-zero contributions of features in the PM phase imply that they add some valuable information to the decision here. However, this has to be weighted against the observation that the accuracy in this region is poor. This further confirms the previous conclusion that relevant information to characterize this phase is largely lost in the preprocessing step, assuming that it was initially present in the raw field configurations. It is worthwhile stressing that this analysis represents an independent confirmation of the results obtained above. Both algorithms (MLP vs. random forest) rely on fundamentally different principles. A model-intrinsic interpretability measure is used for the random forest, whereas for the MLP one relies on LRP, i.e. a post-hoc attribution method.

### Extracting observables from convolutional filters

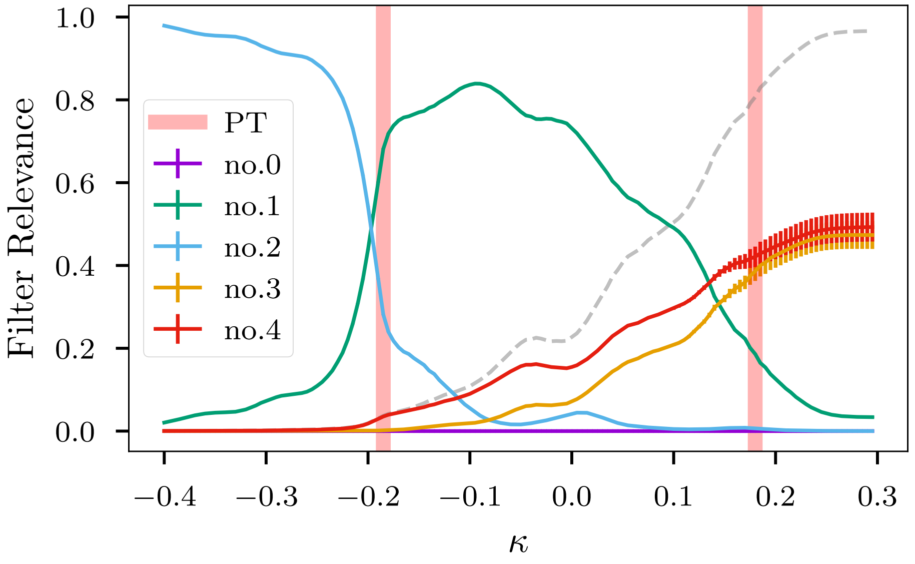
In the previous section, a dataset of known observables was used to reconstruct $\kappa$. Calculating such quantities corresponds to heavy preprocessing of the high-dimensional field configurations. The resulting low-dimensional features are far less noisy, implying distillation of relevant information. This is a common procedure in the field of data science, and may become unavoidable for large lattices and/or theories with more degrees of freedom. For instance, in state-of-the-art lattice QCD calculations, the number of floats in a single field configuration can easily reach $\mathcal{O}(10^{10})$. Nevertheless, using preprocessed data in the form of standard observables introduces strong biases towards known structures. If our perception of the problem or more generally our physical intuition is flawed, machine learning cannot help us—the relevant information may very well be lost in the preprocessing step. In the present case specifically, it appears that important features in the PM phase are neglected by this procedure, assuming that structures characterizing this phase do in fact exist. Therefore, it is instructive to search for signals of such structures by training neural networks directly on field configurations.

As a starting point for this search, a PCA is performed on the field configuration dataset. As previously mentioned, this has been done before with promising results [217–219], albeit not in exactly the same physical setting. PCA immediately identifies the normal and staggered magnetizations as dominant features, essentially reproducing the work of [218]. All higher order principal components show a vanishing explained variance ratio, implying that no other relevant, purely linear features are present in the data. This observation indicates that, if a quantity exists which parametrizes the symmetric PM phase, it cannot simply be a linear combination of the field variables.

An improved approach can be implemented based on a CNN. The training procedure is largely equivalent to that for the MLP in the previous section, with the observable dataset replaced by the full field configurations. A CNN is trained using



(a)



(b)

Figure 5.11: Results for the CNN: Predictions (a) and normalized relevances of individual filters (b). The dashed curve corresponds to the cumulative relevance of filter 3 and 4.

five convolutional filters with a shape of $2 \times 2 \times 2$ and a stride of 1. In order to support interpretability, weight sparsity is encouraged by adding the $L^1$ norm to the loss—also known as LASSO regularization—as suggested in [230]; see Appendix A.4 for details. Due to the nature of the convolution operation, learned filters have a direct interpretation in terms of first-order linear approximations of relevant observables. Hence, one expects the CNN to reproduce the PCA results at the very least, and aims for the identification of other quantities which the network can encode in subsequent layers. It is important to understand this difference between the approaches, even though both extract only linear signals in a first approximation.

The model predictions are shown in Figure 5.11a. One can immediately observe a superior performance in the PM phase compared to the previous results. The CNN succeeds to consistently infer $\kappa$ from the field configuration data with high accuracy. This indicates that it indeed manages to construct internal representations suitable not only to discern the different phases, which would be sufficient for classification purposes, but also for an ordering of data points within each phase.

In order to interpret the predictions and extract knowledge about the learned representations, one has to customize LRP for lattice data. In image recognition, as previously mentioned, one mostly aims at highlighting important regions in the input domain, leading to superimposed heatmaps. This is based on the inherent heterogeneity common to image data, where relevant features are usually localized. For lattice field configurations, due to the translational symmetry and the resulting homogeneity, no particularly distinguished, localized region should be apparent in any given sample. However, each convolutional filter encodes an activation map that is in fact sensitive to a specific feature present in a field configuration. In contrast to the usual ansatz, the spatial homogeneity promotes global pooling over the relevances associated with each filter weight. Hence, instead of assigning relevances to input pixels, one is interested in the cumulative filter relevance which indicates their individual importance for a particular prediction. Analogously to the rationale of the previous section, one can use this approach to build importance hierarchies of filters, thereby facilitating their physical interpretation as signals of relevant observables.

Figure 5.11b shows each filter relevance as a function of $\kappa$. We can recognize some similarities to the relevances in Figure 5.9, highlighting the underlying phase structure of the Yukawa theory. It appears that the model can parametrize each phase individually using one or a small subset of filters, while the others show small or insignificant relevances in the respective region. The learned weight maps are shown in Figure 5.12, where we also assign names to the filters depending on the corresponding associated phase. Only one filter is not shown because it exhibits completely vanishing weights and relevance. It seems to have been dropped entirely by the network, indicating that four filters are sufficient to characterize all phases seen in the data. This reduction is an effect of the weight regularization, and also appears when more filters are initially used. Since the number of non-trivial filters is constant in all training runs, this already indicates how many independent quantities the network needs to learn in order to successfully predict $\kappa$.

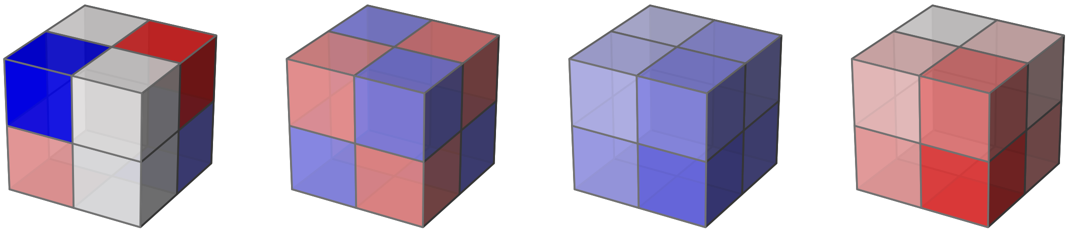


Figure 5.12: Learned weights of convolutional filters. Left to right: PM, AFM, FM, FM. The color map is symmetric around zero. Red (blue) corresponds to positive (negative) weights.

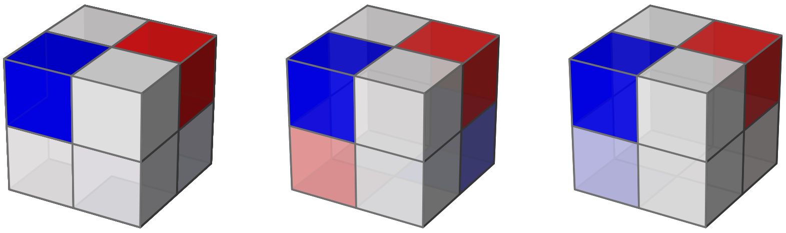


Figure 5.13: Comparison of PM filters for three independent training runs of the CNN.

Let us begin by examining the results that directly correspond to known quantities. One can observe that the two FM filters have entries of roughly uniform magnitude with a globally flipped sign. Accordingly, one can identify them as signals of the negative and positive branches of the magnetization $M$, respectively. This is corroborated by their dominating relevances in the FM phase. The AFM filter exhibits alternating entries of uniform magnitude and therefore corresponds to the staggered magnetization $M_s$, which accordingly dominates the AFM phase. Hence, both order parameters can be explicitly reconstructed from the CNN. The appearance of two filters for the magnetization is easily understood by inspection of the network architecture in Table A.3, the crucial point being the application of a ReLU activation after the convolution operation. Consider the action of a positively-valued filter to negatively magnetized field configurations, or vice versa. The resulting negative activation map is defaulted to zero by the ReLU. Hence, in order to take both branches of $M$ into account, two equivalent filters with opposing signs are required. The comparably large error bars in this region stem from the presence of positively and negatively magnetized samples in the dataset, which lead to a higher per-filter variance. Therefore, also the cumulative relevance of both filters is shown.

We now discuss the main object of interest, namely the PM filter. It supplies the dominant signal for the characterization of this phase. A linear application of this filter to the configurations, as done for the FM and AFM filters, does not produce a monotonic quantity, which would be required for a unique ordering. This further supports the aforementioned evidence gathered by PCA for the absence of an additional, purely linear observable. Hence, the simple reconstruction scheme outlined in the previous paragraphs cannot be applied in this case. Instead, a heuristic reconstruction of the relevant quantity is attempted. To this end, we note that the ReLU activation applied to the convolutional layer's output can effectively correspond to the absolute value function, albeit with less statistics, if the entries of the activation map are distributed accordingly. Inspired by this observation, we define the following observable,

$$\begin{aligned} \mathcal{O}_{\mathrm{PM}} = \frac{1}{|\Lambda|} \sum_{n\in\Lambda} \Big| & \Big[\phi(n) + \phi(n+\hat{\mu}_1)\Big] \\ & - \Big[\phi(n+\hat{\mu}_2+\hat{\mu}_3) + \phi(n+\hat{\mu}_1+\hat{\mu}_2+\hat{\mu}_3)\Big] \Big| \, . \end{aligned} \tag{5.1}$$

As with $M$ and $M_s$, we obtain the corresponding staggered form $\mathcal{O}^s_{\mathrm{PM}}$ by applying the transformation given in Equation (2.11). The resulting pair of quantities is visualized by the following idealized filters.

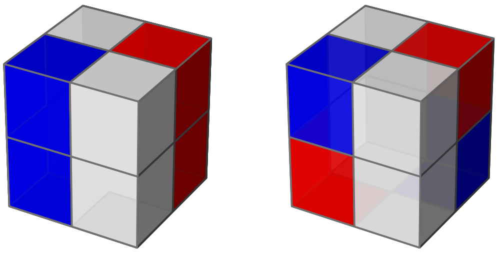


Figure 5.14: Convolutional filters corresponding to the observable $\mathcal{O}_{\mathrm{PM}}$ defined in Equation (5.1) and the staggered counterpart $\mathcal{O}^s_{\mathrm{PM}}$.

The observable $\mathcal{O}_{\mathrm{PM}}$ defined in Equation (5.1) is the sum over all lattice sites of the lattice derivative in the diagonal $\hat{\mu}_2 + \hat{\mu}_3$ direction of blocks in the $\hat{\mu}_1$ direction. This already explains the necessity of taking the absolute value, as otherwise $\mathcal{O}_{\mathrm{PM}}$ would be the sum over all sites of a total derivative, which vanishes identically. It should also be noted that $\mathcal{O}_{\mathrm{PM}}$ can be made isotropic by summing over all directions.

We now discuss the properties of the theory that are measured by $\mathcal{O}_{\mathrm{PM}}$: In the continuum limit, $\mathcal{O}_{\mathrm{PM}}$ naively tends towards the volume integral over $|\nabla\phi|$. Due to the modulus of the derivative, $\langle\mathcal{O}_{\mathrm{PM}}\rangle$ carries the same information as the expectation value of the kinetic term. The blocking in the $\hat{\mu}_1$-direction leads to a sensitivity of $\mathcal{O}_{\mathrm{PM}}$ to sign flips of nearest-neighbors. While no continuum observable is sensitive to these sign flips, the continuum limit of $\langle\mathcal{O}_{\mathrm{PM}}\rangle$ maintains this information. Accordingly, $\langle\mathcal{O}_{\mathrm{PM}}\rangle$ exhibits a distinct behavior in the presence of localized, (anti-)magnetized regions, even if the expectation values vanish globally. Possible local field alignments resulting in different values of $\mathcal{O}_{\mathrm{PM}}$, but not of the standard derivative, are visualized in Figure 5.15.

The construction and discussed sensitivities of $\langle\mathcal{O}_{\mathrm{PM}}\rangle$ demonstrate again the usefulness of LRP: one can identify the learned representation as a feature of the dataset

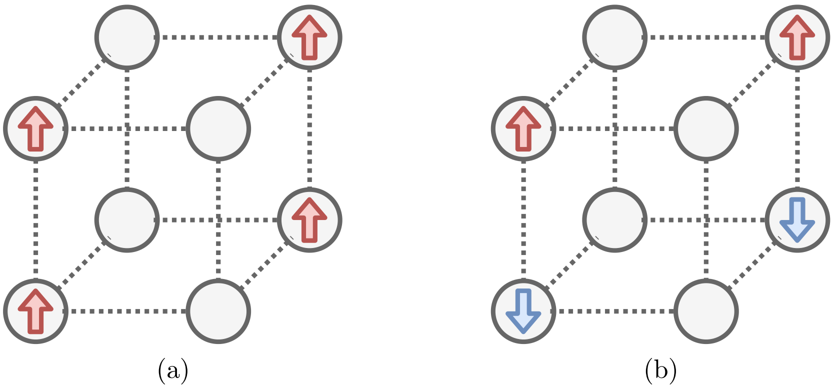


Figure 5.15: Visualization of local structures in field configurations relevant for the PM filter. Sign is encoded by arrow orientation/color. Diagonal neighbors tend to share the same sign everywhere in the phase diagram. On the contrary, nearest neighbors show a preference towards either the same (a) or the opposite (b) orientation. $\mathcal{O}_{\mathrm{PM}}$ is particularly sensitive towards the local presence or absence of such sign flips in the PM phase, without the need for globally non-zero expectation values of the normal and staggered magnetizations.

arising from the lattice discretization. $\langle\mathcal{O}_{\mathrm{PM}}\rangle$ and $\langle\mathcal{O}^s_{\mathrm{PM}}\rangle$ as functions of $\kappa$ are shown in Figure 5.16 together with the other reconstructed observables and their respective analytical counterparts. A monotonic, roughly linear dependence is observed in the PM phase, indicating that the quantity indeed provides a unique mapping which aids the $\kappa$ inference. In fact, if $\mathcal{O}_{\mathrm{PM}}$ is included in the set of predefined observables for the inference approach detailed in the previous section, the prediction accuracy of the MLP accordingly becomes comparable to the CNN in this phase.

In conclusion, we find that the CNN characterizes the PM phase by additionally measuring kinetic contributions in the described manner, rather than only expectation values of the condensate like in the broken phases. Still, $M$ and $M_s$ are being utilized as well, judging from the comparably large relevances of the FM filters in this region. Due to the opacity of the fully-connected layers following the convolution, some ambiguity remains regarding the precise decision rules that the network implements based on these quantities. This residual lack of clarity can likely be resolved by manually enforcing locality in the internal operations, e.g. by introducing artificial bottlenecks into the network. Of course, the form of $\mathcal{O}_{\mathrm{PM}}$ is also not exactly equivalent to the operations of the CNN, even though they share many important features. In particular, there is a mismatch between the averaging procedure and the max pooling layer. Effects associated with the choice of different activation functions and pooling layers, which may be tailored more specifically towards certain types of observables, should be investigated in the future. However, the present analysis shows that the overlap with the learned internal representation is significant.

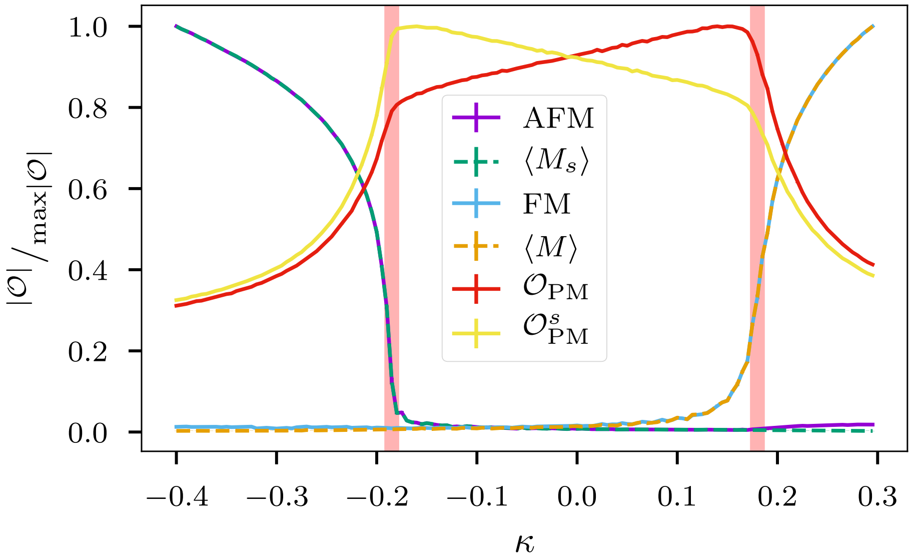


Figure 5.16: Normalized observables reconstructed from the learned filters. The quantities associated with the FM and AFM phases are compared to $M$ and $M_s$. $\mathcal{O}_{\mathrm{PM}}$ and $\mathcal{O}^s_{\mathrm{PM}}$ are related by Equation (2.11) and exhibit an approximate mirror symmetry around $\kappa = 0$.

### 5.2.4 Summary and outlook

The application of interpretability methods to deep neural network classifiers was investigated as a general-purpose framework for the identification of relevant features from lattice data. The approach facilitates an interpretation of a network's predictions, permitting a quantitative understanding of the internal representations that the network learns in order to solve a pretext task—in this case, inference of action parameters. This culminated in the extraction of a novel observable, leading to insights about the phase structure of the studied Yukawa model. With these results, the value of interpretability methods in deep learning analyses of lattice data has been conclusively demonstrated.

In the present work, the emphasis was put on the methodological aspects of the analysis in order to form a comprehensive basis for future efforts. Many interesting aspects, such as an investigation of the fermionic sector, were barely discussed. Instead, we have focused on the inference of the hopping parameter. Including other action parameters into the labels, such as the Yukawa coupling or a chemical potential, is a promising endeavor for the future, as it will likely lead to an improvement in comparison to the current results. This is necessary in order to pave the way towards an application to more interesting scenarios, such as QCD at finite density or competing order regimes in the Hubbard model. Moreover, the introduced machine learning pipeline has the potential to provide insight also in various other areas of computational physics.

# 6 Flow-based sampling for dynamical fermions

In the spirit of our first look at generative neural samplers for lattice field theory in Section 5.1, the present chapter concerns the development of such models for theories with dynamical fermions. However, the normalizing flow architectures considered here exhibit many important differences to the previously used GAN. In the latter case, the unavailability of model probabilities for the generated configurations forced us to implement the sampling as an overrelaxation step in combination with standard HMC. In contrast, as described in Section 4.2, flows provide tractable likelihoods. This enables direct importance sampling using only the model.

The primary contributions of the work presented in this chapter are:

1. Identifying four distinct sampling schemes based on generative models that capture the different tractable decompositions/marginalizations of the target distribution over boson and pseudofermion fields listed in Table 2.1;

2. Constructing and optimizing efficient, expressive flow models that respect the symmetries of the pseudofermion action, in particular the translational symmetry with antiperiodic temporal boundary conditions discussed in Section 2.3;

3. Implementing and numerically benchmarking these sampling approaches in the context of two-dimensional Yukawa theory with one pair of mass-degenerate fermions described in Section 2.4;

4. Solving topological freezing at criticality in the Schwinger model described in Section 2.6, which demonstrates clearly the potential impact of flow-based sampling in key applications such as lattice QCD at scale.

The study of flow-based sampling algorithms for fermionic theories is briefly motivated in Section 6.1. In Section 6.2, the four exact generative sampling schemes for such theories are outlined. Suitable flow architectures as the generative models for use in these sampling schemes are then developed in Section 6.3. Details and numerical results of the application of the proposed framework to Yukawa theory are discussed in Section 6.5. In Section 6.6, the application to the Schwinger model at criticality is presented. In Section 6.7, some comments are made on the applicability of these developments to update-based approaches. Finally, Section 6.8 provides a summary and outlook. The contents of this chapter have been published in [4, 6] together with Michael S. Albergo, Denis Boyda, Kyle Cranmer, Dan C. Hackett, Gurtej Kanwar, Sébastien Racanière, Danilo J. Rezende, Fernando Romero-López, and Phiala E. Shanahan.

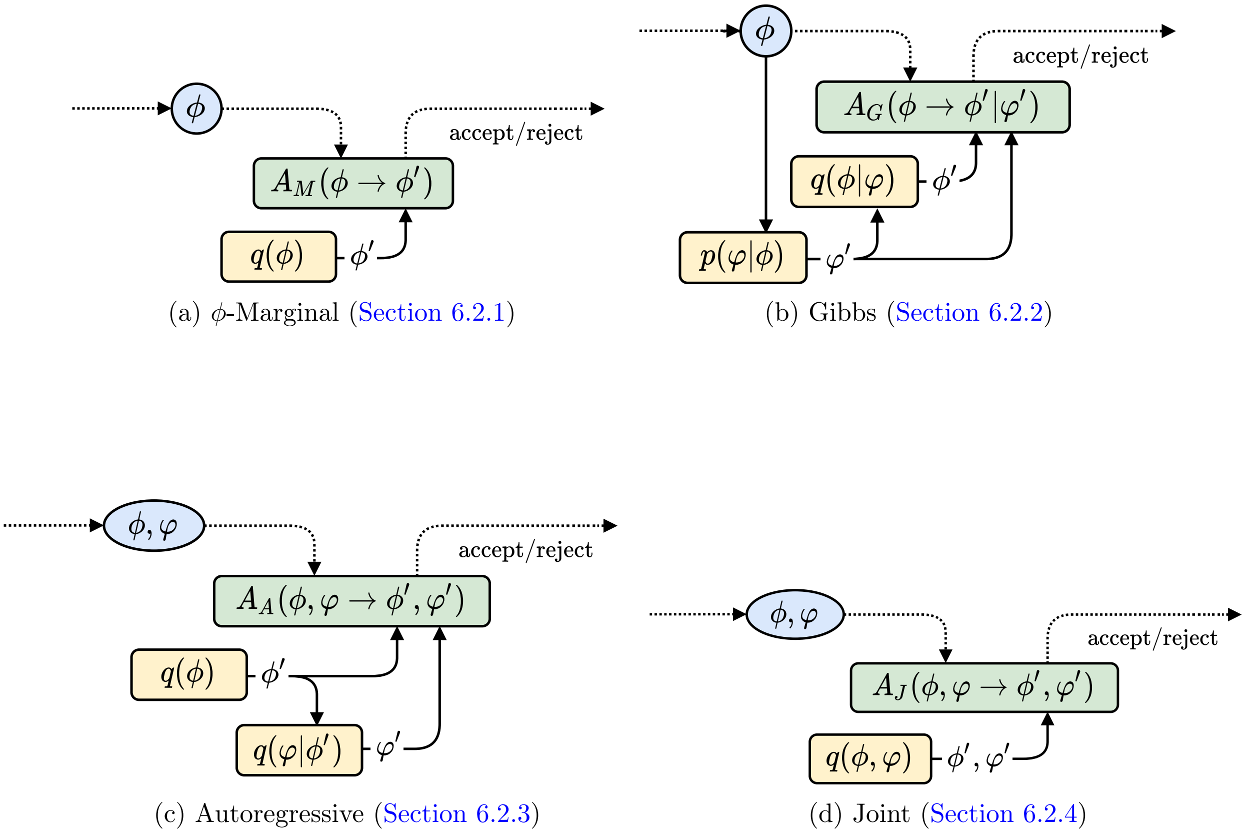
## 6.1 Introduction

In light of the challenges described in Section 3.1, the development of efficient sampling algorithms for lattice field theory based on machine learning has received increasing attention over the last few years. Next to other approaches—such as autoregressive networks, or adversarial learning techniques as discussed in Section 5.1—progress has recently been made in substituting the proposal mechanism in MCMC with a variational ansatz based on normalizing flows, which can be optimized to approximately sample from the target Boltzmann distribution; see Section 4.2. In contrast to many other methods generating uncorrelated samples, in this approach the associated model probabilities can be tractably computed. Therefore, asymptotic exactness can be guaranteed by implementing a Markov chain with a Metropolis accept/reject step or through reweighting.

Though flow-based models have been extended to exactly incorporate gauge symmetry, existing applications have focused on purely bosonic theories. For theories involving fermions, analytically evaluating the associated integrals over Grassmann-valued field variables results in effectively bosonic theories, described by an effective action. The dynamics of the fermion fields are incorporated via fermion determinant terms; see Section 2.3. Flow-based methods can in principle be applied to directly learn this effective action over bosonic fields, which we demonstrate both for Yukawa theory as well as for the Schwinger model. Such an approach is still justified at the comparably small lattice volumes considered here. However, the cost of computing such determinants scales unfavorably with the number of fermionic degrees of freedom, and their exact evaluation is typically intractable at the scale of state-of-the-art calculations. Therefore, we also construct approaches based on the aforementioned pseudofermion method to avoid an explicit computation of these determinants while guaranteeing asymptotic exactness of the sampling schemes.

## 6.2 Exact generative sampling schemes for fermionic theories

Generating importance-weighted field configurations for a lattice field theory involving fermions can proceed via the marginal distribution $p(\phi)$ defined in Equation (2.21), the joint distribution $p(\phi, \varphi)$ defined in Equation (2.27), or through other choices of marginalized distributions defined in Table 2.1. In this work, we develop exact sampling schemes based on generative models that directly approximate these distributions. In defining these sampling schemes, we assume that the model probability density may be computed for each generative model (this property holds for the flow-based models defined below). This section details four asymptotically exact schemes for constructing Markov chains to draw samples of $\phi$, as illustrated in Figure 6.1. It is instructive to compare these illustrations with Figures 2.1 and 2.3.



(a) $\phi$-Marginal (Section 6.2.1)

(b) Gibbs (Section 6.2.2)

(c) Autoregressive (Section 6.2.3)

(d) Joint (Section 6.2.4)

Figure 6.1: Diagrams illustrating the four types of sampling schemes described in Section 6.2. The yellow boxes depict the exactly sampleable densities either produced from generative models or by Equation (6.2). The green boxes correspond to the Metropolis accept/reject steps using the acceptance probabilities defined in the text.

### 6.2.1 Modeling and sampling of $p(\phi)$

Since we must ultimately sample only the field $\phi$, one could directly model the $\phi$-marginal distribution (Row 2 of Table 2.1) by constructing a generative sampler providing a distribution $q(\phi)$ approximating $p(\phi)$. Samples drawn from the model distribution $q(\phi)$ can be used in asymptotically exact sampling schemes by either constructing an independence Metropolis Markov chain or applying reweighting/resampling based on reweighting factors $p(\phi)/q(\phi)$. A direct application of either approach requires computing $p(\phi)$ involving the aforementioned determinant factors. The acceptance probability for such a Metropolis Markov chain is

$$A_M(\phi \to \phi') = \min\left(1, \frac{e^{-S_B(\phi')} \det \mathcal{M}(\phi')}{e^{-S_B(\phi)} \det \mathcal{M}(\phi)} \frac{q(\phi)}{q(\phi')}\right) . \tag{6.1}$$

This sampling scheme is illustrated in Figure 6.1a.

Instead of evaluating the ratio $\det \mathcal{M}(\phi')/\det \mathcal{M}(\phi)$ directly, which becomes prohibitive at scale, it is possible to apply the pseudo-marginal method [240] using stochastic approximations of both the numerator and denominator of Equation (6.1) in a way that retains asymptotic exactness. In this stochastic generalization of the Metropolis algorithm, one computes an estimate of $p(\phi)$ using an unbiased stochastic estimator when $\phi$ is initially proposed. This estimate of $p(\phi)$ is then used in all subsequent accept/reject tests for the next element in the Markov chain. Applied to a theory with fermions, this amounts to computing a stochastic estimate of the fermion determinant for each proposed configuration.[4]

For example, we can use an unbiased estimator based on pseudofermions. An (unnormalized) estimate for $p(\phi)$ can be obtained by generating a pseudofermion $\varphi$ from the conditional $p(\varphi|\phi)$ and measuring the quantity $e^{-\varphi^\dagger(\mathcal{M}^{-1}(\phi)-1)\varphi}e^{-S_B(\phi)}$. The $\varphi$-conditional can be directly sampled according to

$$\varphi = \mathcal{A}(\phi)\chi, \quad \text{where} \quad \chi \sim \frac{1}{Z_\mathcal{N}} e^{-\chi^\dagger \chi} . \tag{6.2}$$

The matrix $\mathcal{A}$ is defined by the identity $\mathcal{M}(\phi) \equiv \mathcal{A}(\phi)\mathcal{A}^\dagger(\phi)$ and reduces to the Dirac matrix $D(\phi)$ in a two-flavor example. This estimate can be extremely noisy in practice and may give poor statistical performance. However, it can be improved upon by using multiple pseudofermion draws; see e.g. [30, 32, 242, 243].

In principle, any unbiased stochastic estimator of the fermion determinant can be applied (whether based on pseudofermions or entirely distinct). The limit of taking arbitrarily precise estimators recovers the exact acceptance probability in Equation (6.1). Acceptance rates obtained by using the exact form can thus be interpreted as an upper bound on sampling performance.

[4] Note that the pseudo-marginal algorithm is not equivalent to the sampling scheme based on stochastic estimates of ratios [241], for which asymptotic exactness has not been demonstrated.

### 6.2.2 Gibbs sampling using $p(\phi|\varphi), p(\varphi|\phi)$

An alternative to modeling $p(\phi)$ directly is to construct samplers for both conditional distributions $p(\phi|\varphi)$ and $p(\varphi|\phi)$ and build an asymptotically exact Gibbs sampler that alternatingly samples from these distributions to update $\phi$ and $\varphi$. For such a Gibbs sampler to satisfy detailed balance, the update to $\phi$ must satisfy detailed balance for $p(\phi|\varphi)$ and the update to $\varphi$ must satisfy detailed balance for $p(\varphi|\phi)$. The $\varphi$-conditional can be exactly and directly sampled as described in Equation (6.2), automatically fulfilling this requirement. On the other hand, the $\phi$-conditional (Row 5 of Table 2.1) may be approximated by a generative model distribution $q(\phi|\varphi) \approx p(\phi|\varphi)$. This model can be incorporated into an exact Markov chain transition for the $\phi$-conditional distribution as follows. Start with a state $\phi$, sample $\varphi'$ from $p(\varphi|\phi)$ using Equation (6.2), conditionally propose $\phi'$ from $q(\phi|\varphi')$, and then apply a Metropolis-Hastings accept/reject step with the acceptance probability given by

$$\begin{aligned} A_G(\phi \to \phi'|\varphi') &= \min\left(1, \frac{p(\phi'|\varphi')}{p(\phi|\varphi')}\frac{q(\phi|\varphi')}{q(\phi'|\varphi')}\right) \\ &= \min\left(1, \frac{p(\phi',\varphi')}{p(\phi,\varphi')}\frac{q(\phi|\varphi')}{q(\phi'|\varphi')}\right) \ . \end{aligned} \tag{6.3}$$

This step satisfies detailed balance for the $\phi$-conditional distribution $p(\phi|\varphi)$ as required and guarantees asymptotic exactness. Note that in contrast to the $\phi$-marginal sampler described in Section 6.2.1, computing the acceptance probability at scale for the sampling scheme described here and in Sections 6.2.3 and 6.2.4 does not rely on unbiased stochastic determinant estimators.

In this approach, the field $\varphi'$ is independently re-sampled conditioned on $\phi$ at each step of the Markov chain, and therefore does not need to be stored. This Gibbs sampler can thus be interpreted as an exact Markov chain over $\phi$ alone, with the sampling of $\varphi'$ contained inside each Markov chain step as depicted in Figure 6.1b. The approach closely mirrors the typical sampling strategy employed in HMC, in which pseudofermions $\varphi'$ are sampled according to the exact conditional distribution $p(\varphi|\phi)$ (see also Figure 2.3) and Hamiltonian evolution is used to construct an update step that satisfies detailed balance for the conditional distribution $p(\phi|\varphi')$. The generative model proposal and Metropolis-Hastings step for $p(\phi|\varphi')$ can thus be considered an optimizable replacement of the molecular dynamics trajectory utilized in HMC, with the difference that the mechanism of generating a proposal configuration $\phi'$ does not directly depend on $\phi$ (as is the case for a symplectic integrator), but only indirectly through $\varphi'$. However, this also means that in contrast to all other schemes described here, the Gibbs sampler is not an independence sampler. Drawing configurations from the model and constructing the Markov chain cannot be done asynchronously, since the generation of a proposal explicitly depends on the previous element of the chain.

### 6.2.3 Autoregressive modeling and sampling of $p(\phi, \varphi)$

The joint distribution $p(\phi, \varphi)$ can be autoregressively decomposed as the product $p(\phi, \varphi) = p(\phi)p(\varphi|\phi)$ in terms of the $\phi$-marginal and $\varphi$-conditional (Rows 2 and 3 of Table 2.1). A generative model for the joint distribution could therefore be produced by approximating both components independently, i.e., $q(\phi, \varphi) = q(\phi)q(\varphi|\phi)$. This autoregressive decomposition allows the joint distribution to be reproduced in terms of two potentially simpler distributions. Note that although the exact sampling procedure described in Equation (6.2) can be applied to draw samples from $p(\varphi|\phi)$, computing the normalizing constant of this $\varphi$-conditional distribution is not tractable. This is not an obstacle when one is only interested in conditionally sampling $\varphi$, as is the case for HMC or the approaches of Sections 6.2.1 and 6.2.2, but motivates modeling the distribution in the case where an approximation with a tractable density is required.

Exactness can be straightforwardly enforced in this approach by employing Markov chain steps in which joint samples $(\phi', \varphi')$ are proposed independently from $q(\phi, \varphi)$, and a Metropolis-Hastings accept/reject step is applied for the proposed transition $(\phi, \varphi) \to (\phi', \varphi')$ according to the acceptance probability

$$A_A(\phi, \varphi \to \phi', \varphi') = \min\left(1, \frac{p(\phi', \varphi')}{p(\phi, \varphi)} \frac{q(\phi)q(\varphi|\phi)}{q(\phi')q(\varphi'|\phi')}\right) . \tag{6.4}$$

This sampling scheme is illustrated in Figure 6.1c. Furthermore, unique reweighting factors can be tractably computed for each configuration $\phi$ as $p(\phi, \varphi)/q(\phi)q(\varphi|\phi)$, thus reweighting approaches may also be used as alternatives to MCMC in order to guarantee exactness here.

