NEW PATHS IN LATTICE GAUGE THEORIES

By

Giovanni Pederiva

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Physics – Doctor of Philosophy

2022

ABSTRACT

NEW PATHS IN LATTICE GAUGE THEORIES

By

Giovanni Pederiva

The theory of Quantum Chromodynamics (QCD) is a crucial element in our current understanding of the laws that govern the universe. It describes the Strong Interaction, the force that governs the behavior of quarks and gluons and confines them at low energies into hadrons, such as protons, neutrons and pions.

Lattice QCD (LQCD) is a numerical approach that allows non-perturbative studies of the strong interaction at hadronic energy scales, where other theoretical methods fail. It works by discretizing space-time on a lattice of points, fixing the fermion field to the lattice sites and representing the gluon field as the links between them.

Our current numerical approach to Lattice QCD is based upon Markov Chain Monte Carlo (MCMC) methods, from which statistical ensembles of lattice gauge field configurations are generated using a discrete version of the QCD action when evaluating the path-integrals. These calculations require the world's largest supercomputing facilities and possible improvements or new ideas are actively being researched. In this dissertation we present three such attempts.

First, we propose a new strategy to employ the recent advances in Machine Learning into LQCD. In particular, we present a method to use Neural Networks to accelerate the calculations of hadron two-point correlation functions and discuss its applicability.

The second effort we present is part of new collaborative undertaking by the OPEN LATtice Initiative (OPENLAT) to generate new state-of-the-art LQCD ensembles using the recently proposed Stabilized Wilson Fermions (SWF) package, which also includes a modified lattice action. The first results for the light hadron spectrum, obtained by Bayesian model averaging, are presented indicating the excellent scaling behavior of SWF towards the continuum.

Finally, the tentative new approach of Quantum Computing for Lattice Field Theories is

introduced as a possible radical solution to the sign problem. We compare three algorithms for quantum state preparation in the case of the Schwinger model, a toy model for QCD, and discuss their applicability to Noisy Intermediate-Scale Quantum (NISQ) systems.

In loving memory of my mother,
you are missed beyond measure.

ACKNOWLEDGMENTS

This thesis marks the end of my life in Michigan and an incredibly intense part of my life. So many people have helped me along the way, in the good and the bad moments.

First, I want to thank my supervisor Andrea Shindler for the opportunities he has given me, the constant support and the knowledge he shared with me. Working with you all these years has been fantastic.

Then, I want to thank the members of my Ph.D. committee, professors A. Bazavov, D. Lee, C. Piermarocchi and J. Singh, for the guidance they have provided me.

A special thanks go to Matt Rizik, Jack Dragos, Jangho Kim, Leon Hostetler and Johannes Weber for all our interesting discussions at MSU. I want to thank also all my other collaborators, especially the members of the OPENLAT initiative, for accepting me on their team. I also want to acknowledge all the people at FRIB, particularly those in the Nuclear Theory Group.

A big thank you goes to Morten Hjorth-Jensen, for first giving me the opportunity to come to MSU six years ago and then for always supporting me.

I ought to thank all the people I met here in Lansing, who helped me along the way and made my years in Michigan memorable. In particular, I want to thank Giordano, Samuel, Alessandro, Yannis, Nikos, Pely, Artemis, Jordan, David, Johan, Giovanni, Anne and Alida. I will keep the memories of our time together forever with me.

I also must thank all the friends that I left back in Europe, who are also there for me whenever I come to visit; you make it worth every time. I want to mention the amazing group of Gianmarco, Emanuele, Sebastiano, Giuliano, Stefano, Andrea and Giovanni. But also, Annalisa and Clarissa. A very special thank you goes to Alessio, for all the fun reunions through the years.

There are no words to express the gratitude I feel for Hannah. Thank you for being a true friend and always being there for me; I will miss our time together so much.

Thanks also goes to the members of my family, for the unconditional support they give me, regardless of the distance; the most recent ones, Federico and little Jack, for all the fun times spent

together; and to my father, who inspired me in my career and keeps supporting me.

And finally, to you two. Life has taken us far away from one another, but no matter where we are, you mean everything to me. Whenever I look at the stars, I think of you, wishing it was a Summer night in Trento. Whatever happens in the future, I know we will be there for each other.