### 6.2.4 Fully joint modeling and sampling of $p(\phi, \varphi)$

Rather than modeling the factors $p(\phi)$ and $p(\varphi|\phi)$, one could instead apply generative models to jointly sample the fields $\phi$ and $\varphi$ according to a distribution $q(\phi, \varphi)$ that directly approximates the joint distribution (Row 1 of Table 2.1). This results in joint samples and density estimates analogous to the autoregressive case above, but is a qualitatively distinct approach to modeling this distribution. Exactness can be enforced using a similar Metropolis-Hastings Markov chain transition with acceptance probability

$$A_J(\phi, \varphi \to \phi', \varphi') = \min\left(1, \frac{p(\phi', \varphi')}{p(\phi, \varphi)} \frac{q(\phi, \varphi)}{q(\phi', \varphi')}\right) , \tag{6.5}$$

or by applying reweighting or direct resampling techniques. This approach is illustrated in Figure 6.1d.

## 6.3 Fermionic flows via pseudofermions

The sampling approaches discussed above for theories involving fermions can in principle use any generative models that enable both efficient sampling and density estimation for the relevant model distributions. In this work, we focus on the normalizing flow architecture introduced in Section 4.2. The present section describes how flow models for each of the distributions required for sampling may be constructed. First, a common training procedure for all such models is described in Section 6.3.1 based on the idea that each distribution aims to approximate some marginalization of the same joint distribution $p(\phi, \varphi)$. This common training procedure motivates some of the architectural decisions for the construction of the models described in Sections 6.3.2 and 6.4. In the following, we label the model densities according to their corresponding target densities in Table 2.1 as $q(\phi, \varphi)$, $q(\phi)$, $q(\varphi|\phi)$, and $q(\phi|\varphi)$. In each sampling approach, using model distributions that better approximate the associated target will generally result in higher acceptance rates with potentially lower autocorrelations.

### 6.3.1 Optimization strategy

We first detail a procedure to optimize the model density $q(\phi, \varphi)$ to directly approximate $p(\phi, \varphi)$. Following Equation (4.12), the Kullback-Leibler divergence between these distributions is defined as

$$\begin{aligned} &D_{\mathrm{KL}}(q(\phi, \varphi)||p(\phi, \varphi)) \\ &\quad = \mathbb{E}_{\phi,\varphi\sim q}\left[\log(q(\phi, \varphi)/p(\phi, \varphi))\right] \\ &\quad = \mathbb{E}_{\phi,\varphi\sim q}\left[\log q(\phi, \varphi) + S_B(\phi) + S_{PF}(\phi, \varphi) + \log Z\right] \ . \end{aligned} \tag{6.6}$$

In practice, a loss function based on this divergence is computed stochastically as

$$L = \frac{1}{N}\sum_{k=1}^{N} \log q(\phi_k, \varphi_k) + S_B(\phi_k) + S_{PF}(\phi_k, \varphi_k) \ , \tag{6.7}$$

in terms of a mini-batch of $N$ samples $\{(\phi_k, \varphi_k)\}_{k=1}^{N}$ drawn from $q(\phi, \varphi)$. As already explained in Section 4.2, the unknown normalizing constant $\log Z$ has been removed in the definition of Equation (6.7), since it is just an overall constant shift and does not affect the relevant structure of the loss function.

If the model probability density $q(\phi, \varphi)$ can be directly computed, we can evaluate the gradient of Equation (6.7) with respect to the model parameters defining this probability density. Gradient-based optimization methods can then be applied to minimize $L$. This training procedure is immediately applicable to the models required for the joint sampling approaches derived in Sections 6.2.3 and 6.2.4. In the former, the distribution $q(\phi, \varphi)$ is defined by $q(\phi)q(\varphi|\phi)$, and this pair of model distributions is simultaneously optimized by minimizing the loss function in Equation (6.7). In the latter, a model for $q(\phi, \varphi)$ is directly constructed and optimized.

The remaining distributions required in Sections 6.2.1 and 6.2.2, namely $q(\phi)$ and $q(\phi|\varphi)$, do not naturally define a joint model probability density. To optimize these distributions using the loss function above, we extend the model architectures by pairing $q(\phi|\varphi)$ and $q(\phi)$ with samplers $q(\varphi)$ and $q(\varphi|\phi)$, respectively. The resulting joint models can be optimized as above, and the auxiliary components can be discarded after training. These auxiliary models are constructed as follows.

We first consider extending the $\phi$-conditional model $q(\phi|\varphi)$ to a joint model, which requires a marginal distribution $q(\varphi)$. None of the sampling approaches presented in Section 6.2 directly require this marginal distribution; however, as we discuss further in Section 6.4, we choose to model $q(\phi|\varphi)$ by a restricted form of a joint sampler which simultaneously models a marginal distribution $q(\varphi)$. In this extended model, both $q(\phi|\varphi)$ and $q(\varphi)$ are described by parameters that are optimized.

A $\phi$-marginal model $q(\phi)$ can be extended to a joint sampler by pairing it with a conditional distribution $q(\varphi|\phi)$. In principle, such an auxiliary model could be constructed solely for the purposes of training. However, in this case we are free to instead use the exact conditional distribution $p(\varphi|\phi)$, which can be exactly and efficiently sampled. The result is a joint distribution defined by first sampling $\phi$ from the $\phi$-marginal model and then sampling the $\varphi$-conditional using Equation (6.2), resulting in the joint density $q(\phi)p(\varphi|\phi)$. Evaluating the joint Kullback-Leibler divergence between this model distribution and the target joint distribution $p(\phi, \varphi)$ requires the evaluation of the normalized density $p(\varphi|\phi)$, which unfortunately includes a normalizing factor of $\det \mathcal{M}(\phi)$. Nevertheless, we only require an unbiased stochastic estimator of the gradients of Equation (6.7) for optimization.

The calculation of loss gradients required for the optimization of the $\phi$-marginal models trained in this work requires the evaluation of gradients

$$\nabla_\phi \log \det \mathcal{M}(\phi) \tag{6.8}$$

taken with respect to the field $\phi$. In general, $\mathcal{M}(\phi)$ is a positive definite matrix either arising from Dirac matrices of a pair of mass degenerate fermions as $\mathcal{M} = DD^\dagger$, or from one-flavor methods; see also Section 2.3. Since the calculation of the exact determinant $\det \mathcal{M}(\phi)$ may be intractable because of the aforementioned unfavorable scaling with the number of lattice degrees of freedom, a stochastic estimator is instead defined in this section to tractably evaluate Equation (6.8).

By assumption, $\mathcal{M}$ is a positive-definite matrix and thus the following stochastic trace estimator is applicable:

$$\begin{aligned} \nabla \log \det \mathcal{M}(\phi) &= \nabla \mathrm{Tr} \log \mathcal{M}(\phi) \\ &= \mathrm{Tr}\left[\mathcal{M}(\phi)^{-1} \nabla \mathcal{M}(\phi)\right] \\ &= \mathbb{E}_{\chi \sim e^{-\chi^\dagger \chi}} [(\mathcal{M}^{-1}(\phi)\chi)^\dagger \nabla \mathcal{M}(\phi) \chi] \ . \end{aligned} \tag{6.9}$$

Here, the noise vector $\chi$ is assumed to be drawn from the unit-variance isotropic normal distribution with an appropriate number of degrees of freedom to match the dimensions of $\mathcal{M}$.

In the case of two degenerate fermionic flavors, an interesting connection can also be made to the gradient of the negative pseudofermion action (where the sign is chosen to match the positive sign of $\log\det\mathcal{M}(\phi)$). This gradient can be evaluated to be

$$\begin{aligned}\nabla(-\varphi^\dagger(DD^\dagger)^{-1}\varphi) &= \varphi^\dagger(DD^\dagger)^{-1}(\nabla DD^\dagger)(DD^\dagger)^{-1}\varphi \\ &= \eta^\dagger(\nabla DD^\dagger)\eta\ ,\end{aligned} \tag{6.10}$$

where $\eta \equiv (DD^\dagger)^{-1}\varphi = (D^\dagger)^{-1}\chi$, in terms of the noise vector $\chi \sim e^{-\chi^\dagger\chi}$ used to generate the pseudofermion field. A short derivation shows that this is equivalent to the stochastic estimator of the two-flavor determinant,

$$\begin{aligned}&\mathrm{Tr}\left[(DD^\dagger)^{-1}(\nabla DD^\dagger)\right] \\ &\quad= \mathrm{Tr}\left[D^{-1}(\nabla DD^\dagger)(D^\dagger)^{-1}\right] \\ &\quad= \mathbb{E}_{\chi\sim e^{-\chi^\dagger\chi}}[((D^\dagger)^{-1}\chi)^\dagger(\nabla DD^\dagger)(D^\dagger)^{-1}\chi] \\ &\quad= \mathbb{E}_{\chi\sim e^{-\chi^\dagger\chi}}[\eta^\dagger(\nabla DD^\dagger)\eta]\ .\end{aligned} \tag{6.11}$$

This relation allows the gradient estimator to be computed using the same tools utilized for the evaluation of HMC forces with respect to the pseudofermion action.

### 6.3.2 Building blocks

Flow-based models are generally constructed by composing several simple, invertible transformation layers, each described by a number of free parameters. This composition produces an expressive overall transformation that is nevertheless invertible and has a tractable Jacobian determinant. The affine coupling layers described in Section 4.2 are one common choice of a simple transformation in which the degrees of freedom of each sample are divided into two subsets and one subset is updated conditioned on the other, 'frozen' subset, as shown in Figure 6.2. A 'masking pattern' describes the division into subsets. Transformations of the updated subset are parameterized by 'context functions' accepting the frozen subset as input, which are typically implemented using neural networks. For example, a simple coupling layer for a real scalar field $\phi(x) \in \mathbb{R}$ could be constructed based on a checkerboard division into even/odd sites, where the field at even sites is (invertibly) transformed by an element-wise rescaling operation plus an additional offset. The scaling factors and offsets are given by the output of an arbitrary context function, which may be parametrized by a neural network acting on the odd sites. The transformation is applied alternatingly between even and odd sites; see [181] for a concrete implementation of such coupling layers. Symmetries may be incorporated in such models using appropriate choices of masking patterns, context functions, and transformations. Other choices of layers are also possible (see Section 6.3.2 below) and are similarly encoded using generic neural networks.

The target densities defined in Table 2.1 are all invariant under translations with appropriate boundary conditions, as discussed in Section 2.3. Previous works have shown that exactly incorporating known symmetries into machine learning models

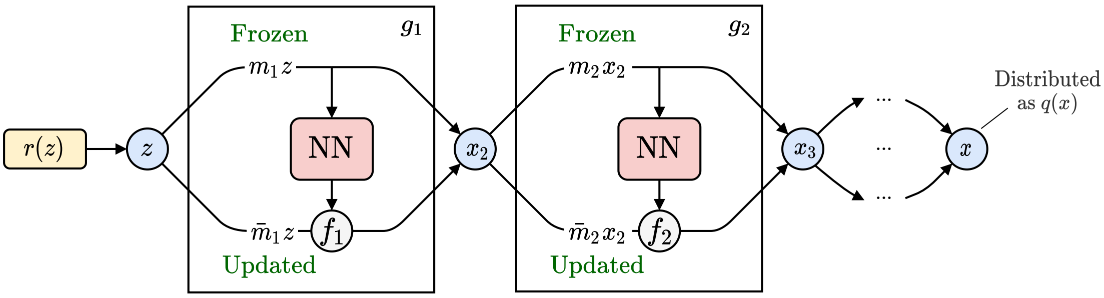


Figure 6.2: Illustration of a coupling-layer-based flow model architecture similar to Figure 4.5. Masks $m_i$ and their complements $\bar{m}_i = 1 - m_i$ split the degrees of freedom into a subset to be updated and a subset that is frozen and used as input to the context function (red box), which provide the parameters for the invertible transformation $f_i$ applied within each coupling layer $g_i$.

can accelerate their training and improve their final quality [244–249]. In the context of normalizing flows, ensuring that the model density is invariant under a symmetry group is achieved by choosing an invariant prior distribution and building transformation layers that are equivariant under the symmetry. Below, we introduce several 'building blocks' which are designed to handle these symmetries and are used in the implementation of the flow-based models constructed in this work.

### Translation-equivariant convolutions via P-fields and AP-fields

The joint distribution $p(\phi, \varphi)$ given in Equation (2.27) is invariant under simultaneous field translations given by

$$\phi(\vec{x}, t) \to \phi'(\vec{x}, t) = \phi(\vec{x} - \delta\vec{x}, t - \delta t) \tag{6.12}$$

and

$$\begin{aligned} \varphi(\vec{x},t) &\to \varphi'(\vec{x}, t) \\ &= \begin{cases} \varphi(\vec{x} - \delta\vec{x}, t - \delta t) & {\scriptstyle (t-\delta t) \bmod 2L_t < L_t} \\ -\varphi(\vec{x} - \delta\vec{x}, t - \delta t) & {\scriptstyle L_t \le (t-\delta t) \bmod 2L_t} \end{cases} \end{aligned} \tag{6.13}$$

for any translations $(\delta\vec{x}, \delta t)$ in the translational symmetry group of the discretized theory. In this work, we label fields transforming as Equation (6.12) as P-fields, and we label fields transforming as Equation (6.13) as AP-fields.[5]

In previous applications of flow models to sampling configurations in lattice field theory, translational symmetry has been implemented for bosonic fields by applying convolutional layers with circular padding (periodic boundary conditions) to generate parameters for transformations implemented in each flow layer [169, 250, 251]; see Figure 6.3 for an illustration. All input, intermediate, and output fields in these

[5] The fields $\phi$ and $\varphi$ in Equations (6.12) and (6.13), and P-fields and AP-fields in general, may have multiple components per site.

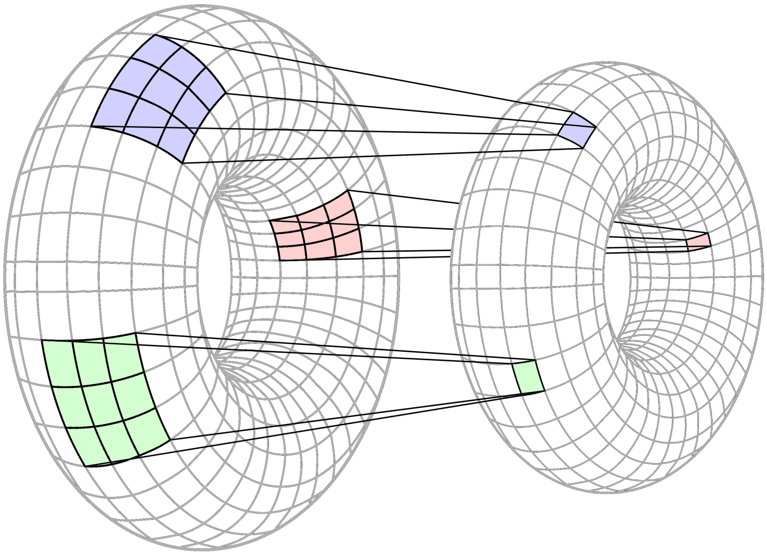


Figure 6.3: Illustration of a convolutional layer with periodic boundary conditions. It is instructive to compare this with Figure 4.2.

applications were P-fields. As a building block for translation-equivariant coupling layers acting on both bosonic and pseudofermionic fields, we extend this approach to define translation-equivariant convolutions that act on a generic set of input P-fields and AP-fields, producing output fields with a desired set of transformation properties, i.e., a specification of whether each channel of the output should be a P-field or AP-field.

To implement such convolutional neural networks, we exploit the fact that P-fields and AP-fields form an algebra under pointwise addition and multiplication and restrict the operations appropriately to satisfy the desired output transformation properties. This can be seen as follows. The set of P-fields is stable under linear combinations and pointwise multiplications. On the other hand, the set of AP-fields is only stable under linear combinations as the product of two AP-fields is a P-field, while the product of a P-field with an AP-field is an AP-field. In other words, the set of P-fields and AP-fields forms a superalgebra [252] under pointwise addition and multiplication. Pointwise application of a function to a P-field results in a new P-field. For AP-fields, more care is required as not all functions can be applied pointwise. Application of an odd function to an AP-field results in another AP-field, while pointwise application of an even function results in a P-field. Below, we explore in more detail how properties of those fields allow one to build expressive neural networks which are equivariant under translations: if $\mathcal{T} \in \mathbb{Z}^d$ is an arbitrary space-time translation and $f(\phi, \varphi) = \phi', \varphi'$ is one of these neural networks, then we demand $f(\mathcal{T}\cdot\phi, \mathcal{T}\cdot\varphi) = \mathcal{T}\cdot\phi', \mathcal{T}\cdot\varphi'$. This discussion is not specific to two-dimensional fields, but applies for any dimension $d$.

First, convolutions can be built for both types of fields. For P-fields, this is achieved by first padding the field using periodic padding, then applying a normal convolution. For a convolution with kernel shape $2k+1$, all fields must be padded by $k$ sites in each direction. As a concrete example, assume a 1-dimensional lattice of size 5, a P-field with values $[1,2,3,4,5]$ and a convolution kernel $[1,1,1]$. The padded P-field would be $[5,1,2,3,4,5,1]$. Applying the convolution would result in a new P-field with values $[8,6,9,12,10]$. For AP-fields, the only necessary change is to use antiperiodic padding along the time dimension, and periodic along the space dimension. Consider the 2-dimensional example

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \tag{6.14}$$

and a $3 \times 3$ kernel with all weights equal to 1. The AP-field should be padded by 1 site in each direction with signs applied to the temporal padding, giving

$$\begin{pmatrix} -9 & 7 & 8 & 9 & -7 \\ -3 & 1 & 2 & 3 & -1 \\ -6 & 4 & 5 & 6 & -4 \\ -9 & 7 & 8 & 9 & -7 \\ -3 & 1 & 2 & 3 & -1 \end{pmatrix} . \tag{6.15}$$

Applying the convolution gives the transformed AP-field

$$\begin{pmatrix} 9 & 45 & 21 \\ 9 & 45 & 21 \\ 9 & 45 & 21 \end{pmatrix} . \tag{6.16}$$

The above construction of P and AP-convolutions was illustrated with only one channel, but the extension to multiple channels is straightforward.

Any non-linearity can be applied between convolutions for a P-field without spoiling translational equivariance; throughout this work, the LeakyReLU [253] activation function is used. For convolutions applied to an AP-field, non-linearities used as activation functions must be restricted to odd functions, for which we choose

$$\operatorname{sign}(\varphi)\log(1+|\varphi|) \ . \tag{6.17}$$

With a P-convolution, a bias can be applied along with a convolution at each step, since a bias is constant across all sites and thus transforms as a P-field. However, a traditional bias cannot be applied to the convolution of an AP-field without spoiling translational equivariance. For an AP-field $\varphi$, a bias-like operation $\varphi \to \varphi + b\operatorname{sign}(\varphi)$ in terms of a constant $b$ can be applied instead. To avoid potential issues with the non-differentiability of the sign function, we used a differentiable approximation given by applying $\varphi \to \varphi + b\tanh(\varphi/4)$.

All of the above constructions (P- and AP-convolutions, non-linearities and biases) are equivariant with respect to translations on the lattice. By stacking them, we create expressive translation-equivariant neural networks. These networks can also jointly transform pairs of P and AP-fields. For example, the following transformation works well:

```
Input: P-field P and AP-field A
P’ = conv(P)
A’ = conv(A)
P" = concatenate(P’, |A’|)
A" = concatenate(A’, P’A’)
Output: P-field P" and AP-field A"
```

The group of translational symmetries of a staggered action described in Section 2.3 only includes translations by even numbers of lattice sites. Implementing symmetries in a network that are not symmetries of the target function restricts the expressivity and may make it difficult or impossible to represent an effective approximation of the target function by the network. For the models targeting the staggered fermion action in the study described in the main text, encoding translational symmetry by an odd number of sites can be avoided by explicitly breaking equivariance with respect to odd translations. For example, to break the symmetry by odd translations along the first dimension of a field $x$, we can fold its even and odd indices along the channel dimension; this doubles its number of channels while halving the number of points along the first dimension. We then apply a convolution with stride 1 and kernel size 1, which mixes all the channels. Finally, we split the channels in two and fold them back along the first dimension to get a new field with the same lattice size as $x$. This approach mirrors the ‘squeezing’ operation applied in Real NVP flows [166].

### Translation-equivariant convolutions via group averages

As an alternative to defining equivariant convolutional neural networks, one can symmetrize a non-equivariant architecture by explicitly averaging over the whole symmetry group [244]. Convolutional layers with periodic padding in all dimensions are already equivariant under translations of P-fields and under all spatial translations of AP-fields, thus only the subgroup of temporal translations needs to be averaged over to ensure equivariance for AP-fields. The result is a generic method to produce convolutional neural networks with prescribed P-field and AP-field transformation properties of each output channel. Compared with standard convolutions or the restricted equivariant architecture given in Section 6.3.2, this method requires a greater computational effort by a factor proportional to the temporal extent of the lattice, $L_t$. However, it allows the use of unrestricted convolutional architectures, including arbitrary activation functions and learned biases.

Let $\mathcal{T}^a_{\vec{x},t} \in \mathbb{Z}^d$ denote a translation by $(\vec{x}, t)$ where antiperiodic boundary conditions are applied in time and periodic boundary conditions are applied in space, and let $\mathcal{T}^p_{\vec{x},t}$ denote a translation by $(\vec{x}, t)$ with periodic boundary conditions for all directions. The action of $\mathcal{T}^a$ and $\mathcal{T}^p$ is the same along all spatial dimensions, and we define $\mathcal{T}_{\vec{x}} = \mathcal{T}^p_{\vec{x},0} = \mathcal{T}^a_{\vec{x},0}$. For simplicity, we consider a two-dimensional $L \times L$ lattice with coordinates $(\vec{x}, t)$, but the following construction immediately generalizes to higher dimensions and non-symmetric lattices. Both P-fields and AP-fields are maps from

$\mathbb{Z}_L \times \mathbb{Z}_L$ to $\mathbb{R}^c$, where $c$ is a number of channels. Under lattice translations, P-fields are acted upon by $\mathcal{T}^p$, while AP-fields are acted upon by $\mathcal{T}^a$.

Consider a function $f$ that maps the pair $(\phi, \varphi)$ of a P-field and an AP-field to another field $f(\phi, \varphi)$. Assume that the output $f(\phi, \varphi)$ transforms with periodic boundary conditions along the space dimension, that is:

$$f(\mathcal{T}_{\vec{x}}\phi, \mathcal{T}_{\vec{x}}\varphi) = \mathcal{T}_{\vec{x}} f(\phi, \varphi) \ . \tag{6.18}$$

Using averaging, we will now construct two maps $u$ and $v$ with the transformation properties:

$$u(\mathcal{T}^p_{\vec{x},t}\phi, \mathcal{T}^a_{\vec{x},t}\varphi) = \mathcal{T}^p_{\vec{x},t} u(\phi, \varphi) \tag{6.19}$$

$$v(\mathcal{T}^p_{\vec{x},t}\phi, \mathcal{T}^a_{\vec{x},t}\varphi) = \mathcal{T}^a_{\vec{x},t} v(\phi, \varphi) \ . \tag{6.20}$$

We define[6]

$$u(\phi, \varphi) = \frac{1}{2L} \sum_{n=0}^{2L-1} \mathcal{T}^p_{0,-n} f(\mathcal{T}^p_{0,n}\phi, \mathcal{T}^a_{0,n}\varphi) \tag{6.21}$$

and

$$v(\phi, \varphi) = \frac{1}{2L} \sum_{n=0}^{2L-1} \mathcal{T}^a_{0,-n} f(\mathcal{T}^p_{0,n}\phi, \mathcal{T}^a_{0,n}\varphi) \ . \tag{6.22}$$

We now wish to prove that the transformation properties in Equation (6.19) and Equation (6.20) apply to these definitions. The proof is roughly the same for both cases,[7] so we will only write it for Equation (6.19):

$$\begin{aligned} u(\mathcal{T}^p_{\vec{x},t}\phi, \mathcal{T}^a_{\vec{x},t}\varphi) &= \frac{1}{2L} \sum_{n=0}^{2L-1} \mathcal{T}^p_{0,-n} f(\mathcal{T}^p_{0,n}\mathcal{T}^p_{\vec{x},t}\phi, \mathcal{T}^a_{0,n}\mathcal{T}^a_{\vec{x},t}\varphi) \\ &= \frac{1}{2L} \sum_{n=0}^{2L-1} \mathcal{T}^p_{0,-n} f(\mathcal{T}_{\vec{x}}\mathcal{T}^p_{0,n+t}\phi, \mathcal{T}_{\vec{x}}\mathcal{T}^a_{0,n+t}\varphi) \\ &= \frac{1}{2L} \sum_{n=0}^{2L-1} \mathcal{T}^p_{0,-n+t}\mathcal{T}_{\vec{x}} f(\mathcal{T}^p_{0,n}\phi, \mathcal{T}^a_{0,n}\varphi) \\ &= \mathcal{T}^p_{\vec{x},t} u(\phi, \varphi) \ . \end{aligned} \tag{6.23}$$

Equation (6.23) was obtained using the change of variables $n \to n - t$ and the equivariance property given in Equation (6.18).

The functions $u$ and $v$ may be used to define equivariant affine coupling layers for the construction of equivariant flows. To achieve equivariance, the underlying function $f$ needs to be evaluated $2L$ times instead of once, hence the aforementioned

[6] Note that if $f$ is odd, then $u$ below will be forced to be independent of $\varphi$. This can be avoided by either using non-odd non-linearities, or by having non-zero biases.

[7] The proof is actually a particular instance of a more general property: if $\pi_1$ and $\pi_2$ are representations of a finite group $G$ on vector spaces $V$ and $W$, and if $f : V \to W$ is any map between these spaces, then $\frac{1}{|G|}\sum_{g\in G} \pi_2(g)^{-1} f(\pi_1(g))$ is an equivariant map from $V$ to $W$.

increase in computational cost. Since the masked affine couplings employed in this work already restrict the translational equivariance to multiples of two, one may also consistently use only every second term in the sums defining $u$ and $v$ without breaking the symmetry further, implying a factor $L$ increase of the cost instead of $2L$. Still, the additional computational requirements are significant compared to the approach detailed in Section 6.3.2, and for large-scale implementations one may have to partially trade equivariance against efficiency by excluding more terms from the sums.

### Affine coupling layers

Translation-equivariant networks constructed by either of the methods discussed in Section 6.3.2 can immediately be applied in the construction of translation-equivariant affine coupling layers suitable for transforming real-valued scalar fields. To reiterate, an affine coupling layer transforms a field $x$ to $ax+b$ (multiplication and addition are applied pointwise), where $a$ and $b$ are fields produced by context functions acting on the frozen components of the field $x$; see also Section 4.2. Coupling layers, context functions, and masking patterns are illustrated in Figure 6.2. Using translation-equivariant convolutional neural networks to produce $a$ and $b$, either a bosonic field or pseudofermionic field can be updated in a translation-equivariant manner as long as:

- The parameters $a$ and $b$ are both P-fields if $x$ is a bosonic field; or
- The parameter $a$ is a P-field and $b$ is an AP-field if $x$ is a pseudofermionic field.

Such coupling layers can be composed to produce translation-equivariant flows.

### Equivariant linear operators

The conditional distribution $p(\varphi|\phi)$ is exactly Gaussian, suggesting that it may be efficiently modeled by flows based on architectures other than coupling layers. For example, one may define a linear operator $\mathcal{W} = \mathcal{W}(\phi)$ to transform the pseudofermion fields. The model distribution $q(\varphi|\phi)$ may then be defined by computing $\varphi = \mathcal{W}\chi$, where $\chi$ is drawn from the Gaussian distribution $\frac{1}{Z_{\mathcal{N}}} e^{-\chi^\dagger \chi}$, such that

$$\begin{aligned} q(\varphi|\phi) &= \frac{1}{Z_{\mathcal{N}}} e^{-\varphi^\dagger (\mathcal{W}\mathcal{W}^\dagger)^{-1}\varphi} (\det \mathcal{W}\mathcal{W}^\dagger)^{-1} \\ &= \frac{1}{Z_{\mathcal{N}}} e^{-\chi^\dagger \chi} (\det \mathcal{W}\mathcal{W}^\dagger)^{-1} \,. \end{aligned} \tag{6.24}$$

To effectively use this flow model, $\det(\mathcal{W}\mathcal{W}^\dagger)$ must be tractable to compute. In the case of a degenerate pair of fermion flavors, the target distribution is defined by

$$p(\varphi|\phi) = \frac{1}{Z_{\mathcal{N}} \det DD^\dagger} e^{-\varphi^\dagger [D(\phi) D^\dagger(\phi)]^{-1} \varphi} \,. \tag{6.25}$$

While it is clearly sufficient for $\mathcal{W}$ to approximate $D$ in this case, it is in fact only necessary that $\mathcal{W}\mathcal{W}^\dagger$ approximates $DD^\dagger$, allowing some freedom in the learned matrix $\mathcal{W}$.

The operator $\mathcal{W}$ is built as a composition of simple linear operators $\mathcal{W} = \mathcal{W}_n \circ \ldots \circ \mathcal{W}_1$, where each $\mathcal{W}_k$ has only local interactions along a fixed dimension, in a fixed direction (that is, with only positive or negative offsets, but not both), allowing the determinant of each matrix to be efficiently computed. The components of each operator $\mathcal{W}_k$ are parametrized by two P-fields, produced from learned translation-equivariant functions of $\phi$. More specifically, we consider $2d$ types of operators, where each type is defined by a sign $s = \pm 1$ and a choice of one of the $d$ lattice directions. For the two-dimensional application described below, there are thus four distinct operator types. The different types of operators are applied alternatingly in the composition, but the specific order can be chosen arbitrarily. The operator type with couplings in the spatial direction and sign $s$ thus updates a field $\chi$ by

$$(\mathcal{W}\chi)_{ij} = a_{ij}\chi_{ij} + b_{ij}\chi_{i+s,j} \tag{6.26}$$

with periodic boundary conditions along the space dimension: $\chi_{L+1,j} = \chi_{1,j}$ and $\chi_{0,j} = \chi_{L,j}$. An operator with temporal couplings updates a field $\chi$ by

$$(\mathcal{W}\chi)_{ij} = a_{ij}\chi_{ij} + b_{ij}\chi_{i,j+s} \tag{6.27}$$

with antiperiodic boundary conditions along the time dimension: $\chi_{i,L+1} = -\chi_{i,1}$ and $\chi_{i,0} = -\chi_{i,L}$. This construction may be understood as a convolutional layer with appropriate boundary conditions and an additional constraint on the kernel to have non-zero entries only in the center and at one of the $2d$ adjacent sites.

With these definitions, each operator $\mathcal{W}_k$ is block diagonal (for a suitable choice of basis). Each block is of the form

$$\begin{bmatrix} a_1 & & & \pm b_1 \\ b_2 & a_2 & 0 & \\ & \dots & \dots & \\ & 0 & & \\ & & b_L & a_L \end{bmatrix}, \tag{6.28}$$

where we have dropped a (spatial or time) index to simplify the notation. The determinant of each block is simply $\Pi_h a_h \pm \Pi_h b_h$, indicating that the Jacobian determinant associated with the full composition can be tractably computed.

### Convex potential flows

Because of the non-local nature of the effective action, we consider an alternative flow architecture to produce a model distribution $q(\phi)$ approximating $p(\phi)$. Convex Potential Flows (CPF) are normalizing flows defined via the gradients of a potential that is strongly convex and twice differentiable almost everywhere [254, 255]. Strong convexity of the potential on a convex support $\mathcal{X}$ guarantees the flow to be invertible on $\mathcal{X}$. It can be shown that this family of normalizing flows is a universal density approximator and optimal in the optimal transport theory sense [255]. We can parametrize strongly convex functions by neural networks with mild constraints on their architecture and weights [256]. Specifically, using $L(x)$ to denote a linear layer and $L^+(x)$ a linear layer with positive weights, an input-convex neural network can be defined as

$$u(z) = L^+_{k+1}(A(x_k)) + L_{k+1}(z) \qquad x_k = L^+_k(A(x_{k-1})) + L_k(z) \qquad x_1 = L_1(z) \ , \tag{6.29}$$

where $A$ is a non-decreasing, convex activation function.