<div align="right">

Giovanni Pederiva

East Lansing, October 7 2022

</div>

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

KEY TO ABBREVIATIONS

AE	Algorithmic Efficiency
AIC	Akaike Information Criterion
ASP	Adiabatic State Preparation
BiCGSTAB	BiConjugate Gradient Stabilized (Algorithm)
BSM	Beyond Standard Model
CG	Conjugate Gradient (Algorithm)
CP	Charge Parity (Symmetry)
FLAG	Flavour Lattice Averaging Group
FP	Fully Precise
GBF	Gaussian Bayes Factor
GNN	Global Neural Network (Model)
GS	Ground State
HMC	Hybrid Monte Carlo (Algorithm)
HP	High Precision
HPC	High-Performance Computing
ILDG	International Lattice Data Grid
KS	Krylov Subspace
LFT	Lattice Field Theory
LP	Low Precision
LQCD	Lattice Quantum Chromodynamics
MC	Monte Carlo
MCMC	Markov Chain Monte Carlo
MD	Molecular Dynamics
ML	Machine Learning
MLP	Multi-Layer Perceptron
nEDM	Neutron Electric Dipole Moment
NISQ	Noisy Intermediate Scale Quantum
NN	Neural Network
OPENLAT	Open Lattice Initiative
PDF	Probability Distribution Function
PDG	Particle Data Group
pQCD	Perturbative Quantum Chromodynamics
PS	Point-Sink Smeared-Source
QAOA	Quantum Approximate Optimization Algorithm
QC	Quantum Computing
QCD	Quantum Chromodynamics
QED	Quantum Electrodynamics
QFT	Quantum Field Theory
RA	Rodeo Algorithm
ReLU	Rectified Linear Unit

SDG   Stochastic Gradient Descent
SM   Standard Model
SMD   Stochastic Molecular Dynamics (Algorithm)
SPNN   Single Point Neural Network (Model)
SS   Smeared-Sink Smeared-Source
SW   Sheikholeslami-Wohlert
SWF   Stabilized Wilson Fermions
TSM   Truncated Solver Method
WC   Wilson Clover
WCF   Wilson Clover Fermions

CHAPTER 1

INTRODUCTION TO QUANTUM CHROMODYNAMICS AND LATTICE FIELD THEORIES

Quantum Field Theories (QFT) are in theoretical physics the framework, developed over the 20th century, in which Special Relativity and Quantum Mechanics are combined. The current best knowledge of the microscopic world, at the subatomic scale, are best described with QFTs. The general approach is to consider particles as quantized localized excitations of more fundamental objects, the quantum fields. The laws governing the behavior of these fields are described using Lagrangian formalism.

The Standard Model (SM) of Particle Physics is the QFT that describes the interactions of the currently known fundamental particles, quarks and leptons, under three fundamental forces —electromagnetism, the weak and strong force— and the Higgs field. A summary of the particles included in the SM is found in Figure 1.1.
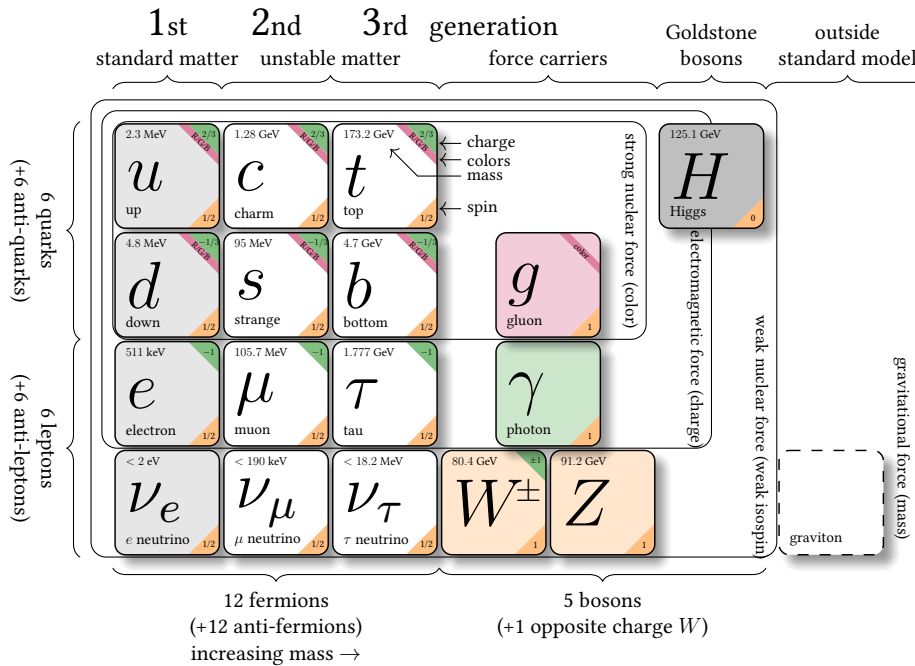
Figure 1.1: Summary of the particles in the Standard Model [1].

From a more technical point of view, the SM is a field theory based on three distinct local gauge symmetry groups, each associated with a fundamental force:

$$\mathrm{SU(3)}_C \times \underbrace{\mathrm{SU(2)}_L \times \mathrm{U(1)}_Y}_{broken\ to\ \ \mathrm{SU(2)}_W \times \mathrm{U(1)}_Q} . \tag{1.1}$$

Through the Higgs mechanism, the $\mathrm{SU(2)}_L \times \mathrm{U(1)}_Y$ symmetries are broken into the $\mathrm{SU(2)}_W$ group associated with the weak force and the $\mathrm{U(1)}_Q$ group associated with the electromagnetic force. However, the piece of the SM that we are interested in this thesis is the $\mathrm{SU(3)}$ group, the one associated with the strong force. In this introductory chapter, we will discuss in the general properties of the theory of Quantum Chromodynamics (QCD) and its prevalent tool for numerical simulations: Lattice Quantum Chromodynamics (LQCD).

Due to the non-linear nature of the strong interaction at low energies, an effect given by its coupling constant α_{S} being large in that regime, the perturbative approaches that are common for the other forces, or the high energy limit of the strong force, cannot be applied. Lattice QCD is instead a non-perturbative approach that has been widely adopted to study QCD effects at the hadronic scale. It is based on formulating the theory on a finite lattice of points in spacetime with spacing usually denoted a, with the fermions residing on the lattice sites while the gluons, the mediators of the strong force, are discretized as the links connecting neighboring sites. The discretization naturally introduces a regulator for the theory, in fact there is a momentum cut-off equal to π/a. A simplified depiction is found in Figure 1.2.

Over the years, since its first formulation by Kenneth Wilson in 1974 [2], many successful numerical calculations have been performed with results covering various research areas such as hadron spectra [3], quark-gluon plasma physics [4], parton distribution functions [5], physics of light nuclei [6] or hadronic matrix elements that can be used for physics for searches of Beyond Standard Model (BSM) physics for example: in the case of calculating CP-violating operators for searches of new physics and the neutron electric dipole moment (nEDM) [7]; $g - 2$, the muon magnetic dipole moment [8]; double-β decay nucleon [9]. In order to obtain results from LQCD, Monte Carlo (MC) methods are employed to generate statistical ensembles of *lattice gauge*
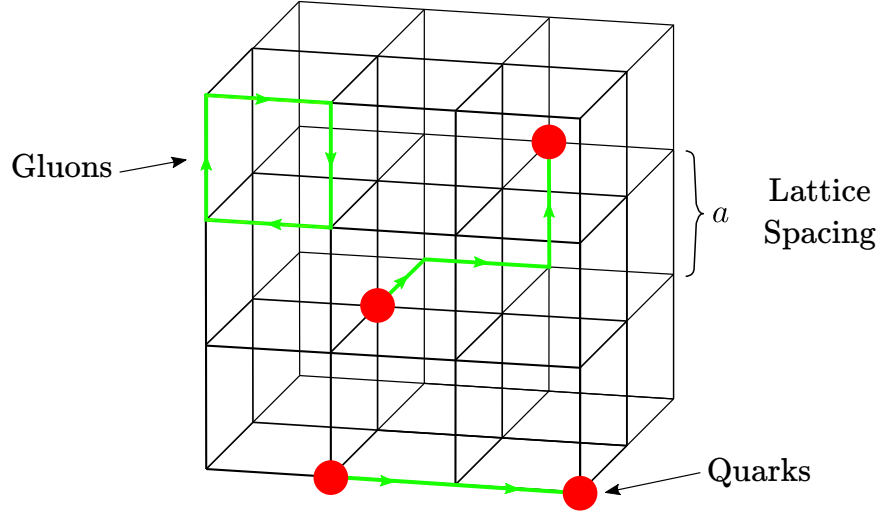
Figure 1.2: Simplified representation of the discretization of QCD on a lattice.

field configurations from which observables are computed. This procedure, however, requires a considerable amount of computational resources; in fact LQCD calculations routinely run on the world's largest supercomputers. Therefore, there is a strong motivation to research possible numerical and algorithmic improvements for LQCD simulations. In this work, we will study some possible paths for improving the state of current lattice QCD calculations.

This thesis is structured as follows: in the rest of this chapter, we will lay the theoretical foundations of QCD and its discretization on the lattice, highlighting the numerical challenges that arise. Then in Chapter 2 we will propose a new method to accelerate the calculation of hadronic two-point functions using Machine Learning (ML). In Chapter 3 a new collaborative effort to generate state-of-the-art lattice gauge field configurations using a set of novel improvements will be introduced. These new ensembles, called Stabilized Wilson Fermions (SWF) utilize a new lattice discretization called Exponential Clover Action. The results of the calculation of the light hadron spectrum of the ensembles will be shown as a first demonstration of their scaling properties. Finally, in Chapter 4 we will present some work related to the broader ongoing effort to formulate Lattice Field Theories (LFT), and lattice QCD in particular, on Quantum Computers. It is known that QC can circumvent the sign problem affecting QCD's MC simulations by dealing with Hamiltonian dynamics directly. However, a formulation of LQCD that is suitable for quantum

hardware is still unknown, and work is being done on simpler models. In particular, we will explore and compare three algorithms for quantum state preparation in the case of the lattice Schwinger Model, i.e. quantum electrodynamics in 1 spatial dimension.