Given a convex potential function $u : \mathcal{X} \to \mathbb{R}$, we define the map

$$[f(z)]_i = \frac{\partial}{\partial z_i} u(z) \ , \tag{6.30}$$

where the index $i$ specifies how each degree of freedom of $z$ is mapped. Starting from a base density $r(z)$, the resulting probability density produced by mapping through $f$ follows as

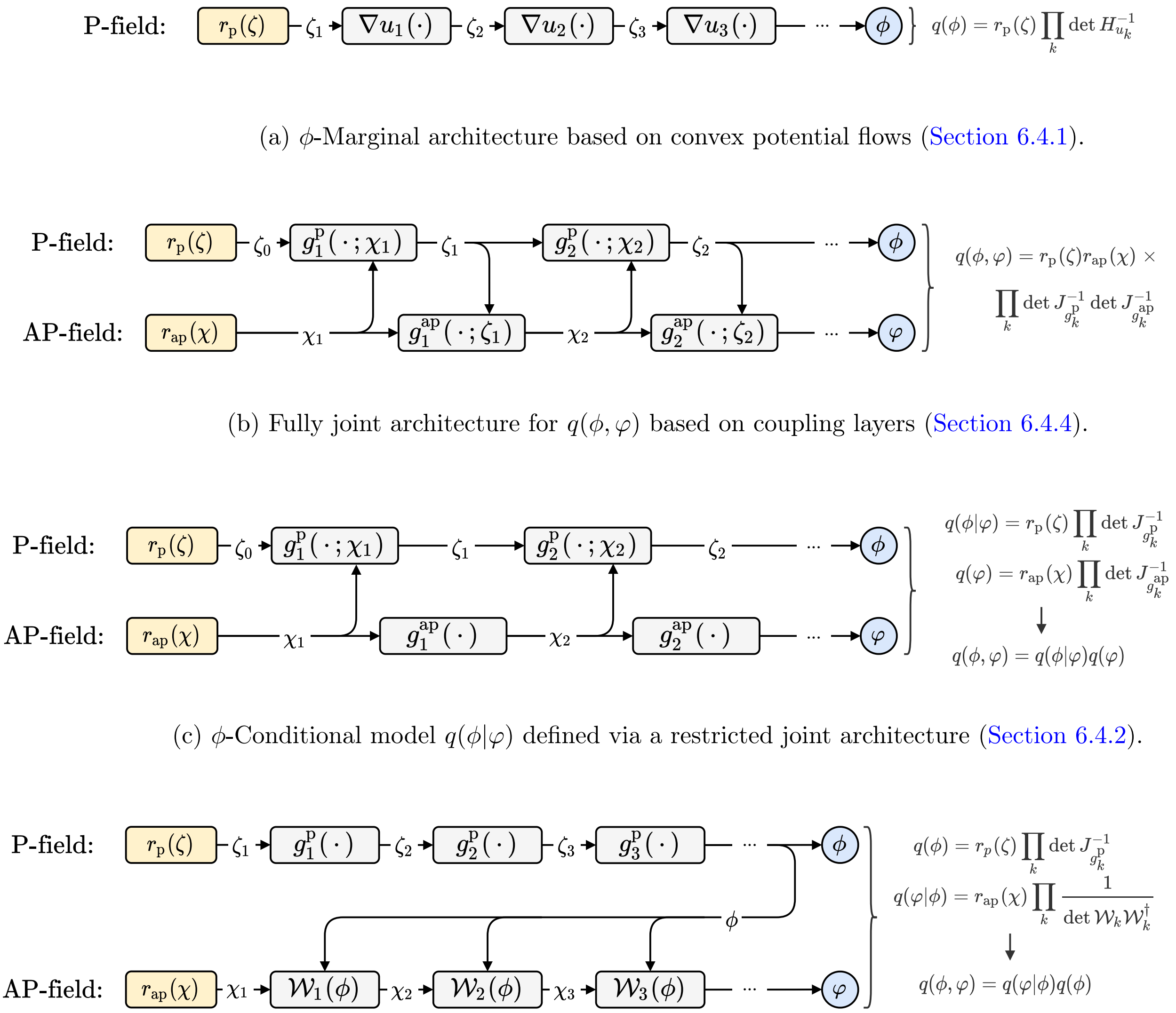
$$q(x) = r(z) \det H_u(z)^{-1} \ , \tag{6.31}$$

where $x = f(z)$ and $H_u(z) = \frac{\partial^2}{\partial z_i \partial z_j} u(z)$ is the Hessian matrix of $u(z)$. Training by minimizing $D_{\mathrm{KL}}(q||p)$ between the model $q$ and a target density $p$ only requires the gradients $\nabla_\theta \log\det H_u(x)$ with respect to the model's parameters $\theta$. Since the Hessian is symmetric and positive-definite for strongly convex potentials, we can directly employ a stochastic trace estimator [255, 257],

$$\begin{aligned}
\nabla \log\det H_u(x) &= \nabla \mathrm{Tr} \log H_u(x) \\
&= \mathrm{Tr}\left[H_u(x)^{-1} \nabla H_u(x)\right] \\
&= \mathbb{E}_{\chi \sim e^{-\chi^\dagger \chi}}[(H_u^{-1}(x)\chi)^\dagger \nabla H_u(x) \chi] \ .
\end{aligned} \tag{6.32}$$

The sample mean over noise vectors $\chi$ can be used to estimate this quantity in practice, and the inverse Hessian applied in $H_u(x)^{-1}\chi$ can be efficiently computed by the application of the conjugate-gradient method. Note that this estimator only requires the computation of Hessian-vector products $H\chi$, which is particularly convenient when the Hessian is sparse.

CPFs can be straightforwardly applied to construct flows that sample bosonic fields $\phi$. They can also be constrained to be translation-equivariant by using the aforementioned convolutional layers with periodic padding. In contrast to coupling layers, the CPF potential is a scalar function based on global information, which may result in transformations of the field $\phi$ that can in general be quite non-local. Evaluating the model probability density for use in asymptotically exact Markov chains requires a precise approximation of the log-det Hessian to avoid systematic errors. An exact calculation of the determinant is feasible only for small lattice volumes. For larger field configurations, one could apply a more scalable estimator, such as the one based on Lanczos tridiagonalization and the quadrature method described in [258].


P-field:
$r_{\mathrm{p}}(\zeta)$
$\zeta_1$
$\nabla u_1(\cdot)$
$\zeta_2$
$\nabla u_2(\cdot)$
$\zeta_3$
$\nabla u_3(\cdot)$
···
$\phi$
$q(\phi) = r_{\mathrm{p}}(\zeta) \prod_k \det H_{u_k}^{-1}$
P-field:
AP-field:
$r_{\mathrm{p}}(\zeta)$
$\zeta_0$
$g_1^{\mathrm{p}}(\,\cdot\,;\chi_1)$
$\zeta_1$
$g_2^{\mathrm{p}}(\,\cdot\,;\chi_2)$
$\zeta_2$
$r_{\mathrm{ap}}(\chi)$
$\chi_1$
$g_1^{\mathrm{ap}}(\,\cdot\,;\zeta_1)$
$\chi_2$
$g_2^{\mathrm{ap}}(\,\cdot\,;\zeta_2)$
$\phi$
$\varphi$
$q(\phi,\varphi) = r_{\mathrm{p}}(\zeta) r_{\mathrm{ap}}(\chi) \times$
$\prod_k \det J_{g_k^{\mathrm{p}}}^{-1} \det J_{g_k^{\mathrm{ap}}}^{-1}$
P-field:
AP-field:
$r_{\mathrm{p}}(\zeta)$
$\zeta_0$
$g_1^{\mathrm{p}}(\,\cdot\,;\chi_1)$
$\zeta_1$
$g_2^{\mathrm{p}}(\,\cdot\,;\chi_2)$
$\zeta_2$
$r_{\mathrm{ap}}(\chi)$
$\chi_1$
$g_1^{\mathrm{ap}}(\,\cdot\,)$
$\chi_2$
$g_2^{\mathrm{ap}}(\,\cdot\,)$
$\phi$
$\varphi$
$q(\phi|\varphi) = r_{\mathrm{p}}(\zeta) \prod_k \det J_{g_k^{\mathrm{p}}}^{-1}$
$q(\varphi) = r_{\mathrm{ap}}(\chi) \prod_k \det J_{g_k^{\mathrm{ap}}}^{-1}$
$q(\phi,\varphi) = q(\phi|\varphi) q(\varphi)$
P-field:
AP-field:
$r_{\mathrm{p}}(\zeta)$
$\zeta_1$
$g_1^{\mathrm{p}}(\,\cdot\,)$
$\zeta_2$
$g_2^{\mathrm{p}}(\,\cdot\,)$
$\zeta_3$
$g_3^{\mathrm{p}}(\,\cdot\,)$
$\phi$
$r_{\mathrm{ap}}(\chi)$
$\chi_1$
$\mathcal{W}_1(\phi)$
$\chi_2$
$\mathcal{W}_2(\phi)$
$\chi_3$
$\mathcal{W}_3(\phi)$
$\varphi$
$q(\phi) = r_p(\zeta) \prod_k \det J_{g_k^{\mathrm{p}}}^{-1}$
$q(\varphi|\phi) = r_{\mathrm{ap}}(\chi) \prod_k \frac{1}{\det \mathcal{W}_k \mathcal{W}_k^\dagger}$
$q(\phi,\varphi) = q(\varphi|\phi) q(\phi)$

(a) $\phi$-Marginal architecture based on convex potential flows (Section 6.4.1).

(b) Fully joint architecture for $q(\phi, \varphi)$ based on coupling layers (Section 6.4.4).

(c) $\phi$-Conditional model $q(\phi|\varphi)$ defined via a restricted joint architecture (Section 6.4.2).

(d) Autoregressive model $q(\phi)q(\varphi|\phi)$ defined via coupling layers and linear flows (Section 6.4.3).

Figure 6.4: Architectures for the flow-based models defined in Section 6.4 for each sampling approach. Note that each coupling layer $g_k^{\mathrm{p}}$ or $g_k^{\mathrm{ap}}$ employs masking of the updated field as shown in Figure 6.2, such that the frozen components of the field are included as input to context functions. Superscripts on coupling layers indicate the translational equivariance structure of coupling layer inputs and outputs (either consistently transforming as P-fields or AP-fields).

## 6.4 Flow architectures

We next define particular architectures for modeling each of the distributions required for the four sampling approaches introduced in Section 6.2. While the space of possible architectures that may be defined from the building blocks of Section 6.3.2 is large and the present discussion is not exhaustive, the use of each sampling method and each building block is demonstrated at least once. The architectures for each approach detailed in this section are illustrated in Figure 6.4.

### 6.4.1 Modeling $p(\phi)$ for $\phi$-marginal sampling

The $\phi$-marginal sampler defined in Section 6.2.1 requires a flow whose model distribution $q(\phi)$ approximates $p(\phi)$. Such a flow only needs to manipulate P-fields. We build this $\phi$-marginal model using a composition of CPF layers, where the output of each layer is defined by computing the gradient of a potential $u_i(\cdot)$ (see Section 6.3.2). These layers act on samples $\zeta$ drawn from some base distribution $r_{\mathrm{p}}(\zeta)$. Figure 6.4a depicts this type of $\phi$-marginal architecture defined by a composition of CPFs acting on $\zeta$. Each $u_i$ contributes a determinant factor $\det H_{u_i}^{-1}$ to $q(\phi)$ in terms of the Hessian $H_{u_i}^{ab} = \frac{\partial^2 u_i(z)}{\partial z_a \partial z_b}$, such that

$$q(\phi) = r_{\mathrm{p}}(\zeta) \prod_i \det H_{u_i}^{-1} \; . \tag{6.33}$$

As discussed in Section 6.3.1, this marginal model is extended to the joint density $q(\phi, \varphi) = q(\phi)p(\varphi|\phi)$ for training. The density cannot be computed efficiently due to the determinants involved in the definition of $q(\phi)$ as well as in the normalizing constant of $p(\varphi|\phi)$, but the flow is nevertheless trainable using stochastic estimates of the gradients. For sampling, the joint density itself may also be estimated using stochastic approximations of the determinant factors.

The architecture of the convex potential network $u(\phi)$ is based on Equation (6.29) and is modified appropriately to account for the periodic boundary conditions. It consists of $K$ layers of convolutions of the form

$$\begin{aligned} h_1 &= L_1(\phi) \\ h_{k+1} &= L_k^+(\mathrm{SoftPlus}(\mathrm{ActNorm}(h_k))) + L_k(\phi) \\ u(\phi) &= w_1 \mathrm{Sum}(h_K) + w_2 \frac{\|\phi\|^2}{2} \; , \end{aligned} \tag{6.34}$$

where $L_j$ is a convolution layer with periodic boundary conditions and unconstrained weights; $L_j^+$ is a convolution layer with periodic boundary conditions and positive-only weights; $\mathrm{ActNorm}(x) = (x - \mu)/\sigma$ is layer that normalizes its inputs using a learnable offset $\mu$ and scale $\sigma$, where $\mu$ and $\sigma$ are initialized as the mean and standard deviation of the inputs of an initialization batch [259]; $w_1, w_2$ are learnable weights used to control closeness of the flow to the identity map at initialization. The use of periodic boundary conditions for $L_j$ and $L_j^+$ and the final Sum operation ensures that $u(\phi)$ is invariant to translations.

### 6.4.2 Modeling $p(\phi|\varphi)$ for Gibbs sampling

The Gibbs sampling scheme described in Section 6.2.2 utilizes the exact conditional $p(\varphi|\phi)$ and a modeled conditional density $q(\phi|\varphi)$. A $\varphi$-marginal model $q(\varphi)$ is required to extend $q(\phi|\varphi)$ to the joint distribution $q(\phi|\varphi)q(\varphi)$ for training. This simultaneous modeling of $q(\phi|\varphi)$ and $q(\varphi)$ can be achieved by using a fully joint architecture with restricted information flow, as shown in Figure 6.4c.

The model consists of a prior distribution over the base configurations $\zeta, \chi$ denoted by $r_{\mathrm{p}}(\zeta)$ and $r_{\mathrm{ap}}(\chi)$, followed by the application of two types of affine coupling layers. First, the layers $g_k^{\mathrm{p}}(\cdot; \chi_k)$ update the P-field configuration conditioned on the AP-field, along with the frozen components of $\zeta_k$, to produce $q(\phi|\varphi)$ as:

$$q(\phi|\varphi) = r_{\mathrm{p}}(\zeta) \prod_k \det J_{g_k^{\mathrm{p}}}^{-1}, \tag{6.35}$$

where $J_{g_k^{\mathrm{p}}}$ is the Jacobian for coupling $g_k^{\mathrm{p}}$. Second, the couplings $g_k^{\mathrm{ap}}(\cdot)$ transform the AP-field $\chi$ conditioned solely on its frozen components to obtain $q(\varphi)$:

$$q(\varphi) = r_{\mathrm{ap}}(\chi) \prod_k \det J_{g_k^{\mathrm{ap}}}^{-1} \tag{6.36}$$

To conditionally re-sample $\phi$ from $q(\phi|\varphi)$ while leaving $\varphi$ unchanged, the bosonic prior variable is re-sampled and the output of the flow is re-evaluated while holding the pseudofermionic prior variable $\chi$ fixed. When $\varphi$ is re-sampled from $p(\varphi|\phi)$ in the alternate step of the Gibbs sampler, it is important to update the value of $\chi$ by passing $\varphi$ through the inverse of the bottom branch of the flow depicted in the figure. This allows future re-sampling of $\phi$ as well as the calculation of the conditional probability density defined by the model.

### 6.4.3 Autoregressive modeling of $p(\phi, \varphi) = p(\phi)p(\varphi|\phi)$

Section 6.2.3 defined an independence sampler based on an autoregressive joint model with the probability density given by $q(\phi, \varphi) = q(\phi)q(\varphi|\phi)$. We implement $q(\phi)$ using masked affine coupling layers whose parameters are given by convolutional networks satisfying translational equivariance through standard periodic boundaries, as described above. On the other hand, $q(\varphi|\phi)$ is implemented using a deep linear flow consisting of learned linear operators $\mathcal{W}_k(\phi)$, as detailed in Section 6.3.2. The parameters of these linear operators are all P-fields obtained by similar periodic convolutional networks. The full joint model is given by the autoregressive combination of these two models, i.e. drawing $\phi$ from the affine model with distribution $q(\phi)$, then drawing $\varphi$ from the conditional deep linear flow with distribution $q(\varphi|\phi)$, as shown in Figure 6.4d. The marginal model is defined by sampling $\zeta$ from the prior distribution $r_{\mathrm{p}}(\zeta)$, then applying the sequence of coupling layers $g_k^{\mathrm{p}}(\cdot)$ such that the marginal model probability density $q(\phi)$ is given by:

$$q(\phi) = r_{\mathrm{p}}(\zeta) \prod_k \det J_{g_k^{\mathrm{p}}}^{-1} \, . \tag{6.37}$$

The conditional linear flow is defined by sampling $\chi$ from the prior distribution $r_{\mathrm{ap}}(\chi)$ and applying the linear operators $\mathcal{W}_k(\phi)$ to obtain the model density

$$q(\varphi|\phi) = r_{\mathrm{ap}}(\chi) \prod_k \frac{1}{\det \mathcal{W}_k \mathcal{W}_k^\dagger} \ . \tag{6.38}$$

We define $r_{\mathrm{ap}}(\chi) = \frac{1}{Z_{\mathcal{N}}} e^{-\chi^\dagger \chi}$ to match the choice for the linear operator flow in Equation (6.24).

Note that the learned components in this approach may also be combined in novel ways. For example, it is possible to discard the conditional flow with distribution $q(\varphi|\phi)$ after training and simply use $q(\phi)$ for $\phi$-marginal sampling as described in Sections 6.2.1 and 6.4.1. This may be advantageous in situations where gradients from an exactly sampleable distribution are not available and training must be fully variational. On the other hand, the conditional deep linear flow may be used by itself as a determinant estimator for given configurations $\phi$.

### 6.4.4 Fully joint modeling of $p(\phi, \varphi)$

Finally, we construct a model that simultaneously samples $\phi$ and $\varphi$ in a fully joint approach, which can be employed in the exact sampler defined in Section 6.2.4. The joint model implemented here is constructed from affine coupling layers that alternatingly transform the bosonic fields conditioned on the pseudofermionic fields, and vice versa, as shown in Figure 6.4b. The model is defined to sample $\zeta$ and $\chi$ from the prior distributions $r_{\mathrm{p}}(\zeta)$, $r_{\mathrm{ap}}(\chi)$ and subsequently apply alternating layers. Coupling layers $g_k^{\mathrm{p}}(\cdot, \chi_k)$ transform the P-field base configuration $\zeta$ conditioned on its frozen components and the AP-field configuration $\chi_k$, while couplings $g_k^{\mathrm{ap}}(\cdot, \zeta_k)$ update the AP-field base configuration $\chi_k$ conditioned on its frozen components and $\zeta_k$. This gives rise to the joint density

$$q(\phi, \varphi) = r_{\mathrm{p}}(\zeta) r_{\mathrm{ap}}(\chi) \prod_k \det J_{g_k^{\mathrm{p}}}^{-1} \det J_{g_k^{\mathrm{ap}}}^{-1} \ . \tag{6.39}$$

## 6.5 Application to Yukawa theory

In this section, we present the application of the flow-based sampling schemes introduced in Sections 6.2 to 6.4 to the scalar Yukawa model described in Section 2.4. Here, we use the dimensionful formulation of the action for the scalar field, but from now on employ the following conventional form for notational simplicity:

$$S_B(\phi) = \sum_{x \in \Lambda} \Big[ -2 \sum_{\mu=1}^{d} \phi(x)\phi(x+\hat{\mu}) + (m^2 + 2d)\phi(x)^2 + \lambda \phi(x)^4 \Big] \ , \tag{6.40}$$

where $\Lambda$ again denotes the set of lattice sites, $m$ the bare scalar mass parameter, $\lambda$ the coupling, and $d$ the dimension. All results reported in this work are computed on a $16 \times 16$ lattice geometry using the two choices of action parameters in the symmetric phase given in Table 6.1. For this theory and lattice discretization, there is no additive renormalization to the bare fermion mass $m_f$. Accordingly, we directly probe the case of vanishing mass by setting $m_f = 0$. The first set of parameters, for which the Yukawa coupling is chosen to be $g = 0.1$, already provides a realistic test scenario in the sense that the average ratio of fermionic to scalar force magnitudes is around 3%, which is similar to the ratio of fermionic to gauge forces reported in the literature for some lattice QCD computations; see e.g. [260, 261]. The second choice with $g = 0.3$ features a much larger force ratio amounting to about 39%, which thus provides a testbed for theories with more prominent fermionic effects. For simplicity, we will refer to these two parameter choices by the associated value of the Yukawa coupling $g$.

### 6.5.1 Even-odd preconditioning

We employ an even-odd preconditioning scheme for the Dirac operator for all models except for the autoregressive model using linear operators. In contrast to the default lexicographic ordering, sorting lattice sites into even and odd allows to bring the matrix into a form that is amenable to an explicit block factorization of the determinant, which leads to improvements in the conditioning and solver performance. This reduces the variance and cost of computing the pseudofermion action required for optimizing models and sampling. Most previous work on improved orderings has focused on techniques for Wilson fermions in the context of gauge theory [27, 29, 262], but the same insights can be applied to the staggered fermion formulation used here.

Ordering lattice sites into even and odd subsets allows writing the Dirac matrix $D$ as a $2 \times 2$ block matrix of the form

$$D = \begin{pmatrix} m_f + g\phi_o & D_{oe} \\ D_{eo} & m_f + g\phi_e \end{pmatrix} \equiv \begin{pmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{C} & \mathcal{D} \end{pmatrix} , \tag{6.41}$$

where we denote the blocks as $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}$ for simplicity. The constant blocks $\mathcal{B} = D_{oe}$ and $\mathcal{C} = D_{eo}$ couple odd to even sites and vice versa, and $\phi_o$ and $\phi_e$ indicate the components of $\phi$ respectively associated with odd and even sites of the lattice. This form allows a more efficient stochastic approximation of the determinant by decomposing it into the determinant of either diagonal block and the associated Schur complement as

$$\begin{aligned} \det D &= \det(\mathcal{A}) \det(\mathcal{D} - \mathcal{C}\mathcal{A}^{-1}\mathcal{B}) \\ &= \det(\mathcal{A}\mathcal{C}^{-1}\mathcal{D} - \mathcal{B}) \det(\mathcal{C}) , \end{aligned} \tag{6.42}$$

or, equivalently,

$$\begin{aligned} \det D &= \det(\mathcal{D}) \det(\mathcal{A} - \mathcal{B}\mathcal{D}^{-1}\mathcal{C}) \\ &= \det(\mathcal{D}\mathcal{B}^{-1}\mathcal{A} - \mathcal{C}) \det(\mathcal{B}) . \end{aligned} \tag{6.43}$$

| $V$ | $m^2$ | $\lambda$ | $g$ | $m_f$ | $\langle\lvert M\rvert\rangle$ | $\langle\lvert\bar{\psi}\psi\rvert\rangle$ | Force ratio |
|---|---|---|---|---|---|---|---|
| $16^2$ | $-4.00$ | 6.0 | 0.1 | 0 | 0.0733(1) | 0.0159(1) | 3% |
| $16^2$ | $-1.55$ | 2.4 | 0.3 | 0 | 0.0791(1) | 0.0490(1) | 39% |

Table 6.1: The two parameter choices for the reported numerical studies and the associated average absolute magnetization and chiral condensate computed with HMC. All uncertainties reported in this work are obtained using data blocking to account for autocorrelations and applying the statistical jackknife method. Force ratios are determined by dividing the average $L^2$-norms of fermionic and bosonic force vectors.

Rewriting from the first to the second form in Equations (6.42) and (6.43) ensures that the resulting expression does not involve terms that mix $\mathcal{A}$ and $\mathcal{D}$ with their respective inverses. This may lead to numerical instabilities if $m_f = 0$, which can result in ill-conditioned $\mathcal{A}$ and $\mathcal{D}$. Since $\mathcal{B}$ and $\mathcal{C}$ are constant, the terms $\det(\mathcal{B}), \det(\mathcal{C})$ drop out of the path integral and thus do not affect acceptance probabilities or gradients for optimization. Hence, they can be ignored for the purpose of training and sampling flow models.[8]

The reduced $V/2 \times V/2$ form of the Dirac operator makes determinant estimation significantly cheaper while keeping the additional computational overhead minimal. Half of the pseudofermion degrees of freedom completely decouple from the scalar field and can be discarded. The reduced operator partially retains the original periodic and antiperiodic boundary conditions when applied to the even or odd sub-lattices, respectively, reducing to a translation symmetry for even shifts. When utilizing affine coupling layers with a checkerboard mask, it is exactly this subset of the translational symmetry group that is preserved, and the translationally equivariant architecture is directly applicable to learning a distribution over the reduced subset of pseudofermions.

The improvement can be pushed to higher order by noting that the diagonal matrix elements of the preconditioned Dirac operator are close to unity, which makes it possible to employ an ILU preconditioning scheme [27, 29]. It relies on the fact that the preconditioning matrices for the even-odd ordering step described above can be computed explicitly, which is not generally true for other ordering schemes. Though also originally designed for the Wilson Dirac operator, the same procedure can again be applied to the staggered fermion formulation. Since ILU preconditioning breaks the translation symmetry of the pseudofermion action completely, it is not directly compatible with any of our equivariant flow constructions that target the distribution of $\varphi$. However, in an experiment modeling the even-odd preconditioned $\phi$-marginal distribution using an affine coupling layer model, additional ILU preconditioning for the same architecture led to a moderately improved acceptance rate.

[8] If one is interested in overall estimates of $\log Z$, the constant contributions from these terms must then be included.

### 6.5.2 Model architectures

For each of the four sampling approaches outlined in Section 6.2 and corresponding model architectures detailed in Section 6.4, specific models are created for both choices of target action parameters given in Table 6.1:

- For the sampling scheme described in Section 6.2.1, a CPF model is constructed defining a $\phi$-marginal distribution $q(\phi)$ approximating the corresponding $p(\phi)$. The model architecture and training follow the generic procedure outlined in Section 6.4.1;
- To build a conditional model $q(\phi|\varphi)$ for the Gibbs sampler described in Section 6.2.2, a restricted affine coupling layer flow is implemented as described in Section 6.4.2. To achieve translational equivariance, the method of group averages described in Section 6.3.2 is employed for each individual context function;
- To produce an autoregressive joint model density $q(\phi, \varphi) = q(\phi)q(\varphi|\phi)$ for the sampling scheme described in Section 6.2.3,a model consisting of affine coupling layers followed by learned equivariant linear transformations is constructed as described in Section 6.4.3;
- For the fully joint sampling scheme described in Section 6.2.4, a model with unrestricted affine coupling layers acting on both $\phi$ and $\varphi$ is implemented, using translation-equivariant convolutions as described in Section 6.3.2. This results in a fully joint model distribution $q(\phi, \varphi)$ as detailed in Section 6.4.4.

The models are optimized for each approach based on the joint Kullback-Leibler divergence discussed in Section 6.3.1. Prior distributions for the initial P-field $\zeta$ and AP-field $\chi$, where they are used according to Figure 6.4, are Gaussians of the form

$$\begin{aligned} r_{\mathrm{p}}(\zeta) &= \frac{1}{\mathcal{Z}_\zeta} e^{-\zeta^\dagger \zeta/(\sigma_\zeta)^2} \\ \text{and } r_{\mathrm{ap}}(\chi) &= \frac{1}{\mathcal{Z}_\chi} e^{-\chi^\dagger \chi/(\sigma_\chi)^2} \ , \end{aligned} \tag{6.44}$$

with specific values of $\sigma_\zeta$ and $\sigma_\chi$ for each model chosen to enhance the training stability and convergence.

Further details about the hyperparameters and training procedure for each of the the models can be found in Appendix B. The chosen settings were found to work well empirically, and an exhaustive search over the available parameter space is beyond the scope of this proof-of-principle study. Nevertheless, it can be expected that tuning the various model hyperparameters may further improve the reported performance metrics, which will be the subject of future work.

| MCMC Approach | Modeled targets | Flow model | Acc. rate | $\langle\lvert M\rvert\rangle$ | $\langle\lvert\bar{\psi}\psi\rvert\rangle$ | $\tau_M^{\mathrm{int}}$ | $\tau_{\bar{\psi}\psi}^{\mathrm{int}}$ |
|---|---|---|---|---|---|---|---|
| $\phi$-Marginal (Section 6.2.1) | $p(\phi)$ | Section 6.4.1 | 92% | 0.0734(1) | 0.0159(1) | 0.72(1) | 0.71(1) |
| | | | 92% | 0.0792(1) | 0.0491(1) | 0.67(1) | 0.67(1) |
| Gibbs (Section 6.2.2) | $p(\phi\vert\varphi)$ | Section 6.4.2 | 60% | 0.0735(1) | 0.0160(1) | 2.02(4) | 2.02(3) |
| | | | 44% | 0.0792(1) | 0.0490(1) | 2.74(4) | 2.73(4) |
| Autoregressive (Section 6.2.3) | $p(\phi), p(\varphi\vert\phi)$ | Section 6.4.3 | 53% | 0.0731(1) | 0.0159(1) | 2.16(3) | 2.16(3) |
| | | | 43% | 0.0790(1) | 0.0489(1) | 3.62(7) | 3.60(7) |
| Fully Joint (Section 6.2.4) | $p(\phi, \varphi)$ | Section 6.4.4 | 37% | 0.0733(1) | 0.0159(1) | 4.98(11) | 4.98(11) |
| | | | 31% | 0.0791(1) | 0.0490(1) | 8.73(30) | 8.67(30) |

Table 6.2: Sampling performance metrics and observables for all approaches, computed from 100 Markov chains with 10k proposals each, where the first 1k are discarded for thermalization. For each model, the first row shows results obtained for $g = 0.1$ and the second row for $g = 0.3$, respectively. For comparison, the values obtained with HMC listed in Table 6.1 are consistent with the measurements from our models. Autocorrelation times $\tau^{\mathrm{int}}$ are computed for each of the 100 chains and then averaged, and errors are obtained with statistical jackknife. The results are discussed in more detail in Section 6.5.3. All models except the autoregressive make use of even-odd preconditioning of the action.

### 6.5.3 Discussion and comparison of sampling schemes

After optimization, each of the models is used to construct asymptotically exact samplers for their respective target distributions according to the four schemes given in Section 6.2. For each case, 100 distinct Markov chains are produced consisting of 10k steps each, of which the first 1k steps are discarded for thermalization. These Markov chains are used for observable measurements and to investigate and compare metrics of the efficiency of sampling via each of these methods.

First, we confirm that each of the observables described above are measured to be consistent across sampling schemes and with HMC baseline results. Calculations of $\langle|M|\rangle$ and $\langle|\bar{\psi}\psi|\rangle$ using each of the generated ensembles are detailed in Table 6.2 and are all consistent with the results obtained through HMC. The scalar and fermionic two-point correlators produced by the four exact Monte Carlo sampling schemes models are also consistent with the HMC baseline, as shown in Figures 6.5 and 6.6.

For the various sampling approaches, Table 6.2 compares the autocorrelations of the magnetization and chiral condensate, as well as the Markov chain acceptance rates. Based on the autocorrelation function $\Gamma_X(\tau)$ of an observable $X$ defined in Equation (2.4), the integrated autocorrelation is

$$\tau_X^{\text{int}} = \frac{1}{2} + \lim_{\tau^{\max}\to\infty} \sum_{\tau=1}^{\tau^{\max}} \frac{\Gamma_X(\tau)}{\Gamma_X(0)} \,. \tag{6.45}$$

The sum can be truncated at a sufficiently large $\tau^{\max}$ due to the exponential suppression of $\Gamma_X(\tau)$; $1 \ll \tau^{\max} \ll N$ should be satisfied to ensure that the values of $\Gamma_X(\tau)$ are reliable. For the $\tau^{\text{int}}$ values reported in this work, we use the Madras-Sokal windowing procedure [264] to choose a suitable $\tau^{\max}$ by identifying the earliest point where $c\tau^{\text{int}} \leq \tau^{\max}$, with $c = 10$. The integrated autocorrelation times $\tau_M^{\text{int}}$ and $\tau_{\bar{\psi}\psi}^{\text{int}}$ are given in Table 6.2 together with acceptance rates for all sampling schemes.

To understand the relative performance of the four sampling approaches, we note that the dependence of the acceptance rate and autocorrelations on model quality is quite distinct in several of these approaches. For one, the $\phi$-marginal sampler involves an exact determinant measurement in the sampling step used for the numerical study above, which is not expected to scale efficiently. If replaced with the pseudo-marginal estimator discussed in Section 6.2.1, the variance of the noisy estimates of each determinant would degrade the statistical performance achieved by even an optimally trained model and improved estimators, in particular when encountering large condition numbers. This is an obstacle to working with light fermion masses or field configuration geometries with many lattice sites (e.g. near the thermodynamic limit) independent of the challenge of training accurate model approximations to the target distribution. Thus, the relatively higher acceptance rates and lower autocorrelation times achieved by the $\phi$-marginal sampler must be contrasted against the potentially difficult scaling challenges or requirements for more precise stochastic estimators. By comparison, there is no non-trivial upper bound on the acceptance rate of the other sampling approaches, and they would achieve 100% if perfect model distributions were constructed.

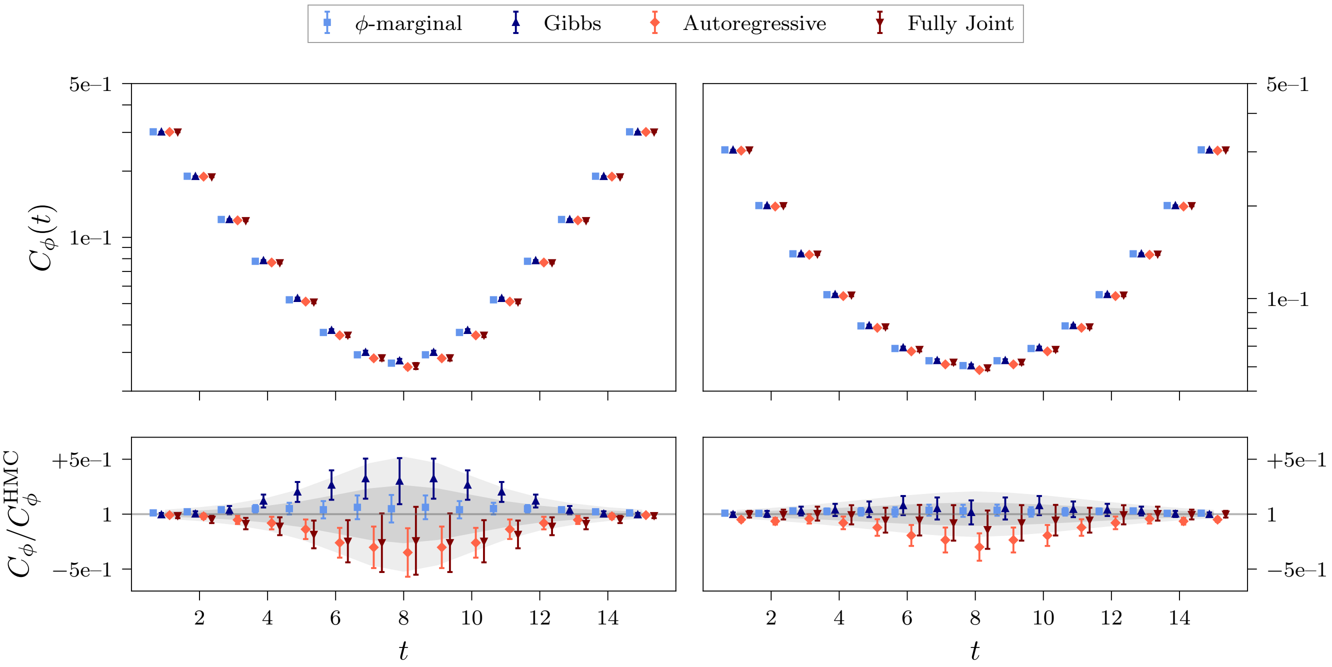


Figure 6.5: Connected two-point correlation functions of the scalar field (projected to zero spatial momentum in each time slice $t$) for each model and choice of action parameters, computed from 100 Markov chains with 9k configurations each. Error estimates are obtained using data blocking with a bin size of 100 and applying statistical jackknife. Left: $g = 0.1$, right: $g = 0.3$. Bottom panels show the ratio of each data point to the HMC baseline, where the shaded regions correspond to the $1\sigma$ and $2\sigma$ uncertainty bands of the HMC results. For both the scalar correlators here and the fermionic ones in Figure 6.6, Hotelling's t-squared statistic [263] comparing each flow model result to the HMC baseline finds results to be consistent with correlated statistical fluctuations.