1.1 Quantum Chromodynamics in the Continuum

The theory of Quantum Chromodynamics (QCD) describes the Strong Interaction, also sometimes called the Strong Nuclear Force. It is the theory that governs the motion of quarks and gluons, the subatomic particles that make up the Hadrons, like protons and neutrons. QCD is contained in the Standard Model and is arguably its most complicated part. The reason for this is the phenomenon of confinement. In this section, we will briefly present some of the basic principles and concepts of the theory of QCD.

The origin of QCD lay in the puzzle stemming from the large number of new particles discovered with bubble chambers starting from the 1950s that were called hadrons. Since it was thought that not all of them could be fundamental particles, a model for some constituents (that were later identified with quarks) was needed. First, the eightfold way was proposed by M. Gell-Mann [10] and Y. Ne'eman [11]. Then, after the proposal of a new quantum number for the quarks, called color charge, by O. W. Greenberg [12] and M.Y. Han and Y. Nambu [13], in 1973 H. Fritzsch, M. Gell-Mann and H. Leutwyler [14] formalized the full QCD Lagrangian.

Quantum Chromodynamics is a non-abelian gauge theory based upon the $SU(3)$ symmetry group. This means that all fermions in the theory come in as three-component vectors and have a quantum number associated with the $SU(3)$, commonly referred to as "color charge". As opposed to electromagnetism, where the two charges are identified with positive and negative, in QCD the three resulting charges are identified with the three colors *red* (r), *green* (g) and *blue* (b).

The theory of QCD has been thoroughly tested experimentally; first deep inelastic scattering experiments showed evidence of particles inside of protons and neutrons, then in e^+e^- colliders the first direct evidence of gluons was found. In modern high-energy physics experiments, QCD is a fundamental component in understanding the phenomenology. For example, in Figure 1.3 we see

the perturbative QCD (pQCD) ratio of the $e^+e^- \to$ hadrons cross section to the $e^+e^- \to \mu^+\mu^-$ one, in which the steps, for example at 4 GeV, can be inferred by considering how many active quark flavors are allowed at a given energy (the mass of the bottom quark is $\approx$ 4 GeV) and contribute to the background cross section.
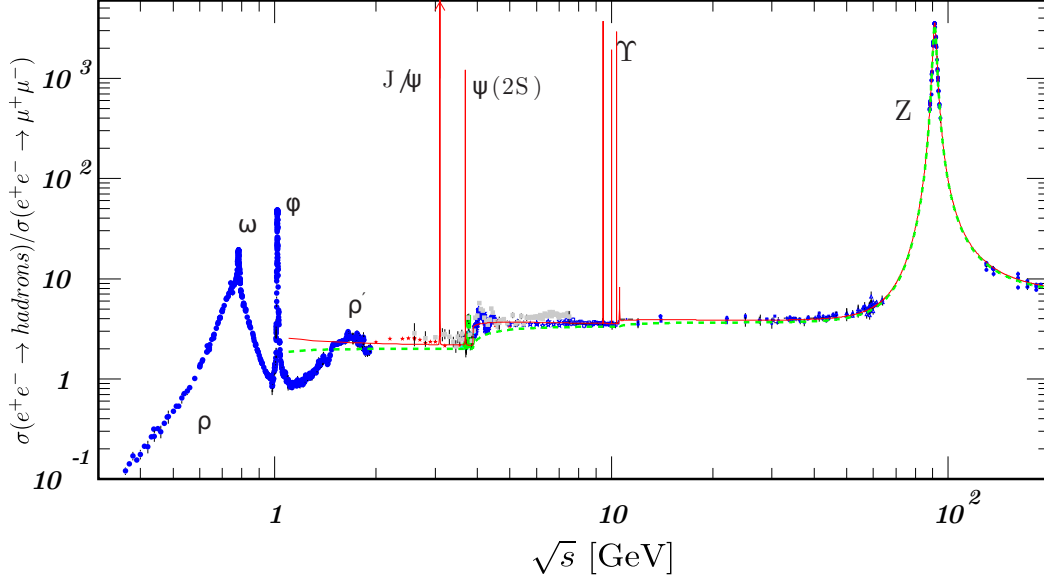
Figure 1.3: Ratio of the experimental cross section of $e^+e^- \to$ hadrons and $e^+e^- \to \mu^+\mu^-$ as collected by the Particle Data Group [15]. The naive quark model is the dashed green line and perturbative QCD (pQCD) results are also shown as a solid red line. pQCD appears to agree well to experiment in non-resonant regions of the energy spectrum.