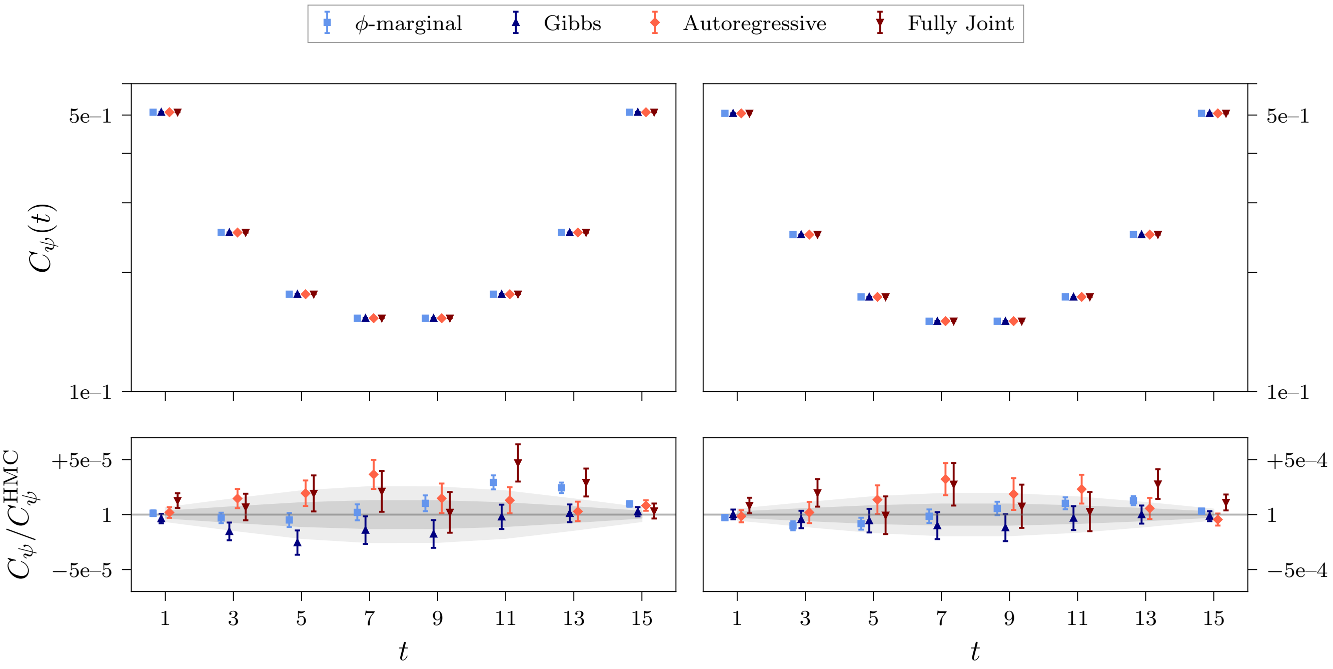


Figure 6.6: Average fermionic two-point correlation in the time direction for each model and choice of action parameters using the same configurations and data blocking as for Figure 6.5. Left: $g = 0.1$, right: $g = 0.3$. The choice of odd $t$ selects staggered spinor components at the sinks that give a non-zero average correlation with the source at $x = 0$. Shaded regions in the bottom panel again depict the $1\sigma$ and $2\sigma$ uncertainties of the HMC baseline results.

Among these three approaches, the Gibbs sampler must also be further contrasted against the autoregressive and fully joint samplers. In particular, the remaining conditional structure of updates to $\phi$ and $\varphi$ in the Gibbs sampler results in autocorrelations even if the acceptance rate is 100%. The magnitude of these residual autocorrelations may be small, but nevertheless puts a bound on the performance that is theoretically achievable by a Gibbs sampler, even in the asymptotic limit of perfect models of the involved distributions. Thus only joint models (either autoregressive or fully joint) can completely eliminate autocorrelations in the ideal limit of perfect models. In practice, however, the distinctions between joint models and Gibbs sampling may be minor. For example, the results presented in Table 6.2 demonstrate that at the similar acceptance rates of roughly 40%–50% for the Gibbs and autoregressive samplers, the integrated autocorrelation times for the magnetization and condensate are similar, despite the additional autocorrelations introduced by the particular conditional structure of the Gibbs sampling scheme. The fully joint sampler shows a lower acceptance rate and greater autocorrelations, indicating that the differences are largely based on the model approximation qualities and the effect of the conditional structure on autocorrelations is largely negligible.

The particular flow-based models implemented to approximate the various distributions used for the four sampling approaches also have distinct scaling prospects. It has been found in previous work [175] that flows based on coupling layers using convolutional networks may be easily transferred between different lattice volumes and thereby trained efficiently. This generalizability applies to the affine coupling layer implementations used for the Gibbs, autoregressive, and fully joint samplers described in this work. The CPF implementation for $\phi$-marginal sampling is also based on convolutional networks for the construction of the convex potentials, thus enabling efficient measurements of these potentials at all lattice volumes. However, in this case computing the Jacobian of the transformation to calculate $q(\phi)$ is potentially expensive, because it requires the evaluation of the Hessian of each $u_i$. Stochastic estimation of these Hessian factors may introduce additional noise in exact sampling schemes based on these particular flow architectures, which could be prohibitive in scaling this approach to large lattices.

In summary, these results numerically demonstrate the effectiveness of the proposed flow models and sampling schemes. The observed performance differences cannot immediately be attributed to inherent advantages of the chosen building blocks, but may also depend strongly on the model implementation details and theory-specific characteristics. The situation may also be quite different for larger volumes and dimensions as well as other types of fields and interactions, and disentangling the effects of implementation details from asymptotic scaling properties will be the subject of future research. Furthermore, there is a large space of possible combinations of the building blocks introduced here that could be explored in future work to determine models that may have more efficient training, sampling, and scaling prospects. While an exhaustive search over this space is beyond the scope of this exploratory work, the present results serve as a guide for the design of custom flows for lattice simulations with dynamical fermions in other applications.

## 6.6 Unfreezing the Schwinger model

Following the above algorithmic developments, in this section flow-based sampling is used to solve topological freezing in a fermionic gauge theory at criticality. Specifically, a numerical demonstration in the Schwinger model at the critical value of the fermion mass is provided, illustrating that the flow-based approach is robust at sample sizes where HMC fails.

### 6.6.1 Flow architecture

As discussed above, in order to achieve efficient sampling via a flow-based approach, it is critical to incorporate the symmetry properties of the target distribution. For the Schwinger model specifically, gauge invariance imposes strong constraints. These are built into the model using the framework of gauge-equivariant flows on compact manifolds developed in [174, 175, 265]. The other challenge of course is the sampling of theories with fermionic degrees of freedom. Out of the four treatments in Section 6.2, here we simply consider a marginal sampler using an exact evaluation of the fermion determinant, which poses no computational problem at the scale of the present study. This means that the model describes only gauge degrees of freedom, and the effective action defined in Equation (2.41) is computed exactly during training and for MCMC sampling. However, it should be emphasized that there are no conceptual barriers to employing any of the more scalable approaches based on the pseudofermion formulation.

Following [174] and the in-depth discussion of equivariance in Section 6.3.2, gauge-equivariant flows are constructed by composing a sequence of equivariant coupling layers. In each layer, gauge-invariant closed Wilson loops are computed from the frozen gauge links and used as inputs for the context functions. The outputs of these functions are used to parametrize the transformation of the active gauge links, which is constrained to commute with gauge transformations. Specifically, each gauge-equivariant coupling layer updates the active subset of the links,

$$\begin{aligned} M^k_{\mu\nu} = \{&U_\mu\big((4n+k)\hat{\mu} + 2m\hat{\nu}\big)\,\big|\,\forall\, n,m \in \mathbb{Z}\} \\ &\cup \{U_\mu\big((4n+2+k)\hat{\mu} + (2m+1)\hat{\nu}\big)\,\big|\,\forall\, n,m \in \mathbb{Z}\}\ , \end{aligned} \tag{6.46}$$

where $k, \mu, \nu$ change for each layer. The flow is constructed by iterating through $k \in \{0,1,2,3\}$ in order, first with $\mu = 0, \nu = 1$, then for $\mu = 1, \nu = 0$. In this way, all links are updated within 8 layers. Here, a model with 48 layers is constructed, so each link is updated a total of 6 times. For active loops, the plaquettes that project forwards from their corresponding active links ("active plaquettes") are used. Combined with a gauge-invariant base distribution, this yields an overall gauge-invariant model. The base distribution $r(U)$ for the flow model is an independent uniform distribution over the $U(1)$ Haar measure on each link. Specifically, random variables $A$ are sampled uniformly over $[0, 2\pi)$ and then used to construct gauge links via $U = e^{iA}$.

Unlike in the limit of pure-gauge theory where the hopping parameter $\kappa$ of the Wilson fermion formulation is taken to zero—corresponding to an infinite fermion mass—the Schwinger model at finite mass exhibits long-range correlations, with the correlation length defined by the inverse of the mass of the lightest particle. This demands new architectural features over those employed for the pure-gauge models considered in [174]. First, a subset of active links is used that is locally more sparse, with each active link completely surrounded by frozen ones. This allows for a better propagation of information over longer distances. Second, larger $2 \times 1$ Wilson loops along with $1 \times 1$ plaquettes are provided as inputs for the context functions. Third, the architecture considered here includes dilated convolutions, which retain translational equivariance, but feature better context aggregation, i.e. an exponential expansion of the receptive field without loss of resolution or coverage [266]. Fourth, we parametrize our transformations using highly expressive neural splines [267].

Specifically, each layer has its own convolutional neural network that outputs the parameters defining the transformation of active plaquettes. These neural networks each take six input channels corresponding to

$$\cos\theta_P,\ \sin\theta_P,\ \cos\theta_{2\times1},\ \sin\theta_{2\times1},\ \cos\theta_{1\times2},\ \sin\theta_{1\times2}\ , \tag{6.47}$$

where $\theta_P$ is the plaquette angle defined in Equation (2.44), and $\theta_{2\times1}, \theta_{1\times2}$ are the arguments of the $2 \times 1, 1 \times 2$ Wilson loops, respectively. The sin and cos transformations are applied to ensure that the input is a continuous function of the gauge fields in order to avoid numerical instabilities at the boundary. Each neural network is built from three convolutions with kernel size 3 and dilation factors $1, 2, 3$ in order (where 1 is a standard undilated convolution). Between each intermediate convolution, there are 64 hidden channels, and the final output has 10 channels. After each intermediate convolution we use LeakyReLU activations, but no activation is applied to the final output. The 10 output channels are used to parametrize the positions and slopes of the 3 knots of a circular rational quadratic spline $s(\theta_P)$, as well as an overall offset $t$. These are used to transform the active plaquettes $\theta_{PA}$ as $\theta'_{PA} = s(\theta_{PA}) + t$. The active links are updated by inferring a link transformation that induces a transformation of the active loops.

We again consider the aforementioned self-training scheme where the loss function is a stochastic estimate of the Kullback-Leibler divergence made with $q$-distributed samples generated by the model,

$$D_{\mathrm{KL}}(q||p) = \int dU\, q(U) \log \frac{q(U)}{p(U)} \approx \big\langle \log q(U) + S(U) \big\rangle_{\phi\sim q(U)} + \log Z\ . \tag{6.48}$$

The performance of flow-based MCMC using the trained model is compared against that of HMC. At finite lattice spacing, a diverging correlation length is realized by tuning $\kappa$ to its critical value, resulting in a vanishing renormalized fermion mass. To achieve this, for a square lattice of extent $L = 16$, one sets $\beta = 2.0$ and $\kappa = 0.276$ [60]. The acceptance rate for sampling from the trained model at these parameters is $\sim 17\%$.

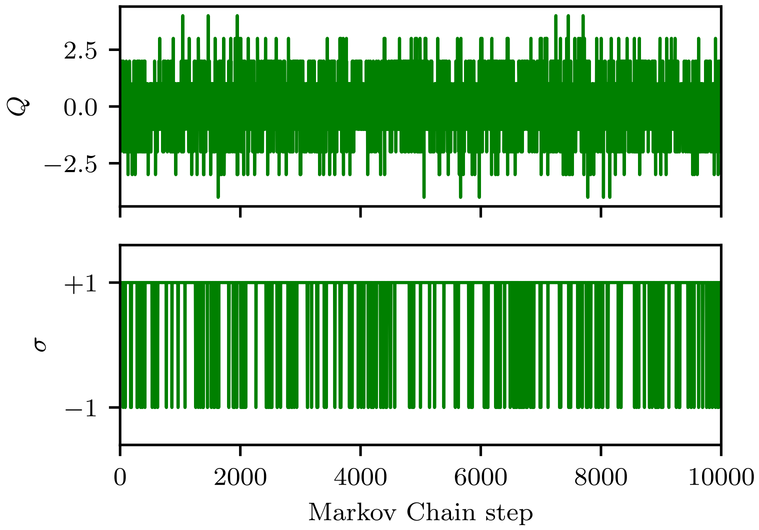


Figure 6.7: Monte Carlo history of the topological charge, $Q$ (top), and the sign of the real part of the determinant of the Dirac operator, $\sigma$ (bottom), computed with augmented HMC.

### 6.6.2 HMC details

The HMC results presented here also use the exact determinant action defined in Equation (2.41), with molecular dynamics forces computed from exact derivatives. Trajectories of length $t_{\mathrm{HMC}} = 1$ divided into 10 steps are used, yielding an acceptance rate of 94%. The HMC data are taken from streams $2 \times 10^5$ trajectories long, with no trajectories discarded between measurements. Each stream is initialized from a hot start, i.e. all links are drawn from independent uniform distributions.

For the Schwinger model, it is possible to implement an augmentation step for HMC that proposes hops to other topological sectors [40–42] by generating configurations $U'$ such that $0 \neq Q(U') - Q(U) \equiv \mathit{\Delta} Q \in \mathbb{Z}$. This is achieved by distributing the proposed change across links according to

$$\begin{aligned} U_0'(x) &= \exp\left(-2\pi\mathrm{i}\frac{\mathit{\Delta} Q}{V} x_1\right) U_0(x) \\ U_1'(x) &= \exp\left(2\pi\mathrm{i}\frac{\mathit{\Delta} Q}{L} x_0 \delta_{x_1, L-1}\right) U_1(x) \ . \end{aligned} \tag{6.49}$$

Here, coordinates are understood in lattice units, i.e. $x_i \in \{0, \ldots, L-1\}$. For simplicity, we restrict $\mathit{\Delta} Q \in \{-2, -1, 0, 1, 2\}$ and propose each $\mathit{\Delta} Q$ with equal probability. The proposal is accepted or rejected with a standard Metropolis step, with an acceptance rate of 34% for the above parameters.

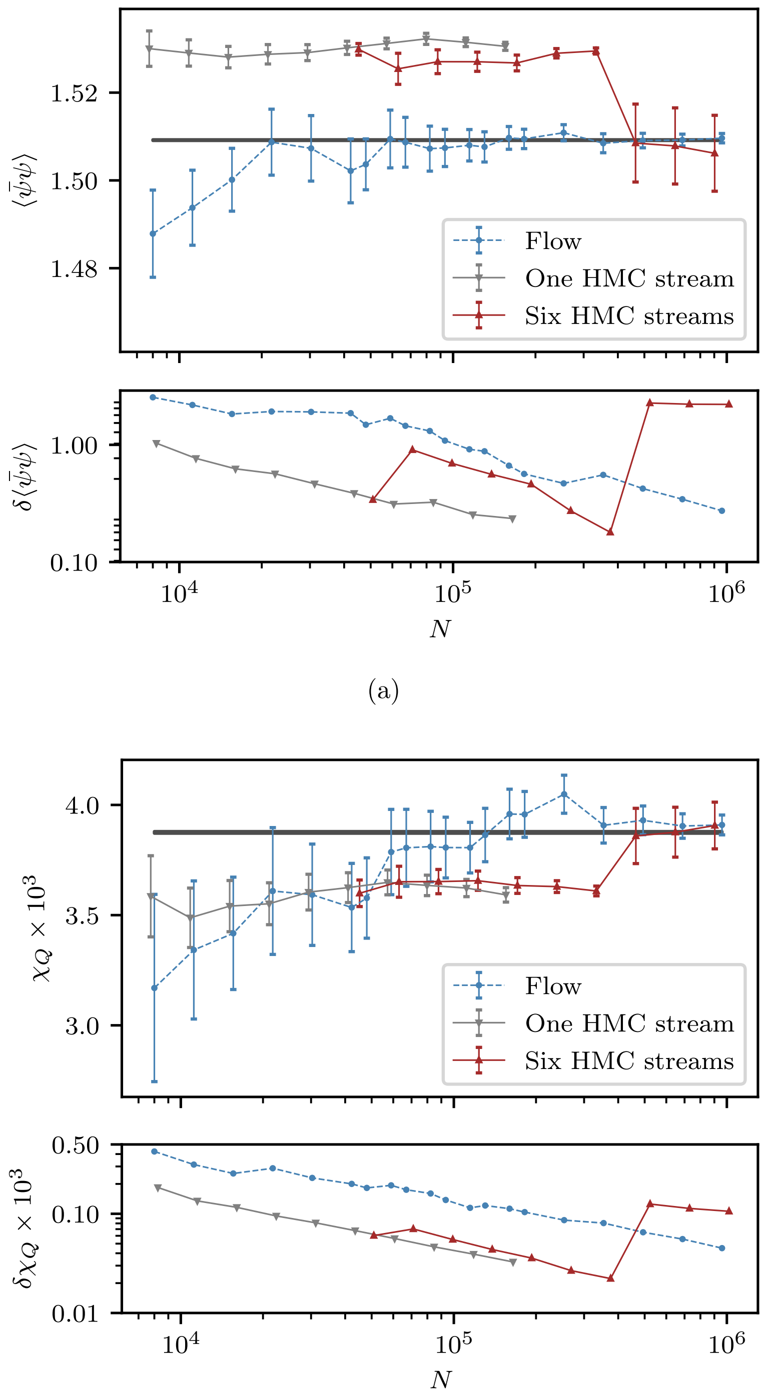
Interleaving this augmentation with HMC steps produces an algorithm with different properties than HMC alone, which we call augmented HMC from now on. No equivalent construction is known for many theories, including QCD, which is part of the motivation to consider flow-based samplers in the first place. This augmented method is used solely to obtain baseline results for the chiral condensate and the topological susceptibility defined in Equations (2.32) and (2.45), respectively. To do so, an ensemble of $1.2 \times 10^7$ configurations is produced using these augmentation hits alternated with HMC steps. The baseline results are estimated to be $\langle\bar{\psi}\psi\rangle = 1.50918(9)$, $\langle\chi_Q\rangle = 0.003875(4)$.

In Figure 6.7, part of the Monte Carlo history of $Q$ and $\sigma$ is shown for the baseline run with augmented HMC. As can be seen, rapid fluctuations occur at a scale comparable to the flow model results discussed below. In particular, $\sigma$ exhibits an asymmetric distribution along positive and negative values, as expected from the higher total weight of even topological sectors.

### 6.6.3 Advantages of flow-based sampling

A clear illustration of the advantages of flow-based sampling for the Schwinger model at criticality is given in Figure 6.8, which compares estimates of the chiral condensate as well as the topological susceptibility from HMC with those from flow-based MCMC and the augmented HMC baseline described above. Uncertainties are quantified using the integrated autocorrelation time with the "gamma method" [268]. Clearly, the single frozen HMC stream yields estimates that are manifestly inconsistent with the baseline result. This strongly indicates severely underestimated uncertainties even at the very large sample size of $N \approx 10^5$. One may hope to remedy this illness by using a dataset of samples from six independent HMC streams, since then information from multiple topological sectors can be incorporated even in the presence of freezing in the individual streams. However, as the figure shows, this estimate is still biased for $N \approx 10^5$ samples, with incorrect uncertainties deceptively scaling as $1/\sqrt{N}$. The estimate becomes consistent with the ground truth only when $N \gtrsim 10^6$. The uncertainty, however, catastrophically increases—a clear indication of an ergodicity problem. This analysis suggests that affordable HMC stream lengths may not be sufficient to diagnose bias. By contrast, flow-based results converge smoothly to the baseline value, with errors scaling as $1/\sqrt{N}$.

Figure 6.9 provides a more direct illustration of freezing in the Monte Carlo histories of topological quantities. The topological sectors of the Schwinger model are distinguished by the integer-valued topological charge defined in Equation (2.43). Due to lattice artifacts, this observable fluctuates even when the topological sector is fixed. A better-suited observable to identify true tunneling events—the sign $\sigma$ of the real part of the fermion determinant factor—was defined in Equation (2.46). In the first HMC stream, $Q$ appears to fluctuate without any evidence of freezing. However, $\sigma$ is completely frozen for all samples shown, implying that these fluctuations arise from discretization effects and do not correspond to tunneling events between topological sectors. In the second HMC stream, we see an abrupt change in the behavior of $Q$. This coincides with a change in $\sigma$, confirming that a true tunneling event has occurred. By contrast, flow-based sampling exhibits rapid fluctuation in both $Q$ and $\sigma$, demonstrating sampling which rapidly mixes topological sectors.



(a)

(b)

Figure 6.8: Demonstration of underestimated uncertainties when using HMC for the chiral condensate (a) and topological susceptibility (b).

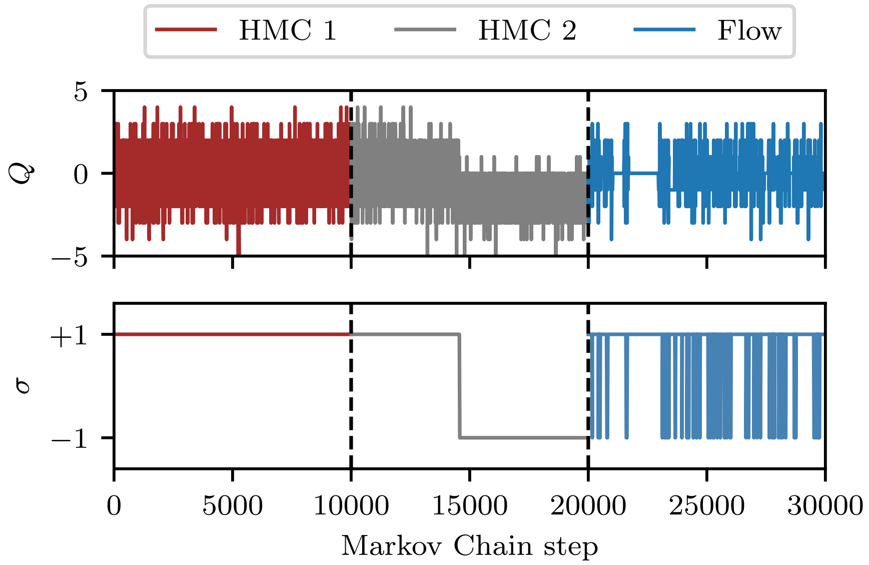


Figure 6.9: Monte Carlo history of the topological charge (top) and the sign of the real part of the determinant of the Dirac operator (bottom).

A fair and comprehensive comparison of the costs of HMC and flow-based MCMC requires quantifying three factors for each: setup costs, the raw computational cost of a sampling step, and the sampling efficiency (i.e. the degree of autocorrelation). Setup costs—predominantly, equilibration for HMC and training for flows—are particularly difficult to compare in this case. On the one hand, fully equilibrating HMC requires observing and discarding many tunneling events, which occur stochastically. On the other hand, training costs for the flow-based approach may vary over many orders of magnitude depending on the training scheme. Raw computational costs can be measured directly, but depend strongly on implementation details. With that being said, flow-based MCMC steps are found to be $\sim 10$ times less expensive than HMC trajectories on the same hardware, due to the frequently required inversions of the Dirac operator in HMC. However, there is room for optimization in both cases, and the results should be taken with a grain of salt.

Nevertheless, an approximate comparison of sampling efficiency is sufficient to show the advantage of flow-based sampling over HMC. Each algorithm exhibits some characteristic time between tunneling events and a chain with many times that number of steps is required in order to incorporate information from all topological sectors. For HMC, tunneling events are observed to be separated by $\sim 20$k trajectories on average. In contrast, the sector changes much more rapidly with flow-based sampling, namely every $\sim 6$ steps. Hence, for this model, the advantage in sampling efficiency of flow-based MCMC over HMC is estimated to be more than three orders of magnitude.

## 6.7 Applicability to update-based approaches

While the sampling schemes presented in this chapter are based on the proposal of statistically independent field configurations (except for the Gibbs sampler; see Section 6.5.3 for further discussion), the flow-based models defined here may also be used in methods that instead propose configuration updates, rather than completely new samples. Importantly, the described flow models may serve as an "engine" for a much broader class of sampling algorithms. A simple way to construct an update-based algorithm with these models would be to formulate stochastic processes in the flow prior that guarantee asymptotic exactness under the target distribution, such as partial heatbath resampling, HMC, or Langevin-type algorithms, rather than independently drawing a completely new prior sample in every update step. Such partial updates have previously been studied in the context of other generative models [254, 269, 270] as well as trivializing map approaches [271, 272]. Moreover, flow-based updates may be interleaved with steps of HMC [170] or other MCMC methods [273]. Such composite algorithms may provide improved sampling over either method alone. Other possible improvements include using the flow models developed here inside a hierarchical multilevel MCMC scheme, as proposed by [44], or as components of stochastic normalizing flows [173, 274, 275].

In contrast to these update-based methods where autocorrelations are always induced by construction—similar to the diffusion-based algorithms we initially set out to replace—direct sampling approaches have the advantage that autocorrelations in the flow-based Markov chain are in principle eliminated for an ideal model. Imperfect models, however, can still result in residual correlations caused by rejections in the Metropolis step. Whether these residual correlations from an imperfect model can outweigh the autocorrelations in corresponding update-based methods is an open question. Since this may depend strongly on the model details and the specific problem under consideration, it is likely impossible to give a general answer here. Instead, which combination of sampling scheme and model architecture yields the best performance must be determined empirically on a case-by-case basis.

Apart from devising modified sampling schemes for the types of flows presented in this work, one may also consider defining flows that directly transform configurations in order to produce proposals for Markov chain updates. Related work on learning improved HMC-like updates includes A-NICE-MC [276], its recent application to the lattice simulation of scalar $\phi^4$-theory [251], L2HMC [277], and DLHMC [176, 178], which was demonstrated to successfully mitigate topological freezing in the context of $U(1)$ lattice gauge theory in two dimensions. These approaches require the implementation of flows suitable for transforming the primary fields and conjugate momenta conditioned on each other. The flows over pseudofermion variables developed in this work can therefore be used to extend such methods to the setting of lattice field theory involving dynamical fermion fields. These insights may also inform the design of novel building blocks for the self-learning Monte Carlo method mentioned previously, which was recently applied to non-Abelian gauge theory with dynamical fermions [195].

## 6.8 Summary and outlook

In this chapter, four asymptotically exact approaches to generative neural sampling for fermionic lattice field theories have been introduced, based on different decompositions of the joint action over bosonic and pseudofermionic fields. Furthermore, several techniques were developed to model the associated distributions via the construction of flow-based models. All sampling methods have been demonstrated to successfully produce asymptotically exact samplers in a proof-of-principle application to a two-dimensional Yukawa theory. Nevertheless, the discussed architectures represent merely a selection from a large class of possible ways to model the aforementioned distributions. Their observed relative performance provides a starting point for understanding the distinctions between different sampling schemes and architectures, but should not be considered a definitive indicator of their performance in the context of other theories or at larger scales.

Importantly, investigating the continuum limit of flow-based samplers is relevant to determine their potential to mitigate critical slowing down at scale. This question arises with or without fermions, and empirical studies are required to understand the scaling of these methods for different theories. Nonetheless, the "building blocks" of flows suitable for fields including pseudofermions, and the sampling strategies outlined in this work, provide a basis for developing efficient flow-based samplers for fermionic theories. In the spirit of this endeavor, an architecture was developed that can successfully model long-range correlations in the Schwinger model at vanishing renormalized fermion mass. The resulting algorithm does not suffer from topological freezing and thus outperforms HMC by orders of magnitude. These results represent an important milestone in first-principles calculations for gauge field theories coupled to fermions using provably exact machine learning.

Nevertheless, challenges remain on the road to large-scale applications, such as state-of-the-art QCD calculations. The sampling approach for the Schwinger model discussed here relied on an exact evaluation of the fermion determinant. For larger volumes and theories in higher dimensions, employing one of the more scalable algorithms based on the pseudofermion method will become necessary. Continued work into improved stochastic approximations of determinants [243, 278–283] complements the flow-based approach presented here and may be combined with the proposed framework to yield further performance improvements.

If the success in the Schwinger model can be extended to QCD calculations at scale, this will have significant impact across nuclear and particle physics. The immediate next steps in this endeavor are the transfer of insights gained from the Yukawa and Schwinger models considered in this work to $SU(3)$ gauge fields coupled to fermions in four dimensions, as well as studying the scalability of the method.

# 7 Flow-based density of states for complex actions

In Chapter 6, it was demonstrated that sampling algorithms based on flows have the potential to solve ergodicity problems in lattice calculations. Recent work regarding the computation of thermodynamic quantities with flows suggests that they are also applicable to the DoS approach to complex action problems mentioned in Section 3.2. The present chapter is dedicated to the development of this idea.

After a brief introduction in Section 7.1, the DoS approach pertinent to the type of complex action problem considered here is reviewed in Section 7.2. The proposed method is explained in Section 7.3 and numerical results are presented in Section 7.4. The contributions are summarized and an outlook is provided in Section 7.5. The contents of this chapter have been published in [7] together with Jan M. Pawlowski.

## 7.1 Introduction

As described in Section 3.2, for many physically interesting theories, the associated Euclidean lattice action is complex-valued, which prohibits the application of standard importance sampling. In this context, it has been shown that with the DoS approach [110, 284–293], certain complex action problems can be successfully treated [111, 294–299]. However, directly computing the DoS is generally not possible due to the intrinsically high variance of the associated observables. Instead, the usual strategy is to measure its derivative via restricted MCMC calculations followed by numerical reconstruction. The high precision required to control the accumulation of errors from the approximation of the integral can be computationally expensive.

Recently, it has been noted that similar thermodynamic quantities in lattice field theory can be computed directly using generative neural sampling [185, 250, 300], thereby completely avoiding the aforementioned numerical reconstruction of the quantity of interest. Hence, flow-based sampling may also be applied to the direct computation of the DoS for lattice field theories with complex actions. This approach is developed here in the context of scalar $\phi^4$-theory with an imaginary external field. First, the exactly solvable, zero-dimensional case is investigated as a toy model for a proof-of-principle demonstration, showing that the DoS as well as the partition function and magnetization as functions of the external field are computed correctly. In particular, the Lee-Yang zeroes [301] of the partition function together with the associated discontinuities in the magnetization can be successfully located. The approach is then applied to actual lattice models in one and two dimensions, accurately reproducing the densities obtained with conventional MCMC methods.

## 7.2 Density of states

We consider lattice field theories with complex-valued actions where the imaginary part is generated by a constant, homogeneous external field, i.e.

$$S(\phi) = S_r(\phi) + \mathrm{i}hX(\phi) \ , \tag{7.1}$$

where $S_r, X, h \in \mathbb{R}$. The partition function and expectation values of observables are defined as

$$Z = \int \mathcal{D}\phi \, e^{-S_r(\phi) - \mathrm{i}hX(\phi)} \ , \tag{7.2}$$

$$\langle \mathcal{O} \rangle = \frac{1}{Z} \int \mathcal{D}\phi \, e^{-S_r(\phi) - \mathrm{i}hX(\phi)} \mathcal{O}(\phi) \ . \tag{7.3}$$

Since the action is complex, standard importance sampling is not directly applicable and reweighting often becomes prohibitively expensive when increasing $h$ due to the average phase factor being close to zero, as discussed in Section 3.2.

One ansatz to make the computation more tractable is to consider the DoS as a function of the quantity that generates the imaginary part of the action, i.e.

$$\rho(c) = \int \mathcal{D}\phi \, e^{-S_r(\phi)} \delta(X(\phi) - c) \ . \tag{7.4}$$

Essentially, $\rho(c)$ corresponds to slices of the partition function for the real part of the action, with the configuration space restricted to hypersurfaces of constant $X(\phi) = c$. In MCMC calculations, this restriction can be achieved e.g. by confining the dynamics through additional rejections, or by replacing the $\delta$-distribution with a Gaussian of finite width, which is the approach used in the present work; see Section 7.3 for details.

If $\rho(c)$ is known, the partition function for the full action as well as expectation values of observables (that are functions of $c$ only) can be computed in terms of one-dimensional integrals with a residual phase,

$$Z = \int \mathrm{d}c \, \rho(c) \, e^{-\mathrm{i}hc} \ , \tag{7.5}$$

$$\langle \mathcal{O} \rangle = \frac{1}{Z} \int \mathrm{d}c \, \rho(c) \, e^{-\mathrm{i}hc} \, \mathcal{O}(c) \ . \tag{7.6}$$

However, similar to partition functions themselves and thermodynamic quantities in general, direct computation of $\rho(c)$ is often infeasible with conventional MCMC algorithms due to the high variance associated with the required observables. Instead, it is usually reconstructed from measurements of $\partial_c \log \rho(c)$, as detailed below.

## 7.3 Flow-based density of states

As already mentioned in Section 7.2, we consider a formulation of the DoS approach where the $\delta$-distribution in Equation (7.4) is replaced by a Gaussian of finite width, following e.g. [290, 293]. This enables the straightforward application of both standard sampling algorithms like Hybrid/Hamiltonian Monte Carlo (HMC) as well as the flow-based approach. Exactness of all expressions can be retained at the cost of a residual sign problem (which is tractable for sufficiently small width) or by extrapolating to the limit of vanishing width.

First, we note that the result of the Gaussian integral

$$\int \mathrm{d}c\, e^{-\frac{P}{2}(c-a)^2} = \sqrt{\frac{2\pi}{P}} \equiv \mathcal{N} \tag{7.7}$$

is independent of $a$. Hence, we can rewrite Equation (7.2) as

$$Z = \int \mathcal{D}\phi \int \mathrm{d}c\, e^{-\frac{P}{2}(c-X(\phi))^2 - \log\mathcal{N}} e^{-S_r(\phi) - \mathrm{i}hX(\phi)} \ . \tag{7.8}$$

We then define the $P$-dependent DoS as

$$\rho_P(c) = \int \mathcal{D}\phi\, e^{-S_{c,P}(\phi)} \ , \tag{7.9}$$

where

$$S_{c,P}(\phi) = S_r(\phi) + \frac{P}{2}(c - X(\phi))^2 + \log\mathcal{N} \ . \tag{7.10}$$

The “true” DoS as defined in Equation (7.4) is recovered in the limit $P \longrightarrow \infty$.