1.1.1   The QCD Lagrangian

The main equation that characterizes a QFT is its *Lagrangian density*. From it, all the dynamics of the theory can be derived, that is the equations of motion of the free particles and their interactions. The Lagrangian of QCD[1] is the so-called Yang-Mills Lagrangian with $\mathrm{SU}(3)$ as the gauge group. For N_f flavors of quarks it reads:

$$\mathcal{L}_{QCD} = \sum_{f=1}^{N_f} \bar{\psi}_f(i\gamma^\mu D_\mu - m_f)\psi_f - \frac{1}{4}(G_{\mu\nu}^a)^2. \tag{1.2}$$

---

[1] The notation in this section is taken from [16]. Refer to the source for a more complete and formal treatment of the matter.

We will briefly describe what all the symbols in this equation mean, starting with $G^a_{\mu\nu}$: the Gluon Field Strength Tensor. This is defined as

$$G^a_{\mu\nu} = \frac{i}{g_0}[D_\mu, D_\nu] = \partial_\mu A^a_\nu - \partial_\nu A^a_\mu + g_0 f^{abc} A^b_\mu A^c_\nu, \tag{1.3}$$

where μ and ν are Lorentz indices while a is the index that runs over the generators of the $\mathrm{SU}(3)$ symmetry gauge group. Here A^a_μ represents the gluon field, which carries a Lorentz index and a group generator index a, it is a spin-1 boson field. Going further, g_0 represents "bare coupling" of the strong force. f^{abc} are the structure constants of the group $\mathrm{SU}(3)$; they satisfy the conditions:

$$[t^a, t^b] = i f^{abc} t^c, \tag{1.4}$$

where t^a are the generators of the associated Lie algebra $\mathfrak{su}(3)$. The covariant derivative D_μ is given by:

$$D_\mu = \partial_\mu - i g_0 t^a A^a_\mu. \tag{1.5}$$

Quarks are the fermions that enter the Lagrangian as the complex-valued Grassmann fields ψ_f and $\bar{\psi}_f$ with mass m_f. They are formally spin-$\frac{1}{2}$ Dirac spinors that carry color charge. They come in 6 flavors in the SM, denoted here with the index f, and these are u (up), d (down), s (strange), c (charm), b (bottom) and t (top).

Gauge invariance is a crucial concept in QFT and it plays a fundamental role in the process of discretization of QCD that we will see in Section 1.2. We can start understanding it by considering the quark field, which is, in fact, a triplet of fermion fields:

$$\psi(x) = \begin{pmatrix} \psi_r(x) \\ \psi_b(x) \\ \psi_g(x) \end{pmatrix}. \tag{1.6}$$

A local gauge transformation in applied to $\psi(x)$ can be defined as:

$$\psi(x) \to \psi'(x) = \Omega(x)\psi(x) = \exp[i\alpha^a(x)t^a]\psi(x), \quad \text{with } \Omega(x) \in \mathrm{SU}(3). \tag{1.7}$$

The gluon field is introduced in the Lagrangian in such a way that the covariant derivative term is invariant under the same gauge transformation, so:

$$D_\mu \psi(x) \to (D_\mu \psi(x))' = (\partial_\mu - ig_0 A'_\mu(x))\Omega(x)\psi(x) = \Omega(x)(D_\mu \psi(x)), \tag{1.8}$$

from which the transformation for $A_\mu(x)$ is fixed to:

$$A_\mu(x) \to A'_\mu(x) = \Omega(x)\left[A_\mu(x) + \frac{i}{g_0}\partial_\mu\right]\Omega^\dagger(x). \tag{1.9}$$

With this procedure, the whole Lagrangian of QCD is invariant under local gauge symmetry.

From the Lagrangian, we can infer some qualitative features of QCD, for example, the self-interaction of gluons, through the three- and four-gluon vertex that come from the $G^a_{\mu\nu}G^{\mu\nu a}$ term. We can also notice that QCD is flavor diagonal, meaning that all interactions do not change the flavor of the quarks.