Using $\rho_P$, the partition function can be expressed as

$$\begin{aligned} Z &= \int \mathrm{d}c \int \mathcal{D}\phi\, e^{-S_{c,P}(\phi)} e^{-\mathrm{i}hX(\phi)} \\ &= \int \mathrm{d}c\, \rho_P(c) \frac{\int \mathcal{D}\phi\, e^{-S_{c,P}(\phi)} e^{-\mathrm{i}hX(\phi)}}{\int \mathcal{D}\phi\, e^{-S_{c,P}(\phi)}} \\ &= \int \mathrm{d}c\, \rho_P(c) \big\langle e^{-\mathrm{i}hX(\phi)} \big\rangle_{\phi\sim e^{-S_{c,P}(\phi)}} \ . \end{aligned} \tag{7.11}$$

Hence, in this formulation, the partition function is still a one-dimensional integral over the $P$-dependent DoS, but with an additional average phase factor computed on ensembles sampled with $S_{c,P}(\phi)$. The fluctuations of this phase factor are tractable as long as the parameter $P$ is large enough, such that $X(\phi)$ does not deviate too strongly from $c$. Accordingly, expectation values of observables can be written as

$$\langle \mathcal{O} \rangle = \frac{\int \mathrm{d}c\, \rho_P(c) \big\langle e^{-\mathrm{i}hX(\phi)} \mathcal{O}(\phi) \big\rangle_{\phi\sim e^{-S_{c,P}(\phi)}}}{\int \mathrm{d}c\, \rho_P(c) \big\langle e^{-\mathrm{i}hX(\phi)} \big\rangle_{\phi\sim e^{-S_{c,P}(\phi)}}} \ . \tag{7.12}$$

As mentioned previously, a direct computation of $\rho_P(c)$ with traditional MCMC methods is often infeasible for problems of interest. Instead, the usual strategy is to compute

$$\begin{aligned}\frac{\partial \log \rho_P(c)}{\partial c} &= \frac{1}{\rho_P(c)} \frac{\partial \rho_P(c)}{\partial c} \\ &= \frac{\int \mathcal{D}\phi \, e^{-S_{c,P}(\phi)}(-P(c - X(\phi)))}{\int \mathcal{D}\phi \, e^{-S_{c,P}(\phi)}} \\ &= \big\langle -P(c - X(\phi))\big\rangle_{\phi \sim e^{-S_{c,P}(\phi)}} \ ,\end{aligned} \tag{7.13}$$

and then to reconstruct $\log(\rho_P(c)/\rho_P(0))$ by numerical integration, e.g. with the trapezoidal rule. In contrast, normalizing flows trained with $S_{c,P}(\phi)$ as the target action allow for a direct computation of $\rho_P(c)$ (including the overall factor $\rho_P(0)$) using configurations sampled from $q(\phi)$, as long as the overlap of the target and model distributions is sufficient. This can be seen by rewriting Equation (7.9) as

$$\begin{aligned}\rho_P(c) &= \int \mathcal{D}\phi \, q(\phi) \frac{e^{-S_{c,P}(\phi)}}{q(\phi)} \\ &= \left\langle e^{-S_{c,P}(\phi) - \log q(\phi)} \right\rangle_{\phi \sim q(\phi)} \ ,\end{aligned} \tag{7.14}$$

similar to Equation (4.14). A successfully trained flow minimizes the fluctuations of the exponent in the last expression, such that the variance of the expectation value remains tractable. This is precisely the crucial advantage of flow-based sampling over conventional MCMC methods that allows the computation of thermodynamic quantities via variationally optimized reweighting [250, 300].

In order to compute $\rho_P$ across a wide range, one could train independent flows for each value of $c$. Alternatively, a more efficient approach would be to start by training one flow at some given point (e.g. $c = 0$) and then perform retraining for each additional point. However, since high precision in $c$ is desired, these strategies seem impractical. Apart from such a training procedure already being computationally expensive, a large number of different parameter sets for all the individual flow transformations would then have to be stored and loaded into memory for evaluation. Instead, the full information about $\rho_P$ for all $c$ can be encoded in a single flow model. This is achieved by promoting the transport map $f(\xi)$ to a conditional transformation $f_c(\xi)$, which additionally depends on $c$. In particular, the context functions $s, t$ of all affine couplings as defined in Equation (4.15) are modified to take $c$ as an additional input. This only marginally increases the computational effort of evaluating the transformation, although it may be necessary to make the flow more expressive overall in order to properly model the dependence on $c$.

Furthermore, an additional $c$-dependent offset is introduced at the last layer, such that the conditional generation of field configurations $\phi$ from prior samples $\xi$ takes the form

$$\phi(\xi|c) = f_c(\xi) + \bar{\phi}(c) \ , \tag{7.15}$$

with $\bar{\phi}(c)$ chosen such that $X(\bar{\phi}(c)) = c$. This offset already provides the correct mean field configuration for each $c$ and thereby greatly simplifies training from the start, because the flow only has to model the distribution around the given $\bar{\phi}(c)$. Since this amounts to just a constant shift, the Jacobian of the transformation remains unchanged.

Taken together, the full transformation defined in Equation (7.15) consisting of a conditional transport map and an additional offset induces a conditional model distribution $q_c$ for each $c$, such that the $P$-dependent DoS may finally be computed as

$$\rho_P(c) = \left\langle e^{-S_{c,P}(\phi) - \log q_c(\phi)} \right\rangle_{\phi \sim q_c(\phi)} . \tag{7.16}$$

For training, similar to the optimization strategy in Chapter 6, one may estimate gradients of $D_{\mathrm{KL}}(q_c || e^{-S_{c,P}})$ (i.e. the conditional Kullback-Leibler divergence) at randomly sampled points $c$, distributed uniformly across a sufficiently large interval in order to enforce optimal generalization for arbitrary $c$.

At this point it should be emphasized again that in order to evaluate the above expression for $\rho_P$, only samples from the model are required. Importantly, this implies that once the flow has been trained, the remaining computations can be performed extremely efficiently by embarrassingly parallel sampling of the model distribution. In particular, field configurations do not need to be arranged in a Markov chain and no Mteropolis accept/reject steps are necessary. This constitutes a further potential advantage of the proposed approach over conventional MCMC calculations.

## 7.4 Results

### 7.4.1 Zero-dimensional model

For a first demonstration of the flow-based approach, we consider a zero-dimensional model of a single two-component scalar field with quartic self-interaction in an imaginary external field. The simplicity of this model facilitates a comparison to exact results. The action is similar to the one for the single-component scalar field theory considered in Section 2.2, but with two components and an additional linear term:

$$S(\phi) = \frac{m^2}{2}\left(\phi_1^2 + \phi_2^2\right) + \frac{\lambda}{4}\left(\phi_1^2 + \phi_2^2\right)^2 + \mathrm{i} h \phi_1 , \tag{7.17}$$

where $\phi_1, \phi_2, m^2, \lambda, h \in \mathbb{R}$. Note that for the zero-dimensional model, there is no kinetic term. We can identify $X(\phi) \equiv \phi_1$ and

$$\begin{aligned} S_{c,P}(\phi) \equiv \frac{m^2}{2}\left(\phi_1^2 + \phi_2^2\right) + \frac{\lambda}{4}\left(\phi_1^2 + \phi_2^2\right)^2 \\ + \frac{P}{2}\left(c - \phi_1\right)^2 + \log \mathcal{N} . \end{aligned} \tag{7.18}$$

A normalizing flow composed of 16 affine coupling layers is trained using this

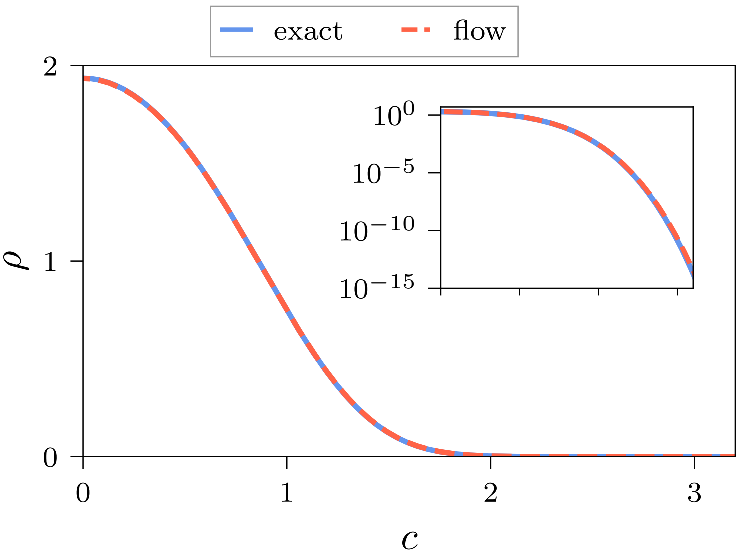
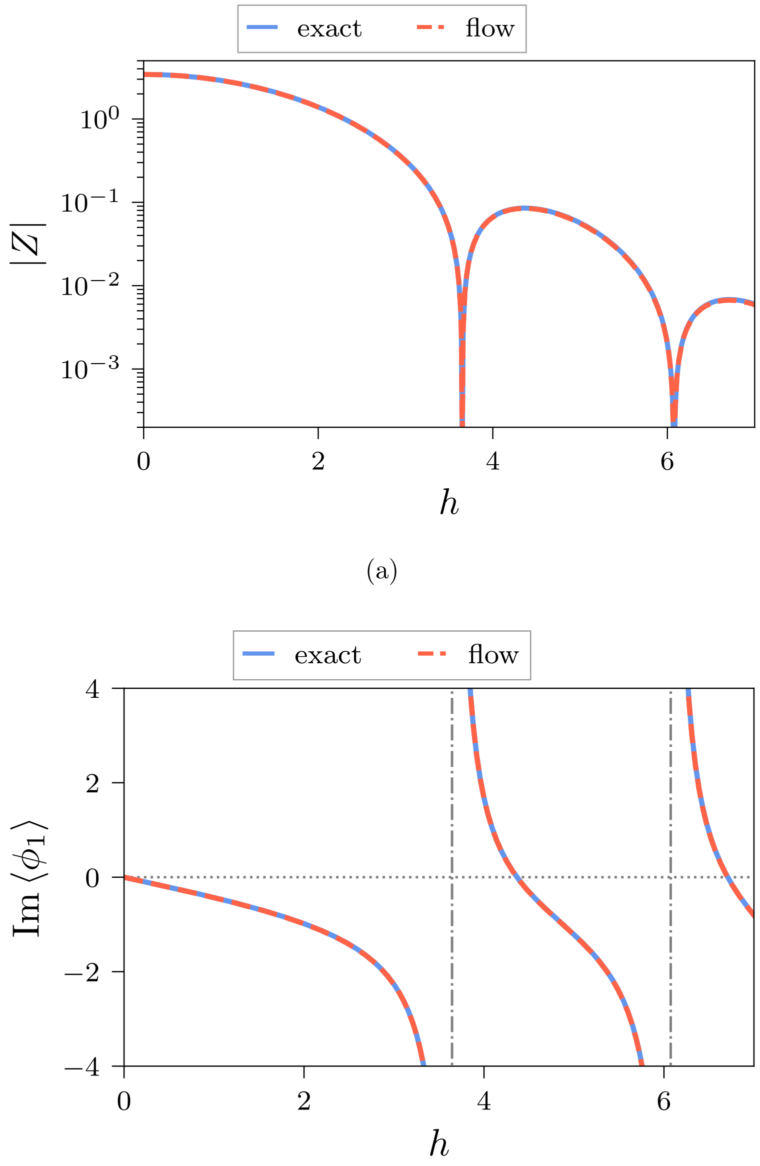


Figure 7.1: Comparison of the DoS computed with flow-based sampling to the exact solution for the zero-dimensional model.

target action with $m^2 = 1, \lambda = 1, P = 1000$.[9] The context functions are implemented as fully-connected networks featuring three hidden layers with 64 neurons each. As activation functions between layers, LeakyReLU is used again, together with a Tanh activation after the final layer. Each network has two input neurons, one for the frozen variable (either $\phi_1$ or $\phi_2$ depending on the layer) and the other one for the condition $c$, as well as two output neurons providing the values for $s, t$ in Equation (4.15). The offset in this case is simply $\bar{\phi}(c) = (c, 0)$ such that $X(\bar{\phi}(c)) \equiv \bar{\phi}_1(c) = c$, as required by the construction of Section 7.3. For the training, the Adam optimizer is applied with a learning rate of $10^{-3}$ and a batch size of 10k, with a total of 5k gradient updates. In order to compute $\rho(c)$, 10k samples are drawn for each value of $c$, with a spacing of $\Delta c = 0.01$.

The DoS computed with flow-based sampling is compared against the exact result in Figure 7.1, conclusively demonstrating the correctness of the flow-based approach across several orders of magnitude. Furthermore, the partition function $Z(h)$ is accurately reproduced, as shown in Figure 7.2a. In particular, the locations of the first two Lee-Yang zeroes can be clearly identified. They are associated with discontinuities in the average imaginary part of $\phi_1$, which is also accurately determined with the proposed method as shown in Figure 7.2b.

[9] For the purpose of this proof-of-principle study, it is simply assumed here that $\rho_P \approx \rho$ for sufficiently large $P$ and use Equations (7.5) and (7.6) instead of Equations (7.11) and (7.12). While the accuracy in reproducing the exact results in this case completely justifies this assumption, it should be emphasized that this is an approximation and one needs to generally extrapolate $P \longrightarrow \infty$ more carefully.



(a)

(b)

Figure 7.2: Comparison of the flow results to the exact solution for the zero-dimensional model: partition function (a) and average imaginary part of $\phi_1$ (b) as functions of $h$. The locations of the Lee-Yang zeroes in (a) are associated with the discontinuities of the observable in (b).

### 7.4.2 One- and two-dimensional models

In order to verify that the flow-based approach also works in a less trivial setting, we consider actual lattice models of the two-component scalar field theory described above. The associated action in $d$ dimensions is defined as

$$\begin{aligned} S(\phi) = \sum_{n\in\Lambda} \Bigg( &\frac{1}{2}\sum_{\mu=1}^{d} |\phi(n) - \phi(n+\hat{\mu})|^2 \\ &+ \frac{m^2}{2}\,|\phi(n)|^2 + \frac{\lambda}{4}|\phi(n)|^4 + \mathrm{i}h\phi_1(n) \Bigg) \,, \end{aligned} \tag{7.19}$$

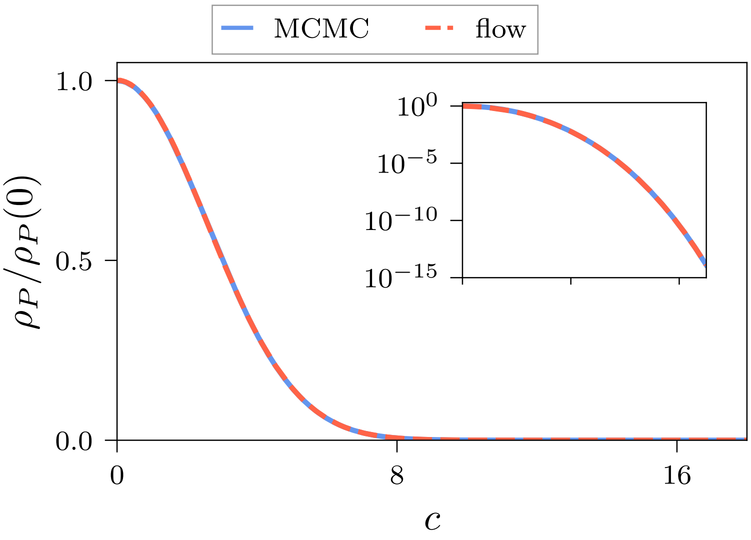
where $\phi(n) = \big(\phi_1(n), \phi_2(n)\big)$, $\Lambda$ is the set of all lattice sites, $\hat{\mu}$ denotes a unit vector, and we assume periodic boundary conditions. The action for each individual site is essentially equivalent to the zero-dimensional model, differing only in the additional kinetic term. Accordingly, we can identify

$$X(\phi) \equiv \sum_{n\in\Lambda} \phi_1(n) \tag{7.20}$$

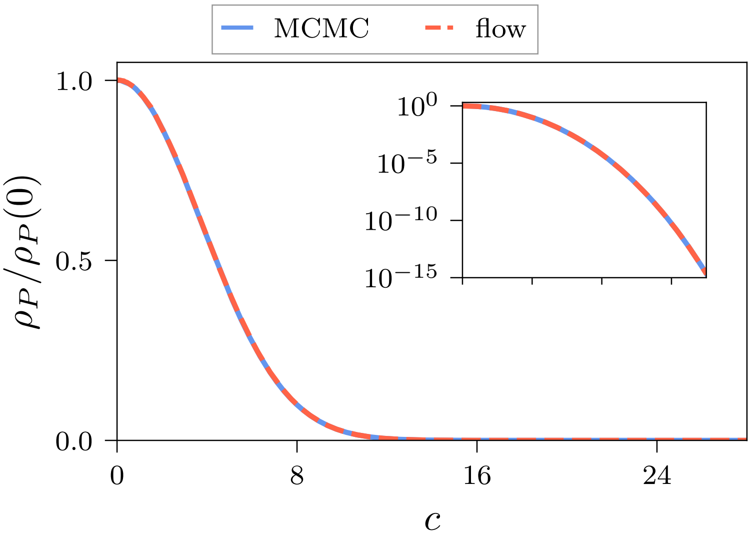
as well as

$$\begin{aligned} S_{c,P}(\phi) \equiv \sum_{n\in\Lambda} \Bigg( &\frac{1}{2}\sum_{\mu=1}^{d} |\phi(n) - \phi(n+\hat{\mu})|^2 \\ &+ \frac{m^2}{2}\,|\phi(n)|^2 + \frac{\lambda}{4}|\phi(n)|^4 \Bigg) \\ &+ \frac{P}{2}\Big(c - \sum_{n\in\Lambda}\phi_1(n)\Big)^2 + \log\mathcal{N} \,. \end{aligned} \tag{7.21}$$

Flows are trained using this target action with $m^2 = 1, \lambda = 1, P = 1000$ for one- and two-dimensional lattices of size 8 and $4 \times 4$, respectively. The flows also consist of 16 affine coupling layers as in the zero-dimensional case. In order to enforce equivariance under translations, the context functions are implemented as convolutional networks featuring two hidden layers with eight channels each, with intermediate LeakyReLU activations and a final Tanh activation as well. Each network has three input channels, two for the frozen subsets of $\phi_1, \phi_2$ (determined by alternating checkerboard masking) and one for the condition $c$; as well as two output channels providing the values for $s, t$ in Equation (4.15). The conditional input is constructed with the same dimensions as a single component of $\phi$ with $c$ evenly distributed across all sites, i.e. with values of $c/|\Lambda|$ on each site where $|\Lambda|$ is the total number of lattice points. For the training, the Adam optimizer is applied again with a learning rate of $10^{-3}$ and a batch size of 1k, using a total of 50k gradient updates. In order to compute $\rho_P(c)$, $10^7$ samples are drawn for each value of $c$, again with a spacing of $\mathit{\Delta} c = 0.01$. The offset in this case is $\bar{\phi}(c) = (c/|\Lambda|, 0)$, where $|\Lambda|$ denotes the total number of lattice sites, such that $X(\bar{\phi}(c)) = c$.



(a)



(b)

Figure 7.3: Comparison of the flow results for the normalized DoS to the reconstructions from MCMC calculations for the one- (a) and two-dimensional (b) models with $P = 1000$.

In order to provide conventional baseline results, HMC streams are generated with the same target action and value for $P$, using a step size of 0.02 and 50 steps per trajectory for the one-dimensional as well as a step size of 0.01 and 100 steps for the two-dimensional case. This results in acceptance rates of roughly 60–90%, with the highest values generally observed around $c = 0$ and decreasing rates for larger $c$. For each $c$, 10k Markov chains are evaluated in parallel, where in each chain the first 1k steps are discarded for equilibration. Subsequently, the chains are evaluated for 100k steps and every 10th configuration is recorded, resulting in a total of $10^8$ configurations for each value of $c$, using a spacing of $\Delta c = 0.01$ as well. As described in the Section 7.3, $\rho_P$ is reconstructed from $\partial_c \log \rho_P$ using the trapezoidal rule and exponentiating the resulting values for $\log \rho_P$.

Figure 7.3 compares $\rho_P(c)/\rho_P(0)$ (i.e. normalized to 1 at $c = 0$) obtained with flow-based sampling to the MCMC baseline. Similar to the zero-dimensional case, the results accurately reproduce the conventional computation, thereby confirming that the proposed approach also works here as intended. It should be emphasized again that for the MCMC baseline, the reconstruction of the DoS at some point $c \neq 0$ by numerical integration necessarily requires precise knowledge of $\partial_c \log \rho_P(c)$ in the interval $(0, c)$. In contrast, with the flow-based approach, the DoS can be independently probed at arbitrary points because it is computed directly.

## 7.5 Summary and outlook

In this chapter, flow-based sampling has been applied to the DoS approach to complex action problems. Specifically, it was shown that flows can be used to compute the DoS directly, thereby disposing of the need to reconstruct it from measurements of a derivative quantity through MCMC calculations. The method was demonstrated in the context of simple models with imaginary external fields, confirming the correctness and accuracy of the proposed approach.

Due to the conceptual and practical differences between the flow-based and conventional strategies, an in-depth comparison of the computational cost is not straightforward and beyond the scope of this proof-of-principle study. However, it should be noted that reaching the same level of accuracy in the final result using the flow-based approach was significantly cheaper in practice on the same hardware. This is likely due to the embarrassingly parallel sampling instead of the sequential evaluation of Markov chains. Furthermore, because of the numerical integration, the conventional ansatz may generally require higher precision in order to achieve an accurate reconstruction, whereas with the flow-based method the DoS can be directly probed at arbitrary points. However, it is unclear a priori how the upfront cost of training the flow compares against thermalizing the Markov chains. Nevertheless, independently of how the cost actually scales in practice, the intrinsic advantages of the flow-based approach described in this work motivate further exploration in this direction.

In the future, the present approach should be extended to higher dimensions, larger volumes, and fields with more components. This may be informative for the study of an approximate model of QCD near the second order phase transition where the external field plays the role of the quark mass [302]. In particular, computing the DoS could help to constrain the location of the Lee-Yang edge singularity. In this context, it may also be worthwhile to implement equivariance of the flow under the residual $O(N-1)$ symmetry in order to better match the symmetries of the target distribution. Further interesting avenues include the relativistic Bose gas at finite chemical potential [303–305] as well as the application to gauge theories via gauge-equivariant flows [174, 175], such as e.g. $U(1)$ gauge theory with a topological term [296] or QCD in the heavy-dense limit [297].

Apart from the aim to solve complex action problems, the DoS method can of course also be employed to compute observables for theories with purely real actions. This may be useful in the treatment of ergodicity problems, since the target distribution can be mapped out explicitly in regions that are pathologically under-sampled with standard MCMC. Hence, the approach presented in this work also constitutes a promising ansatz for circumventing critical slowing down via flow-based methods, complementary to the more commonly investigated independence sampling strategy of Chapter 6.

# 8 Spectral reconstruction with GPR

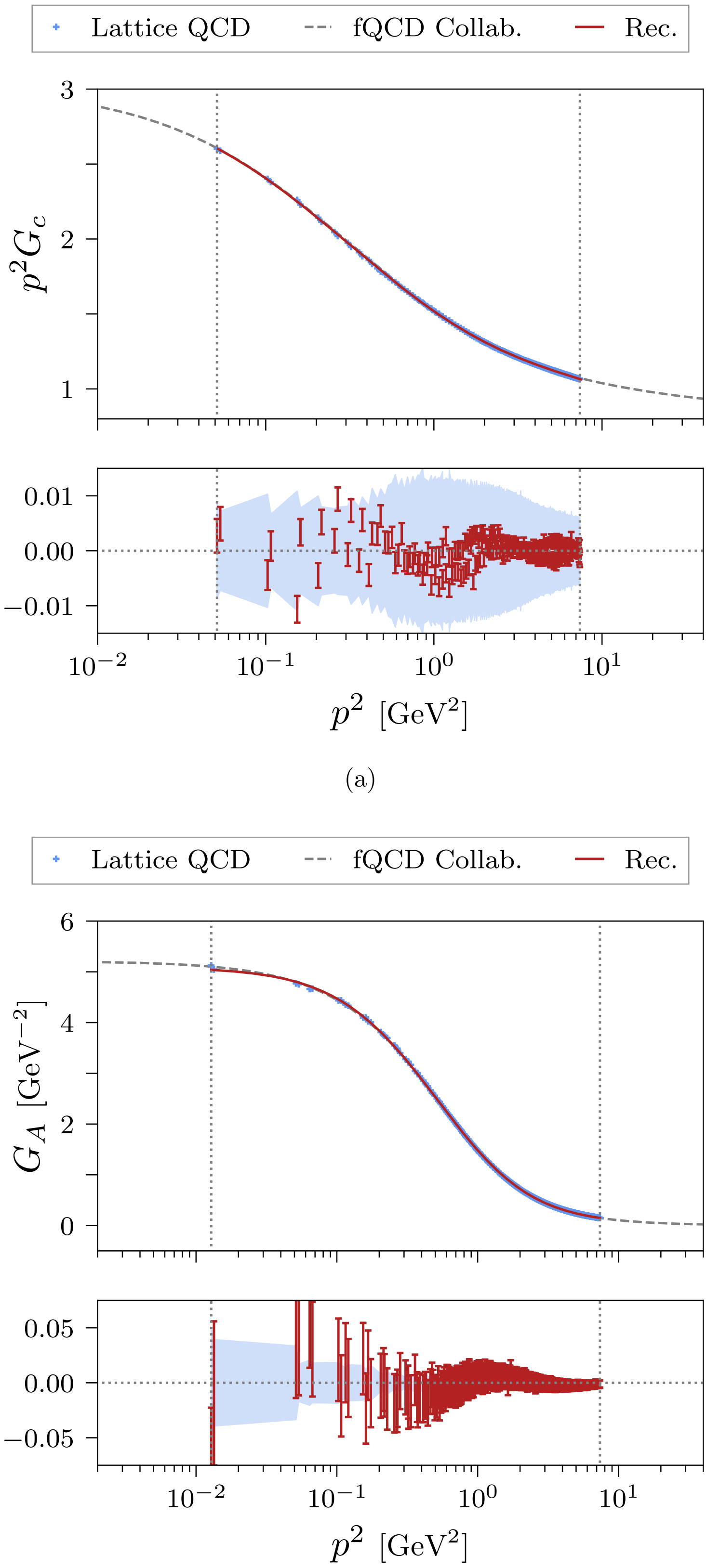
We now turn our attention to the problem of extracting real-time physics from imaginary-time data outlined in Section 3.3, using the GPR method introduced in Section 4.3. As such, the developments here are quite distinct from the deep learning approaches of the previous chapters. Nevertheless, potential bridges between the two frameworks are mentioned, and will be the subject of future work.

Section 8.1 provides a brief introduction and overview of the main ingredients and results. Section 8.2 contains a general description of probabilistic inversion with GPR. Information about the input data used for the computation can be found in Section 8.3. The results are discussed in detail in Section 8.4, with further implementation details given in Section 8.5. The chapter concludes with a summary and outlook in Section 8.6. The contents of this chapter have been published in [5] together with Jan Horak, Jan M. Pawlowski, José Rodríguez-Quintero, Jonas Turnwald, Nicolas Wink, and Savvas Zafeiropoulos.

## 8.1 Introduction

As long as direct, ab-initio computations are prohibited by the real-time sign problem, spectral reconstruction represents the most promising route towards an understanding of non-equilibrium processes in strongly correlated systems. In this chapter, the subject is approached from the perspective of probabilistic inverse theory with Gaussian processes. The key insights that allow GPR to be applied to spectral reconstruction and related linear inverse problems have previously been formulated in [306] and are discussed in detail below. Most importantly, the derivation exploits the inherent analytic tractability associated with Gaussian statistics, which makes it possible to write down the posterior distribution of predictions conditioned on indirect observations in closed form.

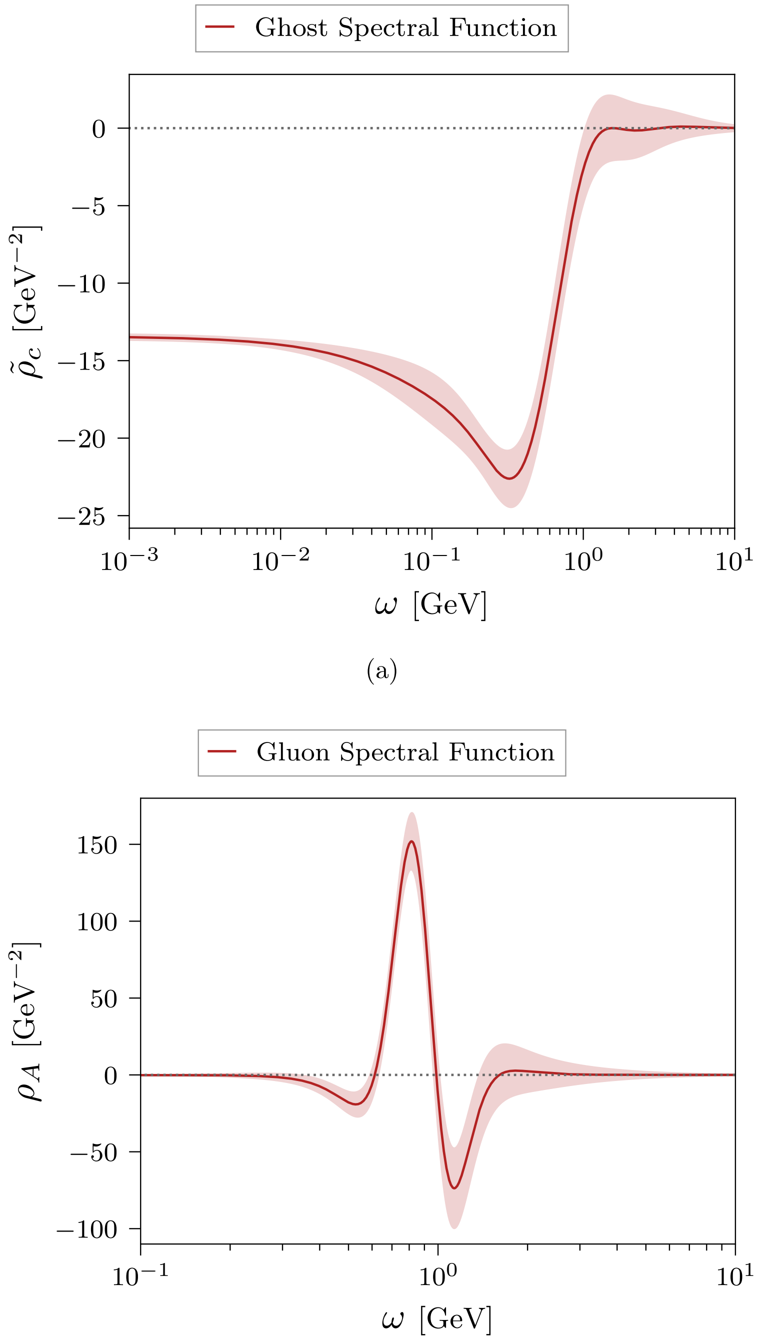
The approach is applied to the computation of ghost and gluon spectral functions based on recent results from 2+1 flavor lattice QCD with domain wall fermions at a pion mass of 139 MeV [307, 308]; further details and references are provided in Section 8.3.1. These lattice data for the ghost dressing function and gluon propagator are shown in Figure 8.1. Furthermore, the systematic error control is improved by incorporating additional data in the infrared (IR) and ultraviolet (UV) regimes from functional renormalization group (fRG) and Dyson-Schwinger (DSE) computations in Yang-Mills theory and QCD [117, 119, 309–313], mostly obtained within the fQCD collaboration. These additional input data and benchmarks provided by one-parameter families of solutions from functional computations are matched to the continuum-extrapolated lattice data; see Section 8.3.2 for details.



(a)

(b)

Figure 8.1: Plots showing the ghost dressing function (a) and gluon propagator (b) from 2+1 flavor lattice QCD simulations, extended by functional computations in Yang-Mills theory and QCD and compared against the correlators obtained from the spectral functions shown in Figure 8.2.



(a)

(b)

Figure 8.2: Plots showing the continuous part of the ghost (a) and the gluon (b) spectral functions computed from the QCD correlators shown in Figure 8.1 using GPR. Shaded areas represent the $1\sigma$-bands of plausible solutions around the mean prediction based on the available observations and precision.

An important property of both the ghost and gluon spectral functions, $\rho_c$ and $\rho_A$, is that the total spectral weight vanishes,

$$\int_0^\infty \frac{\mathrm{d}\omega}{\pi}\, \omega \rho_{c/A}(\omega) = 0 \ . \tag{8.1}$$

For the gluon, this is the well-known Oehme-Zimmermann superconvergence (OZS) condition [314, 315]; for more recent discussions with general fields, see [117–119]. In this work, it is assumed that the gluon spectral function only consists of a continuous part $\rho_A = \tilde{\rho}_A$ satisfying Equation (8.1). This is the generic structure suggested by all functional equations describing the gluon propagator due to the ghost being massless. While derivatives of $\delta$-functions are formally also allowed, these structures are excluded due to the absence of a generic mechanism generating the required roots of the inverse gluon propagator on the real momentum axis. In turn, due to the $1/p^2$ behavior of the Euclidean lattice ghost propagator in the IR, the associated spectral function exhibits a particle peak at vanishing frequency in addition to its continuous part, i.e.

$$\rho_c(\omega) = \frac{\pi}{Z_c}\frac{\delta(\omega)}{\omega} + \tilde{\rho}_c(\omega) \ , \quad \int_0^\infty \frac{\mathrm{d}\omega}{\pi}\, \omega\, \tilde{\rho}_c(\omega) = -\frac{1}{Z_c} \ , \tag{8.2}$$

where $\delta(\omega)/\omega$ has to be understood as a limiting process $\delta(\omega - m)/\omega$ with $m \to 0^+$. Evidently, for $Z_c = 1$ and $\tilde{\rho}_c = 0$ the ghost propagator reduces to the classical one.

The spectral function predictions resulting from the application of GPR are already shown in Figure 8.2. An in-depth discussion of these results takes place in Section 8.4. The associated correlators agree with the input data within the given statistical uncertainties as shown in the bottom panels of Figure 8.1, where the posterior GPs for the correlators are evaluated at the fixed momenta provided by the lattice data, which is then subtracted leaving the error bars intact. The total mean squared errors amount to $\sim 5 \cdot 10^{-6}$ for the ghost and $\sim 4 \cdot 10^{-5}$ for the gluon.

## 8.2 Probabilistic inversion with GPR

In this section, the main ingredients for spectral reconstruction with GPR are presented, based primarily on Section 4.3 and the developments reported in [306]. In the general context of inverse theory, [316] provides a recent review.

We assume our knowledge of the spectral function $\rho(\omega)$ to be described by a GP, written as

$$\rho(\omega) \sim \mathcal{GP}(\mu(\omega), C(\omega, \omega')) \ , \tag{8.3}$$

where $\mu(\omega), C(\omega, \omega')$ denote the mean and covariance functions. Importantly, in this approach we do not restrict the space of possible solutions by choosing a specific functional basis, which often leads to spurious artifacts in the reconstruction in order to compensate for unrepresentable features. Instead, the GP defines a distribution over families of functions with rather generic properties, specified via the kernel parametrization described below.