1.1.2 General Properties of QCD

The study of Quantum Chromodynamics and its properties is a very active field of research. Some key features of the theory are shared with other non-abelian field theories, for example, the self-interaction of the gauge bosons. Other prominent features instead arise when trying to renormalize the theory, in particular when the running of the coupling is considered. At low energies, the theory shows signs of *confinement*, while at high energies, it is *asymptotically free.*

Running Coupling   When renormalizing QCD, one has to consider the Renormalization Group Equation (RGE) for the bare parameters of the theory, i.e. the coupling g_{bare} and the quark masses m^f_{bare}. These equations determine the rate at which the renormalized coupling g and the masses vary as a function of some renormalization scale μ. The resulting differential equation in the case of the coupling is called the beta function $\beta(g)$:

$$\beta(g) = \frac{d}{d\log(\mu)}g(\mu). \tag{1.10}$$

The result is the same for any non-abelian theory $\mathrm{SU}(N)$. One can expand $\beta(g)$ in powers of the coupling as:

$$\beta(g) = b_0 g^3 + b_1 g^5 + b_2 g^7 + \dots \tag{1.11}$$

and then integrate up to an arbitrary order. This results in a specific expression for the dependency of the coupling to the renormalization scale. The different coefficients in Equation (1.11) are obtained by calculating the contributions of higher order diagrams and are, in general, renormalization scheme dependent, starting from b_0 which can be derived by the 1-loop corrections:

$$b_0 = -\frac{1}{(4\pi)^2}\left(\frac{11}{3}N - \frac{2}{3}N_f\right). \tag{1.12}$$

Here N is the size of the symmetry group $\mathrm{SU}(N)$ and N_f is the number of fermion species in the theory. By inspecting the sign of Equation (1.12) one finds that any non-abelian theory where the number of fermions is less than $\frac{11}{2}N$ (in the actual case of $\mathrm{SU}(3)$ this is ≤ 16) has vanishing coupling in the high energy limit, so we say it is "asymptotically free".

If one performs the integration of Equation (1.10) analytically at fixed order in g by substituting Equation (1.12) into Equation (1.10) and integrating by parts, one obtains the following (expressed here in terms of $\alpha_\mathrm{S} = g^2/4\pi$, the analog of the famous fine-structure constant α_{EM} of electromagnetism):

$$\alpha_\mathrm{S}(\mu) = \frac{\alpha_\mathrm{S}^0}{1 + \frac{b_0 \alpha_\mathrm{S}^0}{4\pi}\log(\mu^2/M^2)} = \frac{4\pi}{b_0 \log(\mu^2/\Lambda_{\mathrm{QCD}}^2)}, \tag{1.13}$$

where α_S^0 is the value of the coupling at some fixed energy $\bar{\mu}$ and stems from the the integration. A typical choice for $\bar{\mu}$ is the mass of the Z boson, because we can obtain very precise experimental data about it. It is also a suitable choice as it is a sufficiently large scale where perturbation theory for QCD can be safely applied. The constant Λ_{QCD} represents the typical energy at which the interactions become non-perturbative. We can see this because as $\mu \to \Lambda_{\mathrm{QCD}}$ the coupling diverges. This can be computed from Lattice QCD [17] and estimated experimentally; in the $\overline{MS}$ scheme, experiments suggest that its value is $\approx 200 - 300$ MeV.
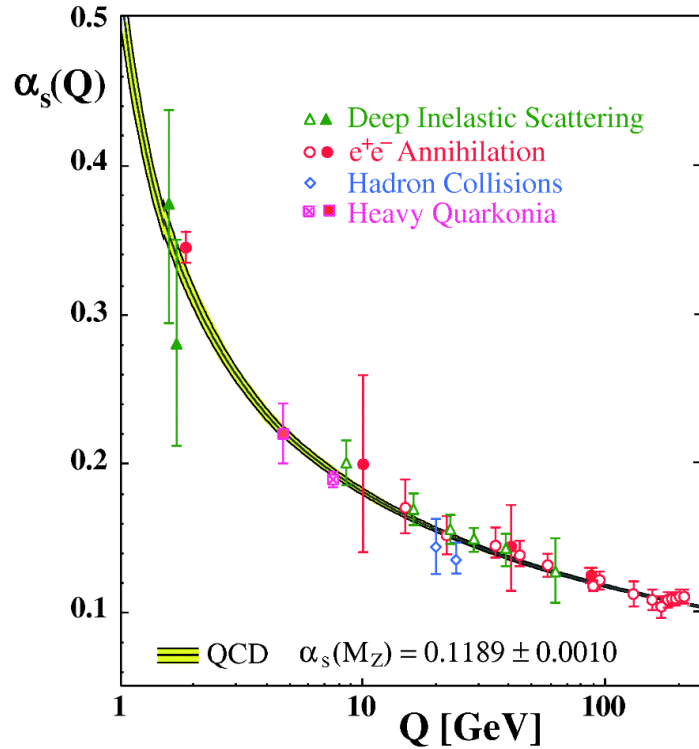
Figure 1.4: The strong coupling as a function of the energy scale, The data points are experimental results; the solid black line and yellow bands represent the QCD prediction using the reported value of the coupling at the mass of the Z boson (α_S^0 in our notation). Image taken from [18].

Asymptotic Freedom   From Equation (1.13) it is clear that the coupling $\alpha_S(\mu)$ at high energies decreases. This implies that as the energy of quarks increases, they are less coupled to the gluon field and the QCD contributions to interaction processes become small. One of the main arguments in support of the search for Grand Unification Theories (GUT) [15] is the fact that at some energy scale, the coupling of the strong interaction becomes comparable to that of the electroweak interaction.

At the time when the theory of QCD had just been proposed, the discovery of asymptotic freedom, made by Gross, Wilczek and Politzer [19, 20] —who also won the Nobel Prize in 2004 for this study— was a stepping stone in affirming that QCD was indeed the correct theory for strong interactions.

Asymptotic freedom permits perturbative expansions in powers of the coupling for sufficiently large energies. It is crucial, for example, at the scales of particle colliders to describe events before

9

quarks and gluons hadronize.

Confinement   The running of the coupling, when going towards low energies, as seen in Equation (1.13) implies that $\alpha_S \sim 1$ for energies on the order of Λ_{QCD}. In this regime, known as confinement, all quarks and gluons cannot appear as free particles and instead form color-neutral bound states. These color-singlets are predominantly of two types: mesons (pairs of quark anti-quark) and baryons (triplets of quarks), but more exotic states, such as tetraquarks, pentaquarks and glueballs are all in principle allowed by QCD and are being studied both theoretically and experimentally.

Quarks are said to be confined because at low energies they can only be found in bound states. A single quark would, within a very small time scale, *hadronize*, which means they spawn new quark/anti-quark pairs from the vacuum to balance the color charge and reform colorless bound states quickly. A typical example is to consider two quarks inside a meson and try to pull them apart. Since the coupling increases for large distances, while being pulled apart more and more energy would be stored in the vacuum space between them, eventually spawning a new quark/anti-quark pair. This is pictorially shown in Figure 1.5. The behavior of the strong force at low energies also explains the large mass difference between quarks and hadrons. In these bound states, most of the mass is, in fact, energy coming from the dynamics of the gluon field, not from the constituent quarks. For example, the proton mass is 938 MeV, while its valence quarks only account for 10 MeV.

Confinement is what fundamentally distinguishes the strong interaction from the other two forces of the SM. It also prevents perturbative methods from being applied in the study of hadrons. This is the reason why non-perturbative methods, such as Lattice QCD are required to fully understand hadronic process.

1.2   Quantum Chromodynamics on a Lattice

The idea of LQCD was introduced by K.G. Wilson in 1974 [2] as a method to study confinement. The main idea is to directly evaluate path integrals of the QCD action, in Euclidean space, by
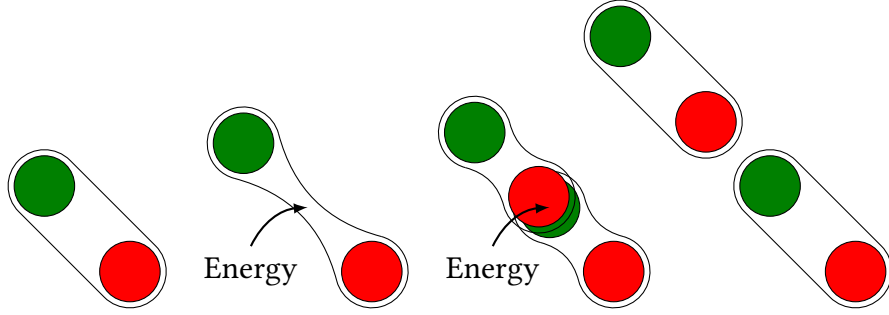
Figure 1.5: Pictorial representation of confinement. As two color sources are pulled apart, the energy stored in the color field between the sources increases so much that a $q\bar{q}$ pair is formed from the vacuum energy.

discretizing spacetime on a lattice. Over time, LQCD has been successfully used to perform calculations and is now considered the most systematic approach for non-perturbative calculations. It should be noted that LFTs are not only used for studies of QCD, but also for other QFTs.

In this section, we will briefly describe how the discretization procedure works, the fundamental tools that are needed and some general results that show how successful this approach can be. In Section 1.4, we will instead look at some of the numerical details and challenges that are open in the field.

1.2.1 Discretization of the QCD Lagrangian

Observables in Lattice QCD are computed using Feynman's path-integrals, not in Minkovski space, but rather in Euclidean space by performing a Wick rotation. This is done to remove the oscillatory nature of the exponential of the action and instead get a real "Boltzmann-like factor". In practice, for the expectation value of an observable O, one needs to evaluate[2]:

$$\langle O \rangle = \frac{1}{Z} \int \mathcal{D}[\psi]\mathcal{D}[\bar{\psi}]\mathcal{D}[A] \, O e^{-S_{QCD}}, \tag{1.14}$$

where Z is the QCD partition function $Z = \int \mathcal{D}[\psi]\mathcal{D}[\bar{\psi}]\mathcal{D}[A]e^{-S_{QCD}}$ and S_{QCD} is the euclidean version of the QCD action, or more simply: $S_{QCD} = \int d^4x \mathcal{L}_{QCD}$, note that here also the Lagrangian is Wick rotated to Euclidean space.