The KL integral in Equation (3.4) is a linear transformation that preserves Gaussian statistics. Hence, given Equation (8.3) one may obtain statistical predictions $G_i$ at $N_G$ specified momenta $p_i$ as

$$\begin{aligned} G_i \sim \mathcal{N}\Bigg( &\int \mathrm{d}\omega\; K(p_i,\omega)\mu(\omega), \\ &\int \mathrm{d}\omega\, \mathrm{d}\omega' K(p_i,\omega)C(\omega,\omega')K(p_j,\omega')\Bigg) \\ &\equiv \mathcal{N}\big(\tilde{\mu}_i, \tilde{C}_{ij}\big)\ . \end{aligned} \tag{8.4}$$

Here, $\mathcal{N}$ denotes a multivariate normal distribution, to be distinguished from distributions over function space denoted by $\mathcal{GP}$. Statistical uncertainties associated with individual prediction points $\tilde{\mu}_i$ may be computed from the diagonal of the covariance matrix as $\tilde{\sigma}_i = \sqrt{\tilde{C}_{ii}}$.

Conversely, the framework also enables inference in the opposite direction. The inherent analytic tractability associated with Gaussian statistics allows formulating the conditional distribution for $\rho(\omega)$ given observations $G_i$ in closed form. The full expression may then be derived as

$$\begin{aligned} \rho(\omega)\,|\,G_i \sim \mathcal{GP}\Bigg(&\mu(\omega) + \\ \sum_{i,j=1}^{N_G} \int \mathrm{d}\eta\, K(p_i,\eta)C(\eta,\omega) &\left(\tilde{C} + \sigma_n^2 \cdot \mathbf{1}\right)^{-1}_{ij} (G_j - \tilde{\mu}_j)\ , \\ C(\omega,\omega') - \sum_{i,j=1}^{N_G} &\int \mathrm{d}\eta \mathrm{d}\eta'\, K(p_i,\eta)C(\eta,\omega) \\ &\left(\tilde{C} + \sigma_n^2 \cdot \mathbf{1}\right)^{-1}_{ij} K(p_j,\eta')C(\eta',\omega')\Bigg)\ . \end{aligned} \tag{8.5}$$

The GP in Equation (8.5) encodes our knowledge of the spectral function after making observations of the propagator and accounting for observational noise with variance $\sigma_n^2$. The corresponding expressions for the dressing function instead of the propagator can be immediately obtained by inserting an additional factor of $p_i^2$ at every occurrence of the KL kernel $K(p_i, \omega)$ in Equations (8.4) and (8.5).

The flexibility of the approach makes it possible to also incorporate further available prior information in various forms into the predictive distribution in the same manner, yielding similar though somewhat more complicated expressions. This may include e.g. direct observations of $\rho$ and its derivatives, assumptions about the asymptotic behavior, or global normalization constraints.

In this work, the standard RBF kernel defined in Equation (4.23) is used. Nevertheless, designing custom kernels for specific problems has been shown to greatly increase the usefulness of the approach in various settings and is also promising here. In particular, it may be interesting to construct kernel functions that can be integrated analytically against the KL kernel, such that the frequency integrals in

Equations (8.4) and (8.5) may be carried out analytically instead of numerically. To this end, one could potentially employ functions of Breit-Wigner type as done for the spectral function itself in [117]. In contradistinction, one may use them to instead define a suitable GP kernel, thereby still avoiding the restriction to a specific functional basis as previously mentioned. We touch upon on this and other possible improvements to the present approach in Section 8.6.

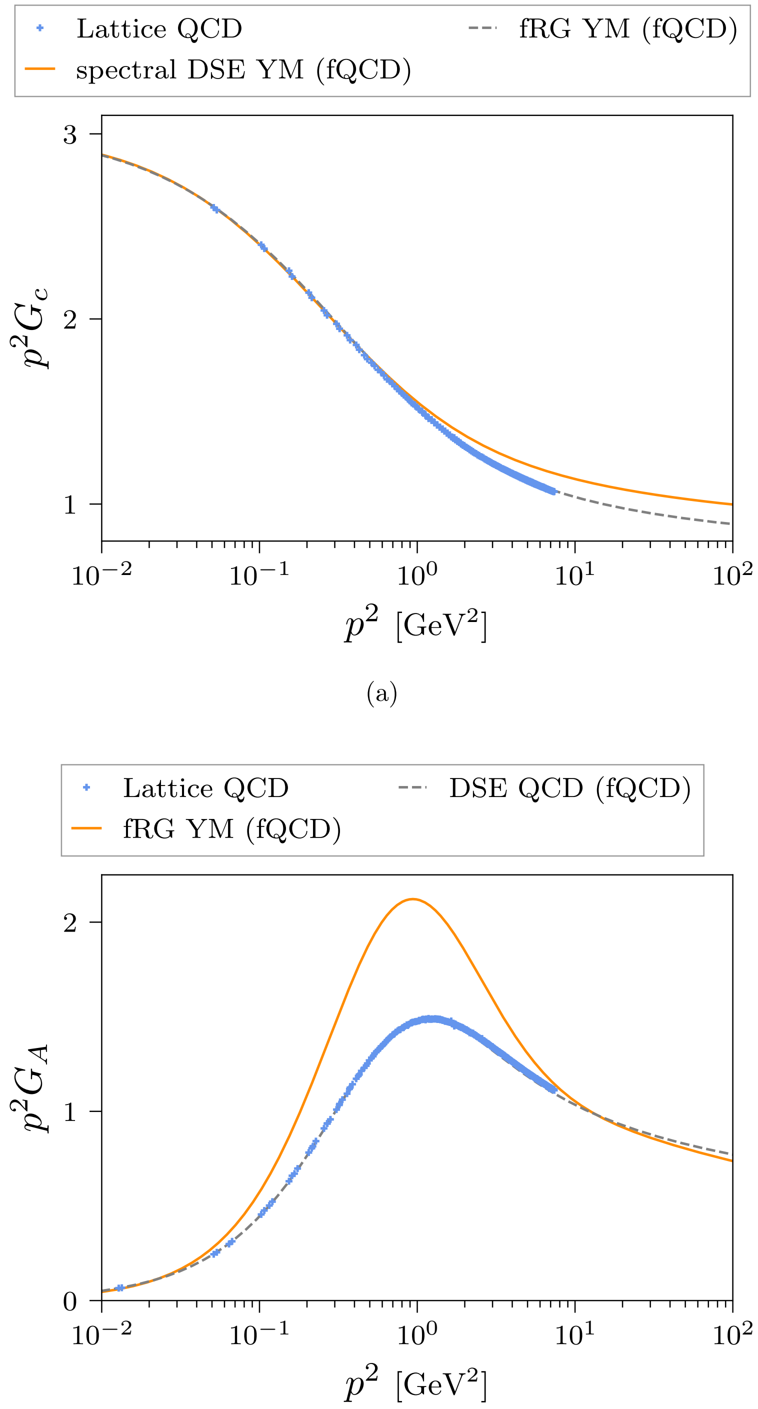
Furthermore, it should be emphasized that the GPR method in principle does not require us to choose a specific set of nodes $\omega_i$. In fact, instead of computing a discrete set of point predictions or coefficients of a predefined functional basis, the prediction for $\rho$ is obtained as a function of $\omega$, albeit only implicitly via the kernel formulation. In particular, the GP also allows computing all of the derivatives of the prediction analytically at any point—including the associated statistical uncertainties—by differentiating the expressions in Equation (8.5) with respect to $\omega$ (as well as $\omega'$ for the covariance). A finite set of nodes $\omega_i$ is chosen only at inference time in order to evaluate the GP, however, the choice is completely arbitrary within the given domain. This property is one of the most attractive features of GPR for spectral reconstruction and probabilistic function prediction in general.

## 8.3 Input data

### 8.3.1 Lattice calculations

In the past two decades, increasing interest in the momentum behavior of the fundamental two-point Green's functions in QCD as well as further correlation functions of higher order has triggered respective lattice calculations of Yang-Mills and QCD propagators; see e.g. [317–331]. The lattice data employed in this work were obtained from configurations generated by the RBC/UKQCD collaboration—first introduced in [332–336]—with 2+1 dynamical quark flavors using the Iwasaki [337] and domain wall fermion [338, 339] actions, respectively for the gauge and quark sectors, at the physical point (a pion mass amounting to 139 MeV) by the particular implementation of the Möbius kernel [340]. These developments were then exploited in [307, 308] in order to calculate the gluon and ghost propagators as well as the strong coupling in a particular scheme [341–343], and an effective charge stemming from it [344]. A description of this calculation is given, for instance, in [326].

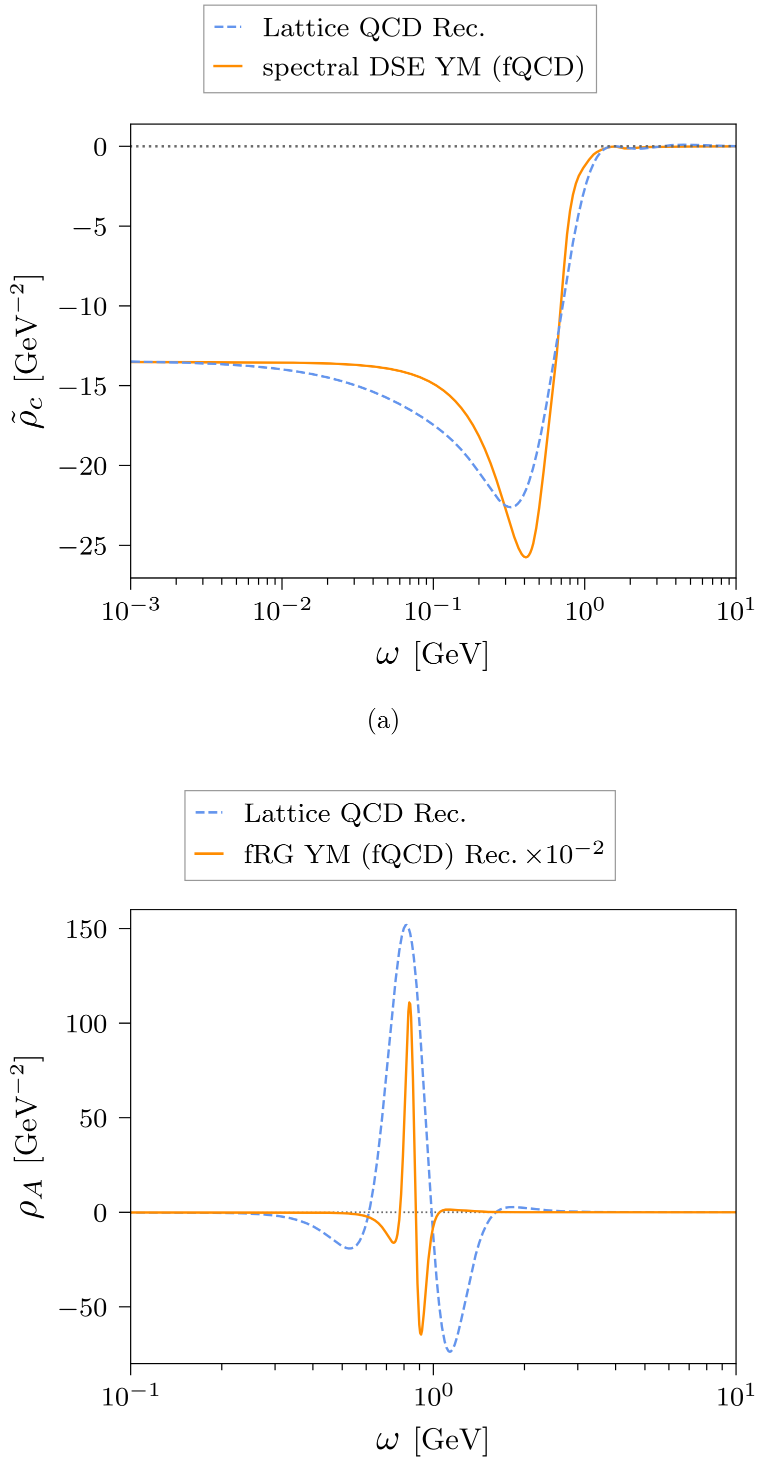
In computing propagators that properly feature the physical running with momenta, the data should be thoroughly cured from regularization artifacts. In particular, as explained in [307], the lattice results are obtained after a careful scrutiny of discretization effects, thereby accounting for the continuum-limit extrapolation, following [345]. As a noteworthy remark, a recent work [331] has revealed the key role played by the procedure of [345] for an adequate removal of discretization artifacts in achieving a consistent description of Yang-Mills two- and three-point correlators, involving both lattice and DSE results.



(a)

(b)

Figure 8.3: Plots showing ghost (a) and gluon (b) dressing functions in 2+1 flavor QCD and Yang-Mills (YM) theory, obtained from the lattice and functional methods computations discussed in Section 8.3.



(a)

(b)

Figure 8.4: Plots comparing the continuous part of the ghost (a) and the gluon spectral function (b) from different approaches in 2+1 flavor QCD and Yang-Mills (YM) theory, as discussed in Sections 8.3 and 8.4. The ghost spectral function $\rho_c$ features an additional massless particle pole in the origin; see Equation (8.2).

### 8.3.2 Functional methods

This section briefly summarizes the results from functional computations in Yang-Mills theory and QCD that are employed in this work to provide additional prior information for the reconstruction. For reviews on the application of functional methods in this context, see e.g. [346–349].

We use the real-time Yang-Mills results from [119] to extend the lattice data of the ghost dressing into the deep IR, as shown in Figure 8.3a. The approach also provides direct access to the associated spectral function, which we employ to fix the low-frequency asymptotic behavior. It is obtained via the spectral ghost DSE, building upon the technique of spectral renormalization [350]. Making use of Equation (3.4) for the ghost and gluon propagator, the momentum integrals appearing in the loop diagrams of the ghost propagator DSE can be solved analytically. This preserves the full analytic momentum dependence and allows evaluating the equation on the real momentum axis. The spectral function can then be directly extracted from the real-time propagator DSE via Equation (3.5); see Figure 8.4a for a comparison to the result of the reconstruction. As input gluon spectral function, the result of [117] based on the scaling solution obtained via the fRG in [309] is used. Assuming a spectral representation for the gluon propagator, in both scaling and decoupling scenario the IR behavior of the gluon spectral function follows directly from the propagator [117]. This is utilized to modify the given scaling spectral function such that we obtain a decoupling-type gluon propagator matching the value of the given lattice propagator well within the given uncertainties.

The lattice QCD data for the gluon propagator are extended towards the UV using earlier results from functional computations in Yang-Mills theory [309]. Differences to the 2+1 flavor QCD result for the gluon propagator reported in [313], being based on [310], are comparably small in the relevant momentum range. A stronger deviation can be observed in the dressing functions, as shown in Figure 8.3b. Despite these differences, the reconstruction still produces remarkably reliable results; see Figure 8.1b. Nevertheless, at some point the Yang-Mills UV extension should be replaced by the 2+1 flavor QCD data from [313] in order to improve the accuracy of the result and mitigate any potential issues. For related results and further correlation functions see [311, 312, 351, 352]. More specifically, the fRG results in [309] are derived within an advanced approximation where the momentum dependence of all vertices is approximated at the symmetric point, for respective DSE results see [353]. For our purposes, this dataset provides the optimal trade-off for momentum range versus accuracy. Due to the high numerical precision, the results are particularly well-suited as an input for spectral reconstruction. The Yang-Mills data have already been employed for this purpose in [117] and we use this earlier reconstruction for comparison; see Figure 8.4b. In summary, the extension of the 2+1 flavor lattice data with the high precision Yang-Mills data up to momenta $p^2 = 10^2\,\mathrm{GeV}^2$ allows a more direct comparison (in terms of scales) with the Yang-Mills reconstruction in [117], while only modifying the large frequency tail of the gluon spectral function for frequencies $\omega \gtrsim 5\,\mathrm{GeV}$, see Figure 8.4.

## 8.4 Discussion of results

The computation of the ghost spectral function is performed using the aforementioned standard RBF kernel. Extending the lattice input data for the dressing function into the deep IR and simultaneously fixing the low-frequency asymptotics of the spectral function using the direct real-time computation in Yang-Mills theory is achieved by treating the spectral DSE result as an additional observation. This procedure uniquely determines the non-zero value of $\rho_c$ for $\omega \to 0^+$, but also increases the reliability of the solution in the most interesting central region with respect to the kernel hyperparameters. Using just the lattice data without the extension by the spectral DSE result leads to a much higher variance in the solution space, with widely different asymptotic behaviors of solution candidates in the IR. The kernel hyperparameters are chosen by optimizing the associated likelihood of observations with an additional Gaussian hyperprior, which we achieve through a fine-grained grid scan; see Section 8.5 for details. The resulting spectral function in Figure 8.2a accurately reproduces the dressing function data within the uncertainties displayed in Figure 8.1a, with a total mean squared error of $\sim 5 \cdot 10^{-6}$.

The features of the prediction are strikingly similar to the aforementioned Yang-Mills result shown in Figure 8.4a in Section 8.3, even though only the IR limit is incorporated into the reconstruction. This is expected heuristically, since the ghost only interacts with the quarks indirectly via the gluon vertices, and the effects of introducing dynamical quarks must hence be of higher order. The similarity is particularly notable considering that the methods are conceptually very different.

For the reconstruction of the gluon spectral function, the extension of the lattice input data into the UV using the earlier fRG computation leads to greatly enhanced stability of the reconstruction with respect to the kernel hyperparameters, similar to the ghost. In particular, it ensures convergence to zero for $\omega \to \infty$, whereas with just the lattice data we often observe convergence to a non-zero constant and in some cases even pathological divergences. A modified frequency scale is used in the RBF kernel in order to suppress spurious oscillations in the IR and UV tails. The hyperparameters are again obtained via optimization of the likelihood with Gaussian hyperpriors while approximately enforcing the OZS condition; see Section 8.5 for details. The reconstruction shown in Figure 8.2b accurately reproduces the lattice data within the given uncertainties, as shown in Figure 8.1b, with a total mean squared error of $\sim 4 \cdot 10^{-5}$. While also being fully consistent, deviations from the lattice propagator are somewhat stronger than for the ghost dressing function and seem to become more pronounced in the IR. This is likely caused by the comparably large uncertainties of the lattice data at small momenta.

The peak structure of the spectral function appears similar to an earlier reconstruction of the Yang-Mills propagator in the fRG framework [117], shown in Figure 8.4b. We emphasize that the UV extension is done with the Yang-Mills data of [309] instead of the full 2+1 flavor results from [313], which facilitates the comparison with the Yang-Mills reconstruction [117]. In particular, the positions of the leading positive peaks approximately coincide, with $\omega \approx 0.818$ for the present result and $\omega \approx 0.835$ for the fRG computation. This reflects the approximate coincidence of the peaks of the Euclidean gluon dressing functions shown in Figure 8.3a. We also note that a small peak to the right of the second local minimum is present in both spectral functions. This feature may be a generic reconstruction artifact since it is not necessitated by theoretical considerations, but is observed in both results from conceptually very distinct methods. However, the comparably large uncertainties in this region also include plausible solutions without additional zero-crossings.

Significant differences between the two results are observed mainly in the overall peak height and width. Generally, the QCD result for the gluon is expected to differ more strongly from the pure gauge theory than the ghost due to the direct coupling to quarks. However, differences may also be attributed in part to the limited availability and precision of data and the resulting difficulty in resolving highly peaked structures. Generating narrower peaks with greater amplitudes by allowing the kernel's magnitude parameter $\sigma_C$ to increase and the length scale $l$ to decrease leads to stronger oscillations in the solution. This is a common feature of conceptually similar reconstruction approaches, such as linear regression with a Tikhonov regularizer (also called ridge regression), which has been applied e.g. in [132]. Introducing such a regularization scheme, which is equivalent to assuming a Gaussian prior, leads to a favoring of solutions that are closer to zero. This additional bias can introduce the unwanted oscillations. Within the GPR approach, the kernel hyperparameters provide more detailed control over the regularization and can be tuned to deliberately suppress such unphysical features. However, this may result in spectral functions that are naturally flatter, which must be taken into account when interpreting and utilizing the result. This demonstrates one of the key advantages of GPR, namely the possibility to dynamically adjust the resolution depending on the available input data, while still matching the observations as accurately as possible.

Although the obtained spectral functions reproduce the lattice data to high accuracy, the asymptotic behaviors of the mean predictions in the deep IR and UV differ from the analytic results derived in [117]. In particular, different scaling exponents are observed and the gluon spectral function shows the opposite sign in the UV. Nevertheless, the analytically expected behavior is still plausibly contained within the computed errors, which are comparably large in these regimes. This indicates that not enough prior information is available to the GP from just the data in order to accurately resolve the tails of the spectral functions, which may come as no surprise. While this issue does not affect the results in the regions of interest, it may be problematic for precision computations that use these spectral functions as inputs. In order to directly enforce the correct asymptotics, potential approaches are the incorporation of the analytically known behaviors into the prior means of the GPs or constructing specialized kernel functions. Furthermore, exploiting the available analytic results to provide additional prior information about the derivative structure may be particularly helpful in stabilizing the tail behavior. To achieve this, one may again write down the joint distribution of the predicted spectral function at any frequency and its associated derivatives to arbitrary order in closed form and derive the conditional posterior distribution similar to Equation (8.5).

## 8.5 Implementation

In this section, certain aspects of the implementation are explored in more detail. We first consider some details of the hyperparameter optimization procedure and the required computational effort. Subsequently, further information is provided about data usage and the incorporation of additional information, as well as specific kernel design choices. We also discuss the implementation of additional theoretical constraints for the reconstructions reported in this work, in particular regarding the OZS condition for the gluon defined in Equation (8.1).

### 8.5.1 Hyperparameter optimization and computational cost

To find optimal values for the kernel's hyperparameters, a fine-grained grid scan of the NLL is performed, with additional hyperpriors where necessary. Alternatively, the NLL may also be minimized with a gradient-based ansatz using a standard optimizer such as L-BFGS [354]. However, mapping out the posterior distribution in more detail tends to be highly instructive for the problem at hand. It is also less prone to numerical problems such as unstable directions and violation of positive definiteness of the covariance, as these can be identified early on, and should hence be preferred when feasible. This is also where the bulk of the computational effort goes, as it involves calculating for each individual grid point the comparably expensive inverse and determinant of the covariance matrix. For a matrix of size $N \times N$, this scales naively like $\mathcal{O}(N^3)$, as discussed already in Section 2.3.2 in the context of the pseudofermion method. For very large datasets where their direct evaluation becomes infeasible, one may resort to cheaper linear solvers for the computation of the inverse, as well as stochastic approximations for the determinant. However, this is unlikely to become necessary in this particular context, since the size of the datasets and the required number of points to achieve a sufficient resolution in the prediction are typically very limited. Cost may also be mitigated by scanning the parameter space hierarchically, starting at low resolution and zooming into the interesting regions. For higher-dimensional parameter spaces—which may result from more sophisticated kernel formulations—a MCMC approach for the exploration of the posterior distribution may be more appropriate, such as provided by the STAN library [355]. This could also lead to an improved estimate of the prediction error.

The hyperparameter scan is trivially parallelizable, as each grid point can be treated independently. At the scale of the present work, each instance was handled by a standard CPU node with low performance requirements. Some first tests were also conducted on a single machine, where mapping out the parameter space for each reconstruction with medium resolution took a few hours at most. In comparison to finding the optimal hyperparameters, the subsequent inference step is negligibly cheap. Of course, the total computational effort for the reconstruction is dwarfed by the requirements of the large-scale lattice simulations described in Section 8.3.1, which are orders of magnitude more expensive.

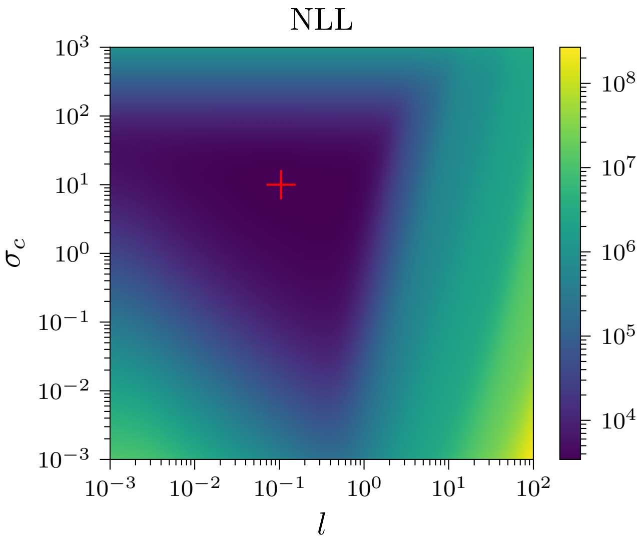


Figure 8.5: Heatmap of the NLL as a function of the RBF kernel hyperparameters $\sigma_C, l$ for the reconstruction of the ghost spectral function, with an additional zero-mean Gaussian hyperprior for $\sigma_C$. A unique minimum can be identified, which provides the optimal values used for the results shown in Figures 8.1a and 8.2a.

## 8.5.2 Reconstruction details

### Ghost

As mentioned previously, in the case of the ghost spectral function, we treat the low-frequency asymptotics extracted from the direct DSE computation in Yang-Mills theory as an additional observation for the GP. This is only possible for the ghost, as a similarly direct determination of the Yang-Mills gluon spectral function is currently not available. The procedure is implemented by including the value of $\rho$ at $\omega = 0$ in the construction of the joint distribution of observations and predictions discussed in Section 8.2, which makes the derivation slightly more complicated. In particular, one needs to compute additional expressions for the covariances of the point $\rho(0)$ and the correlator data. This mainly requires some programming headache, but carries no further conceptual difficulty.

In the identification of optimal hyperparameters for the RBF kernel via the high-resolution grid scan described in the previous section, an unstable direction in the magnitude parameter $\sigma_C$ was noted, which was cured by subjecting it to a zero-mean Gaussian hyperprior. As an illustrative example, the heatmap for the NLL including this additional regularization term for $\sigma_C$ is shown in Figure 8.5. By introducing the hyperprior, a unique set of parameters maximizing the likelihood can be determined, which is used to compute the mean prediction for the ghost spectral function.

### Gluon

In the case of the gluon spectral function, no real-time result in Yang-Mills theory is available to fix the asymptotics. However, as an additional theoretical constraint, we require the solution to respect the aforementioned OZS condition defined in Equation (8.1). While one might expect this to further complicate the reconstruction, it actually helps in narrowing down the space of plausible solutions. The condition can simply be enforced approximately by treating it as an additional indirect observation and checking it a posteriori, similar to the low-frequency asymptotics of the ghost spectral function. The associated transformation is here just the convolution with $\omega$ instead of the KL integral. We confirm that the OZS condition is fulfilled with a relative accuracy of $\sim$1%, computed by evaluating the ratio of the left-hand side of Equation (8.1) and the same expression using the modulus of the integrand, i.e. $\int_0^\infty \mathrm{d}\omega\, |\omega \rho_A(\omega)|$.

As already mentioned in Section 8.4, the standard RBF kernel is modified by non-linearly rescaling the frequency as $\omega \to \tilde{\omega} = \omega^4(1+\omega^4)^{-1}$ before computing the squared distance. This leads to a strongly improved asymptotic stability of the spectral function, in particular at large frequencies, compared to just using $\omega$ itself. The procedure may be interpreted either as a non-stationary modification of the kernel or as a preprocessing step for the data to the same effect. However, similar improvements may be achieved more consistently by constructing suitable kernels that enforce the correct asymptotic behavior a priori.

## 8.6 Summary and outlook

In this chapter, GPR was applied to the computation of ghost and gluon spectral functions in 2+1 flavor QCD at the physical point. These spectral functions are the pivotal building blocks of diagrammatic representations for bound state equations such as Bethe-Salpeter and Faddeev equations, see e.g. [356–358], as well as transport coefficients, see e.g. [124, 359]. Importantly, the gluon spectral function has a pronounced quasi-particle peak, the position of which is related to the mass gap in QCD. This extends previous vacuum and finite-temperature results in Yang-Mills theory [117, 124] to physical QCD. Our findings provide non-trivial QCD support to the phenomenological use of quasi-particle gluon spectral functions for transport computations; see [360] for a recent review. Moreover, the computed spectral functions can be directly employed as first-principle QCD inputs in order to systematically improve the respective phenomenological approaches towards a first-principle treatment of QCD transport processes.

These promising phenomenological applications of the present results also highlight the necessity of further improving the reconstruction approach itself, for which a number of potential directions can be envisaged. This includes the aforementioned possibility of designing custom kernels for the problem at hand, potentially with analytic integrability against the KL kernel. Constructing suitable, expressive kernels may also be automated and improved through the use of hyperkernels [361] or techniques such as deep kernel learning [362]. To account for some variability in the kernel hyperparameters, one may replace the maximum likelihood approach by an integral over parameter space using a suitable hyperprior which encodes any prior assumptions. Alternatively, optimal hyperparameters may also be selected based on a data-driven machine learning approach, using datasets consisting of pairs of correlators and associated spectral functions.

Furthermore, the flexibility of the GPR framework allows the incorporation of various supplementary constraints derived from theoretical arguments, such as information about derivatives, known asymptotic behaviors, or normalization conditions. This is expected to further improve the accuracy and reliability of the reconstruction, in particular for the IR and UV tails of the spectral functions that are otherwise difficult to resolve. This will be the subject of future work, accompanied by direct functional computations of further spectral properties along the lines of [119, 350].

The immediate next steps in the endeavor towards unveiling real-time physics of QCD are the application and extension of the present numerical framework to quark propagators as well as correlation functions computed at finite temperature. This will enable quantitative studies of hitherto theoretically inaccessible non-equilibrium properties of QCD in the transport phase of heavy-ion collisions within a first-principle approach.

# 9 Summary and outlook

In this thesis, I have investigated the applicability of modern AI algorithms to three longstanding computational barriers in lattice QCD and provided several successful proof-of-principle demonstrations. The development of normalizing flow architectures for the generative neural sampling of theories with dynamical fermions—aiming at the treatment of critical slowing down in lattice calculations at scale—may be viewed as the main contribution. With this approach, topological freezing was solved in the Schwinger model at criticality, in a situation where traditional algorithms fail to achieve sufficient ergodicity required for reliable results. Further, normalizing flows were applied to the density-of-states approach to complex action problems. In contrast to conventional strategies relying on restricted MCMC calculations of a derivative quantity, the density can be computed directly within this framework. Using the flow-based method, it was demonstrated that the Lee-Yang zeroes of the partition function of a scalar field theory with an imaginary external field can be successfully located. In the context of real-time properties of QCD, the spectral reconstruction problem was approached from the perspective of probabilistic inverse theory. Specifically, ghost and gluon spectral functions were calculated using Gaussian process regression on combined Euclidean correlation function data from state-of-the-art lattice and functional computations.

The next step in the endeavor to apply normalizing flows to lattice QCD at scale is the construction of suitable architectures capable of modeling $SU(3)$ gauge fields coupled to fermions in four spacetime dimensions. If successful, this line of research may usher in a new era of precision QCD calculations at hitherto inaccessible physical volumes, thereby unlocking novel insights from first principles in nuclear and high energy physics. Furthermore, the present work on solving complex action problems may be useful for the gargantuan task of mapping out the QCD phase diagram at finite temperature and chemical potential. However, the approach will require considerable development efforts before it can be applied in this context. Finally, apart from making further improvements to the proposed spectral reconstruction approach, it will be exciting to apply the method to the study of spectral representations of quarks as well as other interesting objects, such as glueballs and the strong coupling constant. The results of this work will provide important inputs to phenomenological descriptions of non-equilibrium processes, such as heavy-ion collisions at particle colliders. In summary, these research directions may help to complete our understanding of the standard model of particle physics in some of its most notoriously difficult aspects. Furthermore, I hope that the insights gained along the way will also have applications in the study of quantum gravity, such that we may eventually achieve a unifying picture of fundamental physics.

Apart from our ambitions to further elucidate the nature of fundamental interactions, the hard computational challenges described in this thesis are also extremely fascinating in their own right. Solving such problems should not be viewed as merely a means to an end, but would constitute a major intellectual achievement of our civilization. The observation that—even though we seem to be more or less intelligent beings inhabiting this universe—some of nature's deepest secrets currently appear to be inaccessible to us, is as insulting as it is incentivizing. Instead of losing ourselves in the philosophical aspects of this conundrum, the development of AI technologies powerful enough to answer our questions and illuminate some of the most interesting aspects of the physical world is a much more practical approach. While it may be the case that our primate brains by themselves are too limited, we should follow in the footsteps of what human beings have done since the stone age, namely use our given smarts to fashion increasingly sophisticated tools that help us overcome our evolutionary limitations. Furthermore, as physicists in particular, we are in a prime position to go beyond the mere application of these novel methods to problems in our own field, and also push the current boundaries of AI to help build the next generation of transformative technologies. Already today, researchers are discovering exciting parallels between the dynamics of deep neural networks and interacting quantum field theories. This may ultimately contribute to an improved understanding not only of artificial, but also human intelligence. Hence, this wonderful emerging synergy of AI and physics will not only help us grasp the nature of reality and hopefully transform the world into a better place, but may also lead us to magnificent new insights about ourselves.

# Acknowledgments

I would like to thank my advisor Jan M. Pawlowski for his guidance and for providing a wonderful scientific environment for me to thrive in.

My parents Sun-Ah Yoon and Albert Urban, my brother Damian Urban, my sister-in-law Karin Urban and her family, as well as my further relatives in both South Korea and Germany.

My better half Jani Takhsha for her love and support.

My close friends over the years, in alphabetical order: Martin Brinkmöller, Hamied Capitaine, Maya Harms, Moad Abd El Hay, Christopher Hills, Friederike Ihssen, Jonas Karthein, Christian Lorenz, Jonas Massa, Marco Müller, Maximilian Richter, Fabio Schlindwein, Christian Schmidt, Cristian Trache.

My collaborators and colleagues: Michael Albergo, Felipe Attanasio, Marc Bauer, Stefan Blücher, Denis Boyda, Lillian de Bruin, Cristóbal Corvalán, Kyle Cranmer, Lukas Corell, Dan Hackett, Albert Hofmann, Jan Horak, Gurtej Kanwar, Margit Liu, Sébastien Racanière, Danilo Rezende, José Rodríguez-Quintero, Fernando Romero-López, Alexander Rothkopf, Manuel Scherzer, Saswato Sen, Phiala Shanahan, Ion-Olimpiu Stamatescu, Nils Strodthoff, Jonas Turnwald, Sebastian Wetzel, Nicolas Wink, Savvas Zafeiropoulos, Felix Ziegler.

The Heidelberg Institute for Theoretical Physics and Massachusetts Institute of Technology for providing work and computing resources.

# A Novel insights from interpretable AI

## A.1 Lattice datasets

All field configurations composing the datasets used in Section 5.2 are generated with the parameters listed in Table A.1. A single, labeled sample is given by the mapping

$$(\phi, \kappa): \qquad \{\phi_n\} = \{\phi_n \mid n \in \Lambda\} \longrightarrow \kappa \,. \tag{A.1}$$

In order to approximately encode the $\mathbb{Z}_2$ symmetry in the trained neural networks, we use the same configurations twice in the dataset, just with a globally flipped sign. This raw data is directly used to train the CNN. For the MLP, the samples are preprocessed by computing the chosen set of observables for each configuration,

$$(\mathcal{O}, \kappa): \qquad \{|M|, |M_s|, G_c(t)\} \longrightarrow \kappa \,. \tag{A.2}$$

In this case, we can simply take the modulus of the magnetizations without losing information, since only two branches with exactly opposite signs are present in the phase diagram. Due to the finite expectation value of the staggered magnetization, the AFM phase contains unphysical negative correlations. In order to remove these lattice artifacts, we adapt the usual time-sliced two-point correlator to

$$G_c(t) = \Big| \langle \phi(t)\phi(0) \rangle - M^2 - (-1)^t M_s^2 \Big| \,. \tag{A.3}$$

Generally speaking, LRP is designed for classification problems. Therefore, we discretize $\kappa$ to facilitate the formulation of the inference objective as a classification task. All values of $\kappa$ are transformed into individual bins and the networks are tasked to predict the correct bin. In order to retain a notion of locality, the true bins are additionally smeared out with a Gaussian distribution, resulting in the target labels

$$\kappa \longrightarrow y_b = e^{-\frac{(\kappa_b - \kappa_{\text{True}})^2}{2\sigma^2}} \,. \tag{A.4}$$

Here, $b$ denotes the bin number, and the variance was set to $\sigma = 3\Delta\kappa$. In combination with a MSE loss, we obtain qualitatively similar prediction results compared to a standard regression approach.

| $N$ | $\lambda$ | $M$ | $g$ | $\Delta\kappa$ | #samples per $\kappa$ |
|---|---|---|---|---|---|
| 16 | 1.1 | 20 | 0.25 | 0.005 | 200 - Training set<br>100 - Test set |

Table A.1: Action/simulation parameters used for training and test dataset.

## A.2 Propagation rules

This section contains a summary of the mathematical background of LRP, in particular regarding the propagation rules. Generally, the relevance $R_j$ depends on the activation of the previous layer $x_i$. Given some input to the network, its predicted class $f$ is identified by the output neuron with the largest response. This neuron's activation $R_f^{out}$, along with $R_i^{out} = 0$ for all other classes $i \neq f$, defines the relevance vector. This output layer relevance can then be backpropagated through the whole network, which results in the aforementioned heatmap on the input. Importantly, the propagation rules are designed such that the total relevance is conserved,

$$\sum_i R_i^n = \sum_i R_i^{out} \equiv R_f^{out} \ , \tag{A.5}$$

where the index $n$ can indicate any layer. This conservation law ensures that explanations from all layers are closely related and prohibits additional sources of relevance during the backpropagation. A Taylor expansion of this conservation law yields

$$\sum_j R_j(x_i) = \underbrace{\sum_j R_j(\widetilde{x}_i)}_{=0} + \sum_i \underbrace{\sum_j \left.\frac{\partial R_j}{\partial x_i}\right|_{\{\widetilde{x}_i\}} (x_i - \widetilde{x}_i)}_{R_i} \ . \tag{A.6}$$

Here, we choose $\widetilde{x}_i$ to be a so-called root point, which corresponds to an activation with vanishing consecutive layer relevance $R_j(\widetilde{x}_i) = 0$. By definition, it is localized on the layer's decision boundary, which constitutes a hypersurface in the activation space. Hence, the root point is not uniquely defined and we need to impose an additional criterion. However, given such a point, we can identify the first order term as the relevance propagation rule $R_j \mapsto R_i$. The remaining root point dependence gives rise to a variety of possible propagation rules. For instance, the $w^2$ rule minimizes the Euclidean distance between neuron activation $x_i$ and the decision boundary in order to single out a root point. Visualizations of root points, as well as essential derivations and analytical expressions for propagation rules, can be found in [232].

## A.3 Random forest details

Random forests [239] denote a predictive ML approach based on ensembles of decision trees. They utilize the majority vote of multiple randomized trees in order to arrive at a prediction. This greatly improves the generalization performance compared to using a single tree. The elementary building block is a node performing binary decisions based on a single feature criterion. New nodes are connected sequentially with so-called branches. A single decision tree is grown iteratively from a root node to multiple leaf nodes. A concrete prediction corresponds to a unique path from the root to a single leaf. Each node on the path is associated with a specific feature. Hence, we can sum up the contributions to the decision separately for each feature by moving along the path,

$$\text{prediction} = \text{bias} + \sum_i (\text{feature\_contribution})_i \ . \tag{A.7}$$

Here, the bias corresponds to the average prediction at the root node.

We employ the scikit-learn implementation [363] in combination with the TreeInterpreter extension [364]. The latter reference also provides an excellent introduction to the concept of feature contributions.

The random forest is initialized with 10 trees and a maximum tree depth of 10. This parameter is essential for regularization, since an unconstrained depth causes overfitting and thus results in poor generalization performance. In order to fix this parameter, we start at a large value and successively reduce it until the training and test accuracy reach a similar level. This way we can retain as much expressive power as possible in the random forest while simultaneously eliminating systematic errors resulting from overfitting. However, we emphasize that the specific choice of this parameter not relevant to our argument.

## A.4 Network architectures and implementation details

We use the PyTorch framework. The machinery of LRP is included by defining a custom `torch.nn.Module` and equipping all layers with a relevance propagation rule. Furthermore, all biases are restricted to negative values in order to ensure the existence of a root point. For training, we employ the Adam optimizer with default hyperparameters and an initial learning rate of 0.001, using a batch size of 16.

For both networks, the first layer undergoes least absolute shrinkage and selection operator (LASSO) regularization during training, which encourages sparsity and thereby enhances interpretability. This corresponds to simply adding the $L^1$ norm of the respective weights $w_{ij}$ to the MSE loss, which accordingly takes the form

$$L = \frac{1}{d} \sum_{f=1}^{d} (y_f - \hat{y}_f)^2 + \lambda_{\text{Lasso}} \sum_{ij} |w_{ij}| \ . \tag{A.8}$$

Here, $y_f, \hat{y}_f$ denote the prediction and ground truth labels, and $i, j$ the input and output nodes of the first layer. The quantity $\lambda_{\text{Lasso}}$ parametrizes the strength of the regularization.

The network architectures used in this work are given in Tables A.2 and A.3.

| Layer | Specification | Propagation rule |
|---|---|---|
| *Linear* | in=18, out=256 | $w^2$–rule |
| *ReLU* | | $R_i = R_j$ |
| *Linear* | in=256, out=128 | $z^+$–rule |
| *ReLU* | | $R_i = R_j$ |
| *Linear* | in=128, out=140 | $z^+$–rule |
| *LeakyReLU* | negative slope=0.01 | $R_i = R_j$ |

Table A.2: Network architecture of the MLP. The first layer undergoes $L^1$ regularization with $\lambda_{\text{Lasso}} = 5$.

| Layer | Specification | Propagation rule |
|---|---|---|
| *Conv3d* | $\#_{\text{filter}} = 5$, kernel=B, strides=A | $w^2$–rule |
| *ReLU* | | $R_i = R_j$ |
| *MaxPool3d* | kernel=B, strides=B | $z^+$–rule |
| *Linear* | in=1715, out=256 | $z^+$–rule |
| *ReLU* | | $R_i = R_j$ |
| *Linear* | in=256, out=140 | $z^+$–rule |
| *ReLU* | | $R_i = R_j$ |

Table A.3: Network architecture of the CNN, with $A = (1 \times 1 \times 1)$, $B = (2 \times 2 \times 2)$. The first layer undergoes $L^1$ regularization with $\lambda_{\text{Lasso}} = 10$.

| **Model parameters** | $\phi$-Marginal | Gibbs | Autoregressive | Fully Joint |
|---|---|---|---|---|
| flow layers | 3 | 16 | 12 | 12 |
| convs. per layer | 4 | 3 | 10 | 6 |
| number conv channels | 16 | 32 | 64 | 64 |
| $\sigma_\zeta$ | 0.34 | 0.1 | 0.34 | 0.34 |
| $\sigma_\chi$ | - | 0.1 | 0.15 | 0.15 |
| kernel size | 3 | 3 | 3 | 3 |
| activations (inner / final) | SoftPlus/- | LeakyReLU/Tanh | LeakyReLU/- | LeakyReLU/Tanh |
| **Training parameters** | | | | |
| gradient steps | 500k | 30k | 200k | 50k |
| batch size | 3072 | 2000 | 3072 | 3072 |
| learning rate schedule | $10^{-3}$, $10^{-4}$ after 80k | $10^{-3}$, $10^{-5}$ after 20k | $10^{-4}$, $2 \times 10^{-5}$ after 60k, $10^{-5}$ after 120k | $3 \times 10^{-4}$, $6 \times 10^{-5}$ after 30k |
| gradient clipping (value / norm) | 10/32 | -/- | 10/32 | 10/1000 |

Table A.4: Model and training hyperparameters for all architectures discussed in Section 6.5.2. Further details and references are provided in Appendix B, in particular the specifics of the linear operators for the joint autoregressive model that are not listed here.

# B Model and training details for fermionic flows

This section lists all necessary details to reproduce the flow architectures discussed in Section 6.5.2. In particular, all relevant model and training hyperparameters are given in Table A.4. Additional peculiarities of the linear operator and fully joint model implementations not listed in the table are discussed below.

All models are trained using the Adam optimizer with default settings. In some cases, clipping of the gradient value and norm was employed to stabilize training [365]. The deep neural networks providing the context functions and convex potentials in the flow architectures are implemented exclusively in terms of convolutional networks with several hidden layers and channels. We employ ReLU as the nonlinear activation function employed in these networks, in particular the LeakyReLU variant, as well as the SoftPlus function in the case of the CPF layers, as detailed in Section 6.4.1. In some cases, an additional Tanh activation is applied to the output of each network, as specified in Table A.4. For all calculations in this work using the CPF architecture, at initialization we set $w_1 = 5 \times 10^{-3}$ and $w_2 = 1$ required for the layers defined in Equation (6.34). All convolutional layers use a stride of 1.

As mentioned in Section 6.5.1, even-odd preconditioning is not applied for the autoregressive architecture with linear operators, as we observe that the model gives a better approximation to the non-preconditioned action with standard lexicographic ordering. The space of possible model adjustments to make the even-odd decomposition compatible with linear operators is large, and modifications could be explored to improve the results. The conditional density $q(\varphi|\phi)$ is implemented using the composition of 128 equivariant linear operators $\{\mathcal{W}_k\}_{k=1}^{128}$. The 128 linear operators are jointly defined by the stacking of a single squeezing layer breaking invariance under odd translations as explained in Section 6.3.2, followed by a convolutional network with periodic boundary conditions. This network features 10 hidden layers with 64 channels each and uses intermediate LeakyReLU activations. In total, the network has 256 output channels, with each pair of output channels providing the values of $a$ and $b$ in the definition of one of the 128 linear operators. The $a$ output is additionally transformed using a normalized SoftPlus function. We also find it useful to add an $L^2$ regularization loss for both outputs, with a weight of $10^{-5}$.

For the fully joint model, active components of the scalar field are transformed using its frozen components as well as the pseudofermion field. The updated scalar field together with the frozen pseudofermion components are then used to update the active pseudofermion sites.

# C List of figures

# D List of tables

# Bibliography

[1] J. M. Pawlowski and J. M. Urban *Mach. Learn. Sci. Tech.* **1** (2020) 045011, arXiv:1811.03533 [hep-lat].

[2] L. Kades, J. M. Pawlowski, A. Rothkopf, M. Scherzer, J. M. Urban, S. J. Wetzel, N. Wink, and F. P. G. Ziegler *Phys. Rev. D* **102** no. 9, (2020) 096001, arXiv:1905.04305 [physics.comp-ph].

[3] S. Blücher, L. Kades, J. M. Pawlowski, N. Strodthoff, and J. M. Urban *Phys. Rev. D* **101** no. 9, (2020) 094507, arXiv:2003.01504 [hep-lat].

[4] M. S. Albergo, G. Kanwar, S. Racanière, D. J. Rezende, J. M. Urban, D. Boyda, K. Cranmer, D. C. Hackett, and P. E. Shanahan *Phys. Rev. D* **104** no. 11, (2021) 114507, arXiv:2106.05934 [hep-lat].

[5] J. Horak, J. M. Pawlowski, J. Rodríguez-Quintero, J. Turnwald, J. M. Urban, N. Wink, and S. Zafeiropoulos *Phys. Rev. D* **105** no. 3, (2022) 036014, arXiv:2107.13464 [hep-ph].

[6] M. S. Albergo, D. Boyda, K. Cranmer, D. C. Hackett, G. Kanwar, S. Racanière, D. J. Rezende, F. Romero-López, P. E. Shanahan, and J. M. Urban arXiv:2202.11712 [hep-lat].

[7] J. M. Pawlowski and J. M. Urban arXiv:2203.01243 [hep-lat].

[8] C. Gattringer and C. B. Lang, *Quantum chromodynamics on the lattice*, vol. 788. Springer, Berlin, 2010.

[9] I. Montvay and G. Munster, *Quantum fields on a lattice*. Cambridge Monographs on Mathematical Physics. Cambridge University Press, 3, 1997.

[10] H. J. Rothe, *Lattice Gauge Theories : An Introduction (Fourth Edition)*, vol. 43. World Scientific Publishing Company, 2012.

[11] C. Morningstar in *21st Annual Hampton University Graduate Studies Program (HUGS 2006)*. 2, 2007. arXiv:hep-lat/0702020.

[12] USQCD Collaboration, R. C. Brower, A. Hasenfratz, E. T. Neil, S. Catterall, G. Fleming, J. Giedt, E. Rinaldi, D. Schaich, E. Weinberg, and O. Witzel *Eur. Phys. J. A* **55** no. 11, (2019) 198, arXiv:1904.09964 [hep-lat].

[13] USQCD Collaboration, C. Lehner *et al. Eur. Phys. J. A* **55** no. 11, (2019) 195, arXiv:1904.09479 [hep-lat].

[14] USQCD Collaboration, A. S. Kronfeld, D. G. Richards, W. Detmold, R. Gupta, H.-W. Lin, K.-F. Liu, A. S. Meyer, R. Sufian, and S. Syritsyn *Eur. Phys. J. A* **55** no. 11, (2019) 196, arXiv:1904.09931 [hep-lat].

[15] USQCD Collaboration, V. Cirigliano, Z. Davoudi, T. Bhattacharya, T. Izubuchi, P. E. Shanahan, S. Syritsyn, and M. L. Wagman *Eur. Phys. J. A* **55** no. 11, (2019) 197, arXiv:1904.09704 [hep-lat].

[16] USQCD Collaboration, W. Detmold, R. G. Edwards, J. J. Dudek, M. Engelhardt, H.-W. Lin, S. Meinel, K. Orginos, and P. Shanahan *Eur. Phys. J. A* **55** no. 11, (2019) 193, arXiv:1904.09512 [hep-lat].

[17] USQCD Collaboration, A. Bazavov, F. Karsch, S. Mukherjee, and P. Petreczky *Eur. Phys. J. A* **55** no. 11, (2019) 194, arXiv:1904.09951 [hep-lat].

[18] M. Mathur and T. P. Sreeraj *Phys. Rev. D* **94** no. 8, (2016) 085029, arXiv:1604.00315 [hep-lat].

[19] USQCD Collaboration, B. Joó, C. Jung, N. H. Christ, W. Detmold, R. Edwards, M. Savage, and P. Shanahan *Eur. Phys. J. A* **55** no. 11, (2019) 199, arXiv:1904.09725 [hep-lat].

[20] S. P. Meyn and R. L. Tweedie, *Markov chains and stochastic stability*. Springer Science & Business Media, 2012.

[21] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller *J. Chem. Phys.* **21** (1953) 1087–1092.

[22] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth *Phys. Lett. B* **195** (1987) 216–222.

[23] D. H. Weingarten and D. N. Petcher *Phys. Lett. B* **99** (1981) 333–338.

[24] A. D. Kennedy, I. Horvath, and S. Sint *Nucl. Phys. B Proc. Suppl.* **73** (1999) 834–836, arXiv:hep-lat/9809092.

[25] TWQCD Collaboration, K. Ogawa, T.-W. Chiu, and T.-H. Hsieh *PoS* **LAT2009** (2009) 033, arXiv:0911.5532 [hep-lat].

[26] TWQCD Collaboration, Y.-C. Chen and T.-W. Chiu *Phys. Lett. B* **738** (2014) 55–60, arXiv:1403.1683 [hep-lat].

[27] P. de Forcrand and T. Takaishi *Nucl. Phys. B Proc. Suppl.* **53** (1997) 968–970, arXiv:hep-lat/9608093.

[28] P. de Forcrand *Nucl. Phys. B Proc. Suppl.* **73** (1999) 822–824, arXiv:hep-lat/9809145.

[29] M. J. Peardon arXiv:hep-lat/0011080.

[30] M. Hasenbusch and K. Jansen *Nucl. Phys. B Proc. Suppl.* **106** (2002) 1076–1078, arXiv:hep-lat/0110180.

[31] M. A. Clark *PoS* **LAT2006** (2006) 004, arXiv:hep-lat/0610048.

[32] M. A. Clark and A. D. Kennedy *Phys. Rev. Lett.* **98** (2007) 051601, arXiv:hep-lat/0608015.

[33] J. B. Kogut and L. Susskind *Phys. Rev. D* **11** (1975) 395–408.

[34] A. S. Kronfeld *PoS* **LATTICE2007** (2007) 016, arXiv:0711.0699 [hep-lat].

[35] O. Witzel *PoS* **LATTICE2018** (2019) 006, arXiv:1901.08216 [hep-lat].

[36] J. Wambach, B.-J. Schaefer, and M. Wagner *Acta Phys. Polon. Supp.* **3** (2010) 691–700, arXiv:0911.0296 [hep-ph].

[37] R. Percacci *PoS* **ISFTG** (2009) 011, arXiv:0910.5167 [hep-th].

[38] J. S. Schwinger *Phys. Rev.* **128** (1962) 2425–2429.

[39] S. R. Coleman, R. Jackiw, and L. Susskind *Annals Phys.* **93** (1975) 267.

[40] J. Smit and J. C. Vink *Nucl. Phys. B* **303** (1988) 36–56.

[41] H. Dilger *Int. J. Mod. Phys. C* **6** (1995) 123–134, arXiv:hep-lat/9408017.

[42] S. Durr *Phys. Rev. D* **85** (2012) 114503, arXiv:1203.2560 [hep-lat].

[43] D. Albandea, P. Hernández, A. Ramos, and F. Romero-López *Eur. Phys. J. C* **81** no. 10, (2021) 873, arXiv:2106.14234 [hep-lat].

[44] J. Finkenrath arXiv:2201.02216 [hep-lat].

[45] T. Hartung, K. Jansen, F. Y. Kuo, H. Leövey, D. Nuyens, and I. H. Sloan arXiv:2112.05069 [hep-lat].

[46] T. Eichhorn and C. Hoelbling in *38th International Symposium on Lattice Field Theory*. 12, 2021. arXiv:2112.05188 [hep-lat].

[47] L. Funcke, K. Jansen, and S. Kühn *Phys. Rev. D* **101** no. 5, (2020) 054507, arXiv:1908.00551 [hep-lat].

[48] N. Butt, S. Catterall, Y. Meurice, R. Sakai, and J. Unmuth-Yockey *Phys. Rev. D* **101** no. 9, (2020) 094509, arXiv:1911.01285 [hep-lat].

[49] M. C. Bañuls *et al.* *Eur. Phys. J. D* **74** no. 8, (2020) 165, arXiv:1911.00003 [quant-ph].

[50] S. R. Coleman *Annals Phys.* **101** (1976) 239.

[51] A. V. Smilga *Phys. Lett. B* **278** (1992) 371–376.

[52] L. Giusti, G. C. Rossi, M. Testa, and G. Veneziano *Nucl. Phys. B* **628** (2002) 234–252, arXiv:hep-lat/0108009.

[53] G. D'Amico, R. Gobbetti, M. Kleban, and M. Schillo *JCAP* **03** (2013) 004, arXiv:1211.4589 [hep-th].

[54] Y. Shimizu and Y. Kuramashi *Phys. Rev. D* **90** no. 1, (2014) 014508, arXiv:1403.0642 [hep-lat].

[55] Y. Shimizu and Y. Kuramashi *Phys. Rev. D* **90** no. 7, (2014) 074503, arXiv:1408.0897 [hep-lat].

[56] C. Nagele, J. E. Cejudo, T. Byrnes, and M. Kleban *Phys. Rev. D* **99** no. 9, (2019) 094501, arXiv:1811.03096 [hep-th].

[57] C. Nagele, O. Janssen, and M. Kleban arXiv:2010.04803 [quant-ph].

[58] H. Georgi *Phys. Rev. Lett.* **125** no. 18, (2020) 181601, arXiv:2007.15965 [hep-th].

[59] X.-Y. Hu, M. Kleban, and C. Yu *JHEP* **22** (2020) 197, arXiv:2107.04561 [hep-th].

[60] C. R. Gattringer, I. Hip, and C. B. Lang *Nucl. Phys. B* **508** (1997) 329–352, arXiv:hep-lat/9707011.

[61] I. Hip, C. B. Lang, and R. Teppner *Nucl. Phys. B Proc. Suppl.* **63** (1998) 682–684, arXiv:hep-lat/9709030.

[62] C. Czaban and M. Wagner in *31st International Symposium on Lattice Field Theory*. 10, 2013. arXiv:1310.5258 [hep-lat].

[63] K. G. Wilson *Phys. Rev. D* **10** (1974) 2445–2459.

[64] K. G. Wilson in *13th International School of Subnuclear Physics: New Phenomena in Subnuclear Physics*. 11, 1975.

[65] U. Wolff *Nucl. Phys. B Proc. Suppl.* **17** (1990) 93–102.

[66] S. Schaefer, R. Sommer, and F. Virotta *PoS* **LAT2009** (2009) 032, arXiv:0910.1465 [hep-lat].

[67] ALPHA Collaboration, S. Schaefer, R. Sommer, and F. Virotta *Nucl. Phys. B* **845** (2011) 93–119, arXiv:1009.5228 [hep-lat].

[68] J. Hoshen and R. Kopelman *Phys. Rev. B* **14** (1976) 3438–3445.

[69] U. Wolff *Nucl. Phys. B* **300** (1988) 501–516.

[70] R. H. Swendsen and J.-S. Wang *Phys. Rev. Lett.* **58** (1987) 86–88.

[71] R. G. Edwards and A. D. Sokal *Phys. Rev. D* **38** (1988) 2009–2012.

[72] U. Wolff *Phys. Rev. Lett.* **62** (1989) 361.

[73] R. C. Brower and P. Tamayo *Phys. Rev. Lett.* **62** (1989) 1087–1090.

[74] M. Hasenbusch *Nucl. Phys. B* **333** (1990) 581–592.

[75] R. Sinclair *Phys. Rev. D* **45** (1992) 2098–2100.

[76] W. Bietenholz, A. Pochinsky, and U. J. Wiese *Phys. Rev. Lett.* **75** (1995) 4524–4527, `arXiv:hep-lat/9505019`.

[77] H. G. Evertz *Adv. Phys.* **52** (2003) 1, `arXiv:cond-mat/9707221`.

[78] N. Prokof'ev and B. Svistunov *Phys. Rev. Lett.* **87** (2001) 160601, `arXiv:cond-mat/0103146`.

[79] N. Kawashima and K. Harada *Journal of the Physical Society of Japan* **73** (June, 2004) 1379, `arXiv:cond-mat/0312675 [cond-mat.dis-nn]`.

[80] R. Savit *Rev. Mod. Phys.* **52** (1980) 453.

[81] V. Azcoiti, E. Follana, A. Vaquero, and G. Di Carlo *JHEP* **08** (2009) 008, `arXiv:0905.0639 [hep-lat]`.

[82] C. Gattringer, T. Kloiber, and M. Müller-Preussker *Phys. Rev. D* **92** no. 11, (2015) 114508, `arXiv:1508.00681 [hep-lat]`.

[83] E. P. Bernard, W. Krauth, and D. B. Wilson *Phys. Rev. E* **80** (Nov., 2009) 056704, `arXiv:0903.2954 [cond-mat.stat-mech]`.

[84] M. Michel, J. Mayer, and W. Krauth *EPL (Europhysics Letters)* **112** no. 2, (Oct., 2015) 20003, `arXiv:1508.06541 [cond-mat.stat-mech]`.

[85] Y. Nishikawa, M. Michel, W. Krauth, and K. Hukushima *Phys. Rev. E* **92** (Dec., 2015) 063306, `arXiv:1508.05661 [cond-mat.stat-mech]`.

[86] M. Hasenbusch and S. Schaefer *Phys. Rev. D* **98** no. 5, (2018) 054502, `arXiv:1806.11460 [hep-lat]`.

[87] Z. Lei and W. Krauth *EPL* **121** no. 1, (2018) 10008, `arXiv:1711.08375 [cond-mat.stat-mech]`.

[88] G. Aarts *J. Phys. Conf. Ser.* **706** no. 2, (2016) 022004, `arXiv:1512.05145 [hep-lat]`.

[89] P. de Forcrand *PoS* **LAT2009** (2009) 010, arXiv:1005.0539 [hep-lat].

[90] M. Troyer and U.-J. Wiese *Phys. Rev. Lett.* **94** (2005) 170201, arXiv:cond-mat/0408370.

[91] Z. Fodor and S. D. Katz *Phys. Lett. B* **534** (2002) 87–92, arXiv:hep-lat/0104001.

[92] E. Seiler *EPJ Web Conf.* **175** (2018) 01019, arXiv:1708.08254 [hep-lat].

[93] C. E. Berger, L. Rammelmüller, A. C. Loheac, F. Ehmann, J. Braun, and J. E. Drut *Phys. Rept.* **892** (2021) 1–54, arXiv:1907.10183 [cond-mat.quant-gas].

[94] F. Attanasio, B. Jäger, and F. P. G. Ziegler *Eur. Phys. J. A* **56** no. 10, (2020) 251, arXiv:2006.00476 [hep-lat].

[95] F. Attanasio, B. Jäger, and F. P. G. Ziegler arXiv:2203.13144 [hep-lat].

[96] AuroraScience Collaboration, M. Cristoforetti, F. Di Renzo, and L. Scorzato *Phys. Rev. D* **86** (2012) 074506, arXiv:1205.3996 [hep-lat].

[97] A. Alexandru, G. Basar, P. F. Bedaque, G. W. Ridgway, and N. C. Warrington *JHEP* **05** (2016) 053, arXiv:1512.08764 [hep-lat].

[98] A. Alexandru, P. F. Bedaque, H. Lamm, S. Lawrence, and N. C. Warrington *Phys. Rev. Lett.* **121** no. 19, (2018) 191602, arXiv:1808.09799 [hep-lat].

[99] P. de Forcrand, J. Langelage, O. Philipsen, and W. Unger *Phys. Rev. Lett.* **113** no. 15, (2014) 152002, arXiv:1406.4397 [hep-lat].

[100] C. Gattringer *PoS* **LATTICE2013** (2014) 002, arXiv:1401.7788 [hep-lat].

[101] Y. Delgado Mercado, C. Gattringer, and A. Schmidt *Phys. Rev. Lett.* **111** no. 14, (2013) 141601, arXiv:1307.6120 [hep-lat].

[102] F. Bruckmann, C. Gattringer, T. Kloiber, and T. Sulejmanpasic *Phys. Lett. B* **749** (2015) 495–501, arXiv:1507.04253 [hep-lat].

[103] G. Gagliardi and W. Unger *PoS* **LATTICE2018** (2018) 224, arXiv:1811.02817 [hep-lat].

[104] C. Gattringer, D. Göschl, and T. Sulejmanpasic *Nucl. Phys. B* **935** (2018) 344–364, arXiv:1807.07793 [hep-lat].

[105] A. Bazavov *et al.* *Phys. Rev. D* **95** no. 5, (2017) 054504, arXiv:1701.04325 [hep-lat].

[106] HotQCD Collaboration, A. Bazavov *et al.* *Phys. Lett. B* **795** (2019) 15–21, arXiv:1812.08235 [hep-lat].

[107] C. Bonati, M. D'Elia, F. Negro, F. Sanfilippo, and K. Zambello *Phys. Rev. D* **98** no. 5, (2018) 054510, arXiv:1805.02960 [hep-lat].

[108] R. Bellwied, S. Borsanyi, Z. Fodor, J. Günther, S. D. Katz, C. Ratti, and K. K. Szabo *Phys. Lett. B* **751** (2015) 559–564, arXiv:1507.07510 [hep-lat].

[109] S. Borsanyi, Z. Fodor, J. N. Guenther, S. K. Katz, K. K. Szabo, A. Pasztor, I. Portillo, and C. Ratti *JHEP* **10** (2018) 205, arXiv:1805.04445 [hep-lat].

[110] K. Langfeld, B. Lucini, and A. Rago *Phys. Rev. Lett.* **109** (2012) 111601, arXiv:1204.3243 [hep-lat].

[111] K. Langfeld *PoS* **LATTICE2016** (2017) 010, arXiv:1610.09856 [hep-lat].

[112] N. Garron and K. Langfeld *PoS* **LATTICE2016** (2016) 084, arXiv:1611.01378 [hep-lat].

[113] G. Källén. https://www.e-periodica.ch/digbib/view?pid=hpa-001:1952:25::814.

[114] H. Lehmann *Il Nuovo Cimento* **11** no. 4, (Apr., 1954) 342–357.

[115] P. Lowdon *Nucl. Phys. B* **935** (2018) 242–255, arXiv:1711.07569 [hep-th].

[116] P. Lowdon *PoS* **Confinement2018** (2018) 050, arXiv:1811.03037 [hep-th].

[117] A. K. Cyrol, J. M. Pawlowski, A. Rothkopf, and N. Wink *SciPost Phys.* **5** no. 6, (2018) 065, arXiv:1804.00945 [hep-ph].

[118] A. Bonanno, T. Denz, J. M. Pawlowski, and M. Reichert *SciPost Phys.* **12** no. 1, (2022) 001, arXiv:2102.02217 [hep-th].

[119] J. Horak, J. Papavassiliou, J. M. Pawlowski, and N. Wink arXiv:2103.16175 [hep-th].

[120] G. Cuniberti, E. De Micheli, and G. A. Viano *Commun. Math. Phys.* **216** (2001) 59–83, arXiv:cond-mat/0109175.

[121] Y. Burnier, M. Laine, and L. Mether *Eur. Phys. J. C* **71** (2011) 1619, arXiv:1101.5534 [hep-lat].

[122] M. Jarrell and J. E. Gubernatis *Phys. Rept.* **269** (1996) 133–195.

[123] M. Asakawa, T. Hatsuda, and Y. Nakahara *Prog. Part. Nucl. Phys.* **46** (2001) 459–508, arXiv:hep-lat/0011040.

[124] M. Haas, L. Fister, and J. M. Pawlowski *Phys. Rev. D* **90** (2014) 091501, arXiv:1308.4960 [hep-ph].

[125] Y. Burnier and A. Rothkopf *Phys. Rev. Lett.* **111** (2013) 182003, arXiv:1307.6106 [hep-lat].

[126] A. Rothkopf *Phys. Rev. D* **95** no. 5, (2017) 056016, arXiv:1611.00482 [hep-ph].

[127] J. Fei, C.-N. Yeh, and E. Gull *Phys. Rev. Lett.* **126** (Feb, 2021) 056402, arXiv:2010.04572 [cond-mat.str-el].

[128] J. Fei, C.-N. Yeh, D. Zgid, and E. Gull *Phys. Rev. B* **104** (Oct, 2021) 165111, arXiv:2107.00788 [cond-mat.str-el].

[129] D. Binosi and R.-A. Tripolt *Phys. Lett. B* **801** (2020) 135171, arXiv:1904.08172 [hep-ph].

[130] A. F. Falcão, O. Oliveira, and P. J. Silva *Phys. Rev. D* **102** no. 11, (2020) 114518, arXiv:2008.02614 [hep-lat].

[131] M. Ulybyshev, C. Winterowd, and S. Zafeiropoulos *Phys. Rev. B* **96** no. 20, (2017) 205115, arXiv:1707.04212 [cond-mat.str-el].

[132] D. Dudal, O. Oliveira, M. Roelfs, and P. Silva *Nucl. Phys. B* **952** (2020) 114912, arXiv:1901.05348 [hep-lat].

[133] D. Dudal, O. Oliveira, and M. Roelfs *Eur. Phys. J. C* **82** no. 3, (2022) 251, arXiv:2103.11846 [hep-lat].

[134] R. Fournier, L. Wang, O. V. Yazyev, and Q. Wu *Phys. Rev. Lett.* **124** (Feb, 2020) 056401, arXiv:1810.00913 [physics.comp-ph].

[135] H. Yoon, J.-H. Sim, and M. J. Han *Physical Review B* **98** no. 24, (Dec, 2018) 245101, arXiv:1806.03841 [cond-mat.str-el].

[136] L. Kades, J. M. Pawlowski, A. Rothkopf, M. Scherzer, J. M. Urban, S. J. Wetzel, N. Wink, and F. P. G. Ziegler *Phys. Rev. D* **102** no. 9, (2020) 096001, arXiv:1905.04305 [physics.comp-ph].

[137] M. Zhou, F. Gao, J. Chao, Y.-X. Liu, and H. Song *Phys. Rev. D* **104** no. 7, (2021) 076011, arXiv:2106.08168 [hep-ph].

[138] L.-F. Arsenault, R. Neuberg, L. A. Hannah, and A. J. Millis arXiv:1612.04895 [cond-mat.str-el].

[139] S. Offler, G. Aarts, C. Allton, J. Glesaaen, B. Jäger, S. Kim, M. P. Lombardo, S. M. Ryan, and J.-I. Skullerud *PoS* **LATTICE2019** (2019) 076, arXiv:1912.12900 [hep-lat].

[140] M. A. L. Capri, M. S. Guimaraes, I. Justo, L. F. Palhares, and S. P. Sorella *Eur. Phys. J. C* **76** no. 3, (2016) 141, arXiv:1510.07886 [hep-th].

[141] F. Siringo *EPJ Web Conf.* **137** (2017) 13017, arXiv:1606.03769 [hep-ph].

[142] Y. Hayashi and K.-I. Kondo *Phys. Rev. D* **99** no. 7, (2019) 074001, arXiv:1812.03116 [hep-th].

[143] D. Dudal, D. M. van Egmond, M. S. Guimarães, O. Holanda, B. W. Mintz, L. F. Palhares, G. Peruzzo, and S. P. Sorella *Phys. Rev. D* **100** no. 6, (2019) 065009, arXiv:1905.10422 [hep-th].

[144] K.-I. Kondo, Y. Hayashi, R. Matsudo, Y. Suda, and M. Watanabe *PoS* **LC2019** (2019) 053, arXiv:1912.06261 [hep-th].

[145] Y. Hayashi and K.-I. Kondo *Phys. Rev. D* **101** no. 7, (2020) 074044, arXiv:2001.05987 [hep-th].

[146] Y. Hayashi and K.-I. Kondo *Phys. Rev. D* **103** no. 11, (2021) L111504, arXiv:2103.14322 [hep-th].

[147] Y. Hayashi and K.-I. Kondo *Phys. Rev. D* **104** no. 7, (2021) 074024, arXiv:2105.07487 [hep-th].

[148] P. Mehta, M. Bukov, C.-H. Wang, A. G. Day, C. Richardson, C. K. Fisher, and D. J. Schwab *Physics Reports* **810** (May, 2019) 1–124, arXiv:1803.08823 [physics.comp-ph].

[149] J. Carrasquilla and R. G. Melko *Nature Physics* **13** no. 5, (Feb, 2017) 431–434, arXiv:1605.01735 [cond-mat.str-el].

[150] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová *Rev. Mod. Phys.* **91** (Dec, 2019) 045002, arXiv:1903.10563 [physics.comp-ph].

[151] A. Boehnlein *et al.* arXiv:2112.02309 [nucl-th].

[152] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[153] K. Hornik, M. Stinchcombe, and H. White *Neural Networks* **2** no. 5, (1989) 359–366.

[154] A. F. Agarap arXiv:1803.08375 [cs.NE].

[155] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, *Object Recognition with Gradient-Based Learning*, pp. 319–345. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.

[156] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi *Artificial Intelligence Review* **53** no. 8, (Apr, 2020) 5455–5516, arXiv:1901.06032 [cs.CV].

[157] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind *J. Mach. Learn. Res.* **18** no. 1, (Jan, 2017) 5595–5637, arXiv:1502.05767 [cs.SC].

[158] D. P. Kingma and J. Ba arXiv:1412.6980 [cs.LG].

[159] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei arXiv:2005.14165 [cs.CL].

[160] Z. Allen-Zhu, Y. Li, and Y. Liang arXiv:1811.04918 [cs.LG].

[161] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov *Journal of Machine Learning Research* **15** no. 56, (2014) 1929–1958.

[162] G. Zhang, C. Wang, B. Xu, and R. Grosse arXiv:1810.12281 [cs.LG].

[163] E. G. Tabak and E. Vanden-Eijnden *Commun. Math. Sci.* **8** no. 1, (03, 2010) 217–233.

[164] E. G. Tabak and C. V. Turner *Communications on Pure and Applied Mathematics* **66** no. 2, (2013) 145–164.

[165] D. J. Rezende and S. Mohamed arXiv:1505.05770 [stat.ML].

[166] L. Dinh, J. Sohl-Dickstein, and S. Bengio arXiv:1605.08803 [cs.LG].

[167] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan *Journal of Machine Learning Research* **22** no. 57, (2021) 1–64.

[168] L. Tierney *The Annals of Statistics* **22** no. 4, (1994) 1701 – 1728.

[169] M. S. Albergo, G. Kanwar, and P. E. Shanahan *Phys. Rev. D* **100** no. 3, (2019) 034515, arXiv:1904.12072 [hep-lat].

[170] D. C. Hackett, C.-C. Hsieh, M. S. Albergo, D. Boyda, J.-W. Chen, K.-F. Chen, K. Cranmer, G. Kanwar, and P. E. Shanahan arXiv:2107.00734 [hep-lat].

[171] L. Del Debbio, J. M. Rossney, and M. Wilson *Phys. Rev. D* **104** no. 9, (2021) 094507, arXiv:2105.12481 [hep-lat].

[172] M. Caselle, E. Cellini, A. Nada, and M. Panero arXiv:2201.08862 [hep-lat].

[173] A. G. D. G. Matthews, M. Arbel, D. J. Rezende, and A. Doucet arXiv:2201.13117 [stat.ML].

[174] G. Kanwar, M. S. Albergo, D. Boyda, K. Cranmer, D. C. Hackett, S. Racanière, D. J. Rezende, and P. E. Shanahan *Phys. Rev. Lett.* **125** no. 12, (2020) 121601, arXiv:2003.06413 [hep-lat].

[175] D. Boyda, G. Kanwar, S. Racanière, D. J. Rezende, M. S. Albergo, K. Cranmer, D. C. Hackett, and P. E. Shanahan *Phys. Rev. D* **103** no. 7, (2021) 074504, arXiv:2008.05456 [hep-lat].

[176] S. Foreman, X.-Y. Jin, and J. C. Osborn in *9th International Conference on Learning Representations*. 5, 2021. arXiv:2105.03418 [hep-lat].

[177] S. Foreman, X.-Y. Jin, and J. C. Osborn in *38th International Symposium on Lattice Field Theory*. 12, 2021. arXiv:2112.01582 [hep-lat].

[178] S. Foreman, T. Izubuchi, L. Jin, X.-Y. Jin, J. C. Osborn, and A. Tomiya in *38th International Symposium on Lattice Field Theory*. 12, 2021. arXiv:2112.01586 [cs.LG].

[179] S. Lawrence and Y. Yamauchi *Phys. Rev. D* **103** no. 11, (2021) 114509, arXiv:2101.05755 [hep-lat].

[180] M. Rodekamp, E. Berkowitz, C. Gäntgen, S. Krieg, T. Luu, and J. Ostmeyer arXiv:2203.00390 [physics.comp-ph].

[181] M. S. Albergo, D. Boyda, D. C. Hackett, G. Kanwar, K. Cranmer, S. Racanière, D. J. Rezende, and P. E. Shanahan arXiv:2101.08176 [hep-lat].

[182] L. Huang and L. Wang *Phys. Rev. B* **95** (Jan, 2017) 035105, arXiv:1610.02746 [physics.comp-ph].

[183] A. Tanaka and A. Tomiya arXiv:1712.03893 [hep-lat].

[184] D. Wu, L. Wang, and P. Zhang *Phys. Rev. Lett.* **122** (Feb, 2019) 080602, 1809.10606 [cond-mat.stat-mech].

[185] K. A. Nicoli, S. Nakajima, N. Strodthoff, W. Samek, K.-R. Müller, and P. Kessel *Phys. Rev. E* **101** no. 2, (2020) 023304, arXiv:1910.13496 [cond-mat.stat-mech].

[186] K. Nicoli, P. Kessel, N. Strodthoff, W. Samek, K.-R. Müller, and S. Nakajima arXiv:1903.11048 [cond-mat.stat-mech].

[187] L. Wang, Y. Jiang, L. He, and K. Zhou arXiv:2005.04857 [cond-mat.dis-nn].

[188] J. Vielhaben and N. Strodthoff *Phys. Rev. E* **103** no. 6, (2021) 063304, arXiv:2012.10264 [cond-mat.stat-mech].

[189] J. Liu, Y. Qi, Z. Y. Meng, and L. Fu *Phys. Rev. B* **95** (Jan, 2017) 041101, arXiv:1610.03137 [cond-mat.str-el].

[190] J. Liu, H. Shen, Y. Qi, Z. Y. Meng, and L. Fu *Phys. Rev. B* **95** (Jun, 2017) 241104, arXiv:1611.09364 [cond-mat.str-el].

[191] Y. Nagai, H. Shen, Y. Qi, J. Liu, and L. Fu *Phys. Rev. B* **96** (Oct, 2017) 161102, arXiv:1705.06724 [cond-mat.str-el].

[192] H. Shen, J. Liu, and L. Fu *Phys. Rev. B* **97** (May, 2018) 205140, arXiv:1801.01127 [cond-mat.str-el].

[193] C. Chen, X. Y. Xu, J. Liu, G. Batrouni, R. Scalettar, and Z. Y. Meng *Phys. Rev. B* **98** (Jul, 2018) 041102, arXiv:1802.06177 [cond-mat.str-el].

[194] Y. Nagai, M. Okumura, and A. Tanaka *Phys. Rev. B* **101** (Mar, 2020) 115111, arXiv:1807.04955 [cond-mat.str-el].

[195] Y. Nagai, A. Tanaka, and A. Tomiya arXiv:2010.11900 [hep-lat].

[196] Z. Liu, S. P. Rodrigues, and W. Cai arXiv:1710.04987 [cond-mat.dis-nn].

[197] K. Zhou, G. Endrődi, L.-G. Pang, and H. Stöcker *Phys. Rev. D* **100** no. 1, (2019) 011501, arXiv:1810.12879 [hep-lat].

[198] A. Singha, D. Chakrabarti, and V. Arora arXiv:2111.00574 [hep-lat].

[199] S. Kullback and R. A. Leibler *The Annals of Mathematical Statistics* **22** no. 1, (1951) 79 – 86.

[200] D. G. Krige *Journal of the Southern African Institute of Mining and Metallurgy* **52** no. 6, (1951) 119–139.

[201] M. Kanagawa, P. Hennig, D. Sejdinovic, and B. K. Sriperumbudur arXiv:1807.02582 [stat.ML].

[202] H. Liu, Y.-S. Ong, X. Shen, and J. Cai arXiv:1807.01065 [stat.ML].

[203] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[204] M. Krasser. `https://krasserm.github.io/2018/03/19/gaussian-processes/`.

[205] D. Duvenaud. `https://www.cs.toronto.edu/~duvenaud/cookbook/`.

[206] C. A. Micchelli, Y. Xu, and H. Zhang *Journal of Machine Learning Research* **7** no. 95, (2006) 2651–2667.

[207] M. G. Genton *J. Mach. Learn. Res.* **2** (Mar, 2002) 299–312.

[208] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio `arXiv:1406.2661 [stat.ML]`.

[209] J. Gui, Z. Sun, Y. Wen, D. Tao, and J. Ye `arXiv:2001.06937 [cs.LG]`.

[210] M. De Bernardi, M. Khouzani, and P. Malacaria `arXiv:1810.00378 [cs.LG]`.

[211] M. Creutz *Phys. Rev. D* **36** (Jul, 1987) 515–519.

[212] R. Anirudh, J. J. Thiagarajan, B. Kailkhura, and T. Bremer `arXiv:1805.07281 [cs.CV]`.

[213] J. M. Pawlowski, I.-O. Stamatescu, and F. P. G. Ziegler *Phys. Rev. D* **96** no. 11, (2017) 114505, `arXiv:1705.06231 [hep-lat]`.

[214] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, and E. F. and R. Garnett, eds., pp. 8024–8035. Curran Associates, Inc., 2019. `arXiv:1912.01703 [cs.LG]`.

[215] M. Mirza and S. Osindero `arXiv:1411.1784 [cs.LG]`.

[216] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek *PLOS ONE* **10** no. 7, (07, 2015) 1–46.

[217] L. Wang *Phys. Rev. B* **94** (Nov, 2016) 195105, `arXiv:1606.00318 [cond-mat.stat-mech]`.

[218] S. J. Wetzel *Phys. Rev. E* **96** (Aug, 2017) 022140, `arXiv:1703.02435 [cond-mat.stat-mech]`.

[219] W. Hu, R. R. P. Singh, and R. T. Scalettar *Phys. Rev. E* **95** (Jun, 2017) 062122, arXiv:1704.00080 [cond-mat.stat-mech].

[220] S. J. Wetzel and M. Scherzer *Phys. Rev. B* **96** (Nov, 2017) 184410, arXiv:1705.05582 [cond-mat.stat-mech].

[221] P. Broecker, J. Carrasquilla, R. Melko, and S. Trebst *Scientific Reports* **7** (08, 2016) , arXiv:1608.07848 [cond-mat.str-el].

[222] K. Ch'ng, J. Carrasquilla, R. G. Melko, and E. Khatami *Phys. Rev. X* **7** (Aug, 2017) 031038, arXiv:1609.02552 [cond-mat.str-el].

[223] E. P. L. van Nieuwenburg, Y.-H. Liu, and S. D. Huber *Nature Physics* **13** no. 5, (Feb, 2017) 435–439, arXiv:1610.02048 [cond-mat.dis-nn].

[224] W. Rządkowski, N. Defenu, S. Chiacchiera, A. Trombettoni, and G. Bighin *New Journal of Physics* **22** no. 9, (Sep, 2020) 093026, arXiv:1907.05417 [cond-mat.dis-nn].

[225] P. E. Shanahan, D. Trewartha, and W. Detmold *Phys. Rev. D* **97** no. 9, (2018) 094506, arXiv:1801.05784 [hep-lat].

[226] K. Kashiwa, Y. Kikuchi, and A. Tomiya *Progress of Theoretical and Experimental Physics* **2019** no. 8, (08, 2019) , 1812.01522 [cond-mat.dis-nn]. 083A04.

[227] P. Suchsland and S. Wessel *Phys. Rev. B* **97** (May, 2018) 174435, arXiv:1802.09876 [cond-mat.stat-mech].

[228] P. Ponte and R. G. Melko *Phys. Rev. B* **96** (Nov, 2017) 205146, arXiv:1704.05848 [cond-mat.stat-mech].

[229] K. Liu, J. Greitemann, and L. Pollet *Phys. Rev. B* **99** (Mar, 2019) 104410, arXiv:1810.05538 [cond-mat.stat-mech].

[230] C. Casert, T. Vieijra, J. Nys, and J. Ryckebusch *Phys. Rev. E* **99** (Feb, 2019) 023304, arXiv:1807.02468 [cond-mat.stat-mech].

[231] M. Ancona, E. Ceolini, C. Öztireli, and M. Gross in *NIPS Workshop on Interpreting, Explaining and Visualizing Deep Learning - Now What?* ETH Zurich, 2017. arXiv:1711.06104 [cs.LG].

[232] G. Montavon, W. Samek, and K.-R. Müller *Digital Signal Processing* **73** (2018) 1–15, arXiv:1706.07979 [cs.LG].

[233] K. Simonyan, A. Vedaldi, and A. Zisserman in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*, Y. Bengio and Y. LeCun, eds. 2014. arXiv:1312.6034 [cs.CV].

[234] M. Sundararajan, A. Taly, and Q. Yan in *ICML*. 2017. arXiv:1703.01365 [cs.LG].

[235] A. Shrikumar, P. Greenside, and A. Kundaje in *Proceedings of the 34th International Conference on Machine Learning*, D. Precup and Y. W. Teh, eds., vol. 70 of *Proceedings of Machine Learning Research*, pp. 3145–3153. PMLR, International Convention Centre, Sydney, Australia, 06–11 aug, 2017. arXiv:1704.02685 [cs.CV].

[236] M. D. Zeiler and R. Fergus in *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*, D. J. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds., vol. 8689 of *Lecture Notes in Computer Science*, pp. 818–833. Springer, 2014. arXiv:1311.2901 [cs.CV].

[237] K. A. Nicoli, P. Kessel, M. Gastegger, and K. T. Schütt arXiv:1810.09751 [physics.comp-ph].

[238] A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje arXiv:1605.01713 [cs.LG].

[239] L. Breiman *Mach. Learn.* **45** no. 1, (Oct., 2001) 5–32.

[240] C. Andrieu and G. O. Roberts *The Annals of Statistics* **37** no. 2, (2009) 697 – 725, arXiv:0903.5480 [math.ST].

[241] G. Bhanot, U. M. Heller, and I. O. Stamatescu *Phys. Lett. B* **129** (1983) 440–444.

[242] M. Hutchinson *Communications in Statistics - Simulation and Computation* **19** no. 2, (1990) 433–450.

[243] P. de Forcrand and L. Keegan *Phys. Rev. E* **98** no. 4, (2018) 043306, arXiv:1808.01829 [hep-lat].

[244] T. S. Cohen and M. Welling arXiv:1602.07576 [cs.LG].

[245] T. S. Cohen, M. Geiger, J. Koehler, and M. Welling arXiv:1801.10130 [cs.LG].

[246] T. S. Cohen, M. Weiler, B. Kicanaoglu, and M. Welling arXiv:1902.04615 [cs.LG].

[247] A. Tomiya and Y. Nagai arXiv:2103.11965 [hep-lat].

[248] M. Favoni, A. Ipp, D. I. Müller, and D. Schuh *Phys. Rev. Lett.* **128** no. 3, (2022) 032003, arXiv:2012.12901 [hep-lat].

[249] S. Bulusu, M. Favoni, A. Ipp, D. I. Müller, and D. Schuh *Phys. Rev. D* **104** no. 7, (2021) 074504, `arXiv:2103.14686 [hep-lat]`.

[250] K. A. Nicoli, C. J. Anders, L. Funcke, T. Hartung, K. Jansen, P. Kessel, S. Nakajima, and P. Stornati *Phys. Rev. Lett.* **126** no. 3, (2021) 032001, `arXiv:2007.07115 [hep-lat]`.

[251] M. Medvidovic, J. Carrasquilla, L. E. Hayward, and B. Kulchytskyy `arXiv:2012.01442 [cond-mat.dis-nn]`.

[252] P. Deligne, P. Etingof, D. S. Freed, L. C. Jeffrey, D. Kazhdan, J. W. Morgan, D. R. Morrison, and E. Witten, eds., *Quantum fields and strings: A course for mathematicians. Vol. 1, 2.* 1999.

[253] B. Xu, N. Wang, T. Chen, and M. Li `arXiv:1505.00853 [cs.LG]`.

[254] L. Zhang, W. E, and L. Wang `arXiv:1809.10188 [cs.LG]`.

[255] C.-W. Huang, R. T. Q. Chen, C. Tsirigotis, and A. Courville `arXiv:2012.05942 [cs.LG]`.

[256] B. Amos, L. Xu, and J. Z. Kolter `arXiv:1609.07152 [cs.LG]`.

[257] K. Dong, D. Eriksson, H. Nickisch, D. Bindel, and A. G. Wilson `arXiv:1711.03481 [stat.ML]`.

[258] S. Ubaru, J. Chen, and Y. Saad *SIAM Journal on Matrix Analysis and Applications* **38** no. 4, (2017) 1075–1099.

[259] D. P. Kingma and P. Dhariwal in *Advances in Neural Information Processing Systems*, vol. 31. Curran Associates, Inc., 2018. `arXiv:1807.03039 [stat.ML]`.

[260] M. Luscher *Comput. Phys. Commun.* **165** (2005) 199–220, `arXiv:hep-lat/0409106`.

[261] C. Urbach, K. Jansen, A. Shindler, and U. Wenger *Comput. Phys. Commun.* **174** (2006) 87–98, `arXiv:hep-lat/0506011`.

[262] A. Frommer, V. Hannemann, B. Nockel, T. Lippert, and K. Schilling *Int. J. Mod. Phys. C* **5** (1994) 1073–1088, `arXiv:hep-lat/9404013`.

[263] H. Hotelling *The Annals of Mathematical Statistics* **2** no. 3, (1931) 360 – 378.

[264] N. Madras and A. D. Sokal *Journal of Statistical Physics* **50** no. 1-2, (Jan., 1988) 109–186.

[265] D. J. Rezende, G. Papamakarios, S. Racanière, M. S. Albergo, G. Kanwar, P. E. Shanahan, and K. Cranmer `arXiv:2002.02428 [stat.ML]`.

[266] F. Yu and V. Koltun arXiv:1511.07122 [cs.CV].

[267] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios in *Advances in Neural Information Processing Systems*, pp. 7511–7522. 2019. arXiv:1906.04032 [stat.ML].

[268] ALPHA Collaboration, U. Wolff *Comput. Phys. Commun.* **156** (2004) 143–153, arXiv:hep-lat/0306017. [Erratum: Comput.Phys.Commun. 176, 383 (2007)].

[269] S.-H. Li and L. Wang *Phys. Rev. Lett.* **121** (Dec., 2018) 260601, arxiv:1802.02840 [cond-mat.stat-mech].

[270] H.-Y. Hu, S.-H. Li, L. Wang, and Y.-Z. You *Phys. Rev. Res.* **2** no. 2, (2020) 023369, arXiv:1903.00804 [cond-mat.dis-nn].

[271] M. Luscher *Commun. Math. Phys.* **293** (2010) 899–919, arXiv:0907.5491 [hep-lat].

[272] G. P. Engel and S. Schaefer *Comput. Phys. Commun.* **182** (2011) 2107–2114, arXiv:1102.1852 [hep-lat].

[273] M. Gabrié, G. M. Rotskoff, and E. Vanden-Eijnden arXiv:2105.12603 [physics.data-an].

[274] H. Wu, J. Köhler, and F. Noé *Advances in Neural Information Processing Systems* **33** (2020) 5933–5944, arXiv:2002.06707 [stat.ML].

[275] D. Nielsen, P. Jaini, E. Hoogeboom, O. Winther, and M. Welling *Advances in Neural Information Processing Systems* **33** (2020) 12685–12696, arXiv:2007.02731 [cs.LG].

[276] J. Song, S. Zhao, and S. Ermon arXiv:1706.07561 [stat.ML].

[277] D. Levy, M. D. Hoffman, and J. Sohl-Dickstein arXiv:1711.09268 [stat.ML].

[278] H. Avron and S. Toledo *J. ACM* **58** no. 2, (Apr., 2011) 1.

[279] J. Finkenrath, F. Knechtli, and B. Leder *Comput. Phys. Commun.* **184** (2013) 1522–1534, arXiv:1204.1306 [hep-lat].

[280] I. Han, D. Malioutov, and J. Shin arXiv:1503.06394 [cs.DS].

[281] J. Sohl-Dickstein arXiv:2005.06553 [stat.CO].

[282] R. A. Meyer, C. Musco, C. Musco, and D. P. Woodruff arXiv:2010.09649 [cs.DS].

[283] S. Passenheim and E. Hoogeboom arXiv:2012.13311 [cs.LG].

[284] G. Bhanot, S. Black, P. Carter, and R. Salvador *Phys. Lett. B* **183** (1987) 331–336.

[285] G. Bhanot, K. Bitar, and R. Salvador *Phys. Lett. B* **188** (1987) 246–252.

[286] G. Bhanot, K. Bitar, S. Black, P. Carter, and R. Salvador *Phys. Lett. B* **187** (1987) 381.

[287] G. Bhanot, R. Salvador, S. Black, P. Carter, and R. Toral *Phys. Rev. Lett.* **59** (1987) 803.

[288] G. Bhanot, A. Gocksch, and P. Rossi *Phys. Lett. B* **199** (1987) 101–107.

[289] F. Wang and D. P. Landau *Phys. Rev. Lett.* **86** no. 10, (2001) 2050, arXiv:cond-mat/0011174.

[290] Z. Fodor, S. D. Katz, and C. Schmidt *JHEP* **03** (2007) 121, arXiv:hep-lat/0701022.

[291] K. Langfeld and J. M. Pawlowski *Phys. Rev. D* **88** no. 7, (2013) 071502, arXiv:1307.0455 [hep-lat].

[292] K. Langfeld, B. Lucini, R. Pellegrini, and A. Rago *Eur. Phys. J. C* **76** no. 6, (2016) 306, arXiv:1509.08391 [hep-lat].

[293] S. Borsanyi and D. Sexty *Phys. Lett. B* **815** (2021) 136148, arXiv:2101.03383 [hep-lat].

[294] K. Langfeld and B. Lucini *Phys. Rev. D* **90** no. 9, (2014) 094502, arXiv:1404.7187 [hep-lat].

[295] C. Gattringer and P. Törek *Phys. Lett. B* **747** (2015) 545–550, arXiv:1503.04947 [hep-lat].

[296] C. Gattringer, M. Giuliani, A. Lehmann, and P. Törek *POS* **LATTICE2015** (2016) 194, arXiv:1511.07176 [hep-lat].

[297] N. Garron and K. Langfeld *Eur. Phys. J. C* **76** no. 10, (2016) 569, arXiv:1605.02709 [hep-lat].

[298] M. Körner, K. Langfeld, D. Smith, and L. von Smekal *Phys. Rev. D* **102** no. 5, (2020) 054502, arXiv:2006.04607 [hep-lat].

[299] B. Lucini, O. Francesconi, M. Holzmann, D. Lancaster, and A. Rago *J. Phys. Conf. Ser.* **2207** no. 1, (2022) 012052, arXiv:2111.00353 [hep-lat].

[300] K. A. Nicoli, C. J. Anders, L. Funcke, T. Hartung, K. Jansen, P. Kessel, S. Nakajima, and P. Stornati in *38th International Symposium on Lattice Field Theory*. 11, 2021. arXiv:2111.11303 [hep-lat].

[301] T. D. Lee and C.-N. Yang *Phys. Rev.* **87** (1952) 410–419.

[302] F. Attanasio, M. Bauer, L. Kades, and J. M. Pawlowski in *38th International Symposium on Lattice Field Theory*. 11, 2021. arXiv:2111.12645 [hep-lat].

[303] G. Aarts *Phys. Rev. Lett.* **102** (2009) 131601, arXiv:0810.2089 [hep-lat].

[304] L. Bongiovanni, K. Langfeld, B. Lucini, R. Pellegrini, and A. Rago arXiv:1601.02929 [hep-lat].

[305] O. Francesconi, M. Holzmann, B. Lucini, and A. Rago *Phys. Rev. D* **101** no. 1, (2020) 014504, arXiv:1910.11026 [hep-lat].

[306] A. P. Valentine and M. Sambridge *Geophysical Journal International* **220** no. 3, (11, 2019) 1632–1647.

[307] S. Zafeiropoulos, P. Boucaud, F. De Soto, J. Rodríguez-Quintero, and J. Segovia *Phys. Rev. Lett.* **122** no. 16, (2019) 162002, arXiv:1902.08148 [hep-ph].

[308] Z.-F. Cui, J.-L. Zhang, D. Binosi, F. de Soto, C. Mezrag, J. Papavassiliou, C. D. Roberts, J. Rodríguez-Quintero, J. Segovia, and S. Zafeiropoulos *Chin. Phys. C* **44** no. 8, (2020) 083102, arXiv:1912.08232 [hep-ph].

[309] A. K. Cyrol, L. Fister, M. Mitter, J. M. Pawlowski, and N. Strodthoff *Phys. Rev. D* **94** no. 5, (2016) 054005, arXiv:1605.01856 [hep-ph].

[310] A. K. Cyrol, M. Mitter, J. M. Pawlowski, and N. Strodthoff *Phys. Rev. D* **97** no. 5, (2018) 054006, arXiv:1706.06326 [hep-ph].

[311] W.-j. Fu, J. M. Pawlowski, and F. Rennecke *Phys. Rev. D* **101** no. 5, (2020) 054032, arXiv:1909.02991 [hep-ph].

[312] F. Gao and J. M. Pawlowski *Phys. Rev. D* **102** no. 3, (2020) 034027, arXiv:2002.07500 [hep-ph].

[313] F. Gao, J. Papavassiliou, and J. M. Pawlowski *Phys. Rev. D* **103** no. 9, (2021) 094013, arXiv:2102.13053 [hep-ph].

[314] R. Oehme and W. Zimmermann *Phys. Rev. D* **21** (1980) 1661.

[315] R. Oehme *Phys. Lett. B* **252** (1990) 641–646.

[316] W. Menke and R. Creel *Surveys in Geophysics* **42** no. 3, (Apr., 2021) 473–503.

[317] F. D. R. Bonnet, P. O. Bowman, D. B. Leinweber, and A. G. Williams *Phys. Rev. D* **62** (2000) 051501, arXiv:hep-lat/0002020.

[318] A. Sternbeck, E. M. Ilgenfritz, M. Muller-Preussker, and A. Schiller *Phys. Rev. D* **72** (2005) 014507, arXiv:hep-lat/0506007.

[319] P. Boucaud, J. P. Leroy, A. Le Yaouanc, A. Y. Lokhov, J. Micheli, O. Pene, J. Rodriguez-Quintero, and C. Roiesnel *Phys. Rev. D* **72** (2005) 114503, arXiv:hep-lat/0506031.

[320] P. J. Silva and O. Oliveira *Phys. Rev. D* **74** (2006) 034513, arXiv:hep-lat/0511043.

[321] A. Cucchieri, A. Maas, and T. Mendes *Phys. Rev. D* **74** (2006) 014503, arXiv:hep-lat/0605011.

[322] A. Cucchieri, A. Maas, and T. Mendes *Phys. Rev. D* **77** (2008) 094510, arXiv:0803.1798 [hep-lat].

[323] O. Oliveira and P. J. Silva *Phys. Rev. D* **79** (2009) 031501, arXiv:0809.0258 [hep-lat].

[324] I. L. Bogolubsky, E. M. Ilgenfritz, M. Muller-Preussker, and A. Sternbeck *Phys. Lett. B* **676** (2009) 69–73, arXiv:0901.0736 [hep-lat].

[325] T. Iritani, H. Suganuma, and H. Iida *Phys. Rev. D* **80** (2009) 114505, arXiv:0908.1311 [hep-lat].

[326] A. Ayala, A. Bashir, D. Binosi, M. Cristoforetti, and J. Rodriguez-Quintero *Phys. Rev. D* **86** (2012) 074512, arXiv:1208.0795 [hep-ph].

[327] A. Athenodorou, P. Boucaud, F. De Soto, J. Rodríguez-Quintero, and S. Zafeiropoulos *EPJ Web Conf.* **175** (2018) 12012, arXiv:1802.00698 [hep-lat].

[328] A. G. Duarte, O. Oliveira, and P. J. Silva *Phys. Rev. D* **94** no. 7, (2016) 074502, arXiv:1607.03831 [hep-lat].

[329] A. C. Aguilar, F. De Soto, M. N. Ferreira, J. Papavassiliou, J. Rodríguez-Quintero, and S. Zafeiropoulos *Eur. Phys. J. C* **80** no. 2, (2020) 154, arXiv:1912.12086 [hep-ph].

[330] A. C. Aguilar, F. De Soto, M. N. Ferreira, J. Papavassiliou, and J. Rodríguez-Quintero *Phys. Lett. B* **818** (2021) 136352, arXiv:2102.04959 [hep-ph].

[331] A. C. Aguilar, C. O. Ambrósio, F. De Soto, M. N. Ferreira, B. M. Oliveira, J. Papavassiliou, and J. Rodríguez-Quintero *Phys. Rev. D* **104** no. 5, (2021) 054028, arXiv:2107.00768 [hep-ph].

[332] RBC, UKQCD Collaboration, C. Allton *et al.* *Phys. Rev. D* **76** (2007) 014504, arXiv:hep-lat/0701013.

[333] RBC-UKQCD Collaboration, C. Allton *et al.* *Phys. Rev. D* **78** (2008) 114509, arXiv:0804.0473 [hep-lat].

[334] RBC, UKQCD Collaboration, R. Arthur *et al.* *Phys. Rev. D* **87** (2013) 094514, arXiv:1208.4412 [hep-lat].

[335] RBC, UKQCD Collaboration, T. Blum *et al.* *Phys. Rev. D* **93** no. 7, (2016) 074505, arXiv:1411.7017 [hep-lat].

[336] P. A. Boyle, L. Del Debbio, A. Jüttner, A. Khamseh, F. Sanfilippo, and J. T. Tsang *JHEP* **12** (2017) 008, arXiv:1701.02644 [hep-lat].

[337] Y. Iwasaki *Nucl. Phys. B* **258** (1985) 141–156.

[338] D. B. Kaplan *Phys. Lett. B* **288** (1992) 342–347, arXiv:hep-lat/9206013.

[339] Y. Shamir *Nucl. Phys. B* **406** (1993) 90–106, arXiv:hep-lat/9303005.

[340] R. C. Brower, H. Neff, and K. Orginos *Nucl. Phys. B Proc. Suppl.* **140** (2005) 686–688, arXiv:hep-lat/0409118.

[341] A. Sternbeck, K. Maltman, L. von Smekal, A. G. Williams, E. M. Ilgenfritz, and M. Muller-Preussker *PoS* **LATTICE2007** (2007) 256, arXiv:0710.2965 [hep-lat].

[342] P. Boucaud, F. De Soto, J. P. Leroy, A. Le Yaouanc, J. Micheli, O. Pene, and J. Rodriguez-Quintero *Phys. Rev. D* **79** (2009) 014508, arXiv:0811.2059 [hep-ph].

[343] A. Sternbeck, E. M. Ilgenfritz, K. Maltman, M. Muller-Preussker, L. von Smekal, and A. G. Williams *PoS* **LAT2009** (2009) 210, arXiv:1003.1585 [hep-lat].

[344] D. Binosi, C. Mezrag, J. Papavassiliou, C. D. Roberts, and J. Rodriguez-Quintero *Phys. Rev. D* **96** no. 5, (2017) 054026, arXiv:1612.04835 [nucl-th].

[345] P. Boucaud, F. De Soto, K. Raya, J. Rodríguez-Quintero, and S. Zafeiropoulos *Phys. Rev. D* **98** no. 11, (2018) 114515, arXiv:1809.05776 [hep-ph].

[346] D. Binosi and J. Papavassiliou *Phys. Rept.* **479** (2009) 1–152, arXiv:0909.2536 [hep-ph].

[347] M. Q. Huber *Phys. Rept.* **879** (2020) 1–92, arXiv:1808.05227 [hep-ph].

[348] C. S. Fischer *Prog. Part. Nucl. Phys.* **105** (2019) 1–60, arXiv:1810.12938 [hep-ph].

[349] N. Dupuis, L. Canet, A. Eichhorn, W. Metzner, J. M. Pawlowski, M. Tissier, and N. Wschebor *Phys. Rept.* **910** (2021) 1–114, arXiv:2006.04853 [cond-mat.stat-mech].

[350] J. Horak, J. M. Pawlowski, and N. Wink *Phys. Rev. D* **102** (2020) 125016, arXiv:2006.09778 [hep-th].

[351] A. C. Aguilar, D. Binosi, and J. Papavassiliou *Phys. Rev. D* **86** (2012) 014032, arXiv:1204.3868 [hep-ph].

[352] R. Williams, C. S. Fischer, and W. Heupel *Phys. Rev. D* **93** no. 3, (2016) 034026, arXiv:1512.00455 [hep-ph].

[353] M. Q. Huber *Phys. Rev. D* **101** (2020) 114009, arXiv:2003.13703 [hep-ph].

[354] D. Liu and J. Nocedal *Mathematical Programming* **45** no. 1-3, (Aug., 1989) 503–528.

[355] B. Carpenter, A. Gelman, M. D. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. Brubaker, J. Guo, P. Li, and A. Riddell *Journal of Statistical Software* **76** no. 1, (2017) 1–32.

[356] I. C. Cloet and C. D. Roberts *Prog. Part. Nucl. Phys.* **77** (2014) 1–69, arXiv:1310.2651 [nucl-th].

[357] G. Eichmann, H. Sanchis-Alepuz, R. Williams, R. Alkofer, and C. S. Fischer *Prog. Part. Nucl. Phys.* **91** (2016) 1–100, arXiv:1606.09602 [hep-ph].

[358] H. Sanchis-Alepuz and R. Williams *Comput. Phys. Commun.* **232** (2018) 1–21, arXiv:1710.04903 [hep-ph].

[359] N. Christiansen, M. Haas, J. M. Pawlowski, and N. Strodthoff *Phys. Rev. Lett.* **115** no. 11, (2015) 112002, arXiv:1411.7986 [hep-ph].

[360] M. Bluhm *et al. Nucl. Phys. A* **1003** (2020) 122016, arXiv:2001.08831 [nucl-th].

[361] C. S. Ong, A. J. Smola, and R. C. Williamson *Journal of Machine Learning Research* **6** no. 36, (2005) 1043–1071.

[362] A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing arXiv:1511.02222 [cs.LG].

[363] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay *Journal of Machine Learning Research* **12** (2011) 2825–2830.

[364] A. Saabas, https://github.com/andosa/treeinterpreter, 2015.

[365] J. Zhang, T. He, S. Sra, and A. Jadbabaie arXiv:1905.11881 [math.OC].

Erklärung:

Ich versichere, dass ich diese Arbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Heidelberg, den 19. April 2022 ....................................