---

[2]The formal treatment of Lattice QCD and the notation is primarily based upon [21].

Evaluating path integrals is an impossible task, as they include a sum over all possible variations of a classical path. However, one can fix the number of degrees of freedom by introducing a finite lattice of points at which to evaluate the integral, effectively reducing the path-integral to a product of integrals over the field's value at the lattice sites. Since we are dealing with four-dimensional Euclidean space, the most straightforward choice is to fix a hyper-cubic lattice Λ:

$$\Lambda = \{n = (n_x, n_y, n_z, n_t)\} \text{ where } n_x, n_y, n_z = 1, 2, \ldots, N; \ n_t = 1, 2, \ldots, N_T. \quad (1.15)$$

Here N is the number of points in the spatial dimensions and N_T is the number of points in the temporal one (we will see later why these two are set differently and with typically $N_T > N$), hence the volume is set as $N^3 \times N_T$. A crucial parameter in simulations is the *lattice spacing*, denoted simply as a.

The discretization procedure has two main benefits: it makes numerical evaluations of path-integrals possible and introduces a momentum cutoff as a regulator for the theory. To connect the lattice results to the physical world, one has to perform the continuum limit $a \to 0$ and the infinite volume limit $aN, aN_T \to \infty$. In order to choose proper values for the volume and the lattice spacing, one needs to consider both a fine enough value for a such that it resolves the hadronic scale (so $a \ll \Lambda_{QCD}$) and a total volume larger than the typical scale of the lightest particles active in the interactions, so for hadronic processes one needs to consider the Compton wavelength of pions, for example, m_π^{-1}.

The Wilson Gauge Action To start the formal discretization procedure of QCD on the lattice, we begin with the fermionic part of the action. Since fermions can only be evaluated at fixed sites, we must use finite difference to perform derivatives, in particular we consider $\partial_\mu \psi(x) \to \frac{\psi(n+\hat{\mu})-\psi(n-\hat{\mu})}{2a}$. With this, the fermion action for a fixed point becomes:

$$S_F[\psi, \bar{\psi}] = \int d^4x \, \bar{\psi}(x)(\gamma_\mu \partial^\mu + m)\psi(x) \quad (1.16)$$

$$\to a^4 \sum_{n \in \Lambda} \bar{\psi}(n) \left[\sum_{\mu=1}^{4} \gamma_\mu \frac{\psi(n+\hat{\mu}) - \psi(n-\hat{\mu})}{2a} + m\psi(n) \right].$$

In order to preserve local gauge invariance, as in the continuum case, an extra degree of freedom must be introduced. In fact, it is clear to see that under a local gauge transformation $\Omega(n)$

$$\bar{\psi}(n)\psi(n \pm \hat{\mu}) \rightarrow \bar{\psi}(n)\Omega^{\dagger}(n)\Omega(n \pm \hat{\mu})\psi(n \pm \hat{\mu}). \tag{1.17}$$

is not gauge-invariant. The problem can be solved by introducing a field $U(n)$ that cancels the term $\Omega^{\dagger}(n)\Omega(n \pm \hat{\mu})$. This is what we identify as the lattice version of the gauge fields; by convention it is written as $U(n, n + \hat{\mu}) = U_{\mu}(n)$ and transforms as:

$$U_{\mu}(n) \rightarrow U'_{\mu}(n) = \Omega(n)U_{\mu}(n)\Omega^{\dagger}(n + \hat{\mu}) \tag{1.18}$$

$$U_{-\mu}(n) \rightarrow U'_{-\mu}(n) = \Omega(n)U_{-\mu}(n)\Omega^{\dagger}(n - \hat{\mu}).$$

The field $U_{\mu}(n)$ is a collection of "links" from one lattice site to the neighboring ones and has the fundamental property that it is orientable $U(n, n - \hat{\mu}) \equiv U_{-\mu}(n) = U_{\mu}(n - \hat{\mu})^{\dagger}$. Its values are elements of $\mathrm{SU}(3)$. A pictorial representation is given in Figure 1.6, With this new insight, the

Figure 1.6: Schematic representation of the link variables $U_{\mu}(n)$ and $U_{-\mu}(n)$ on the lattice.

gauge-invariant lattice fermion action is:

$$S_F[\psi, \bar{\psi}] = a^4 \sum_{n \in \Lambda} \bar{\psi}(n) \left[\sum_{\mu=1}^{4} \gamma_{\mu} \frac{U_{\mu}(n)\psi(n + \hat{\mu}) - U_{-\mu}(n)\psi(n - \hat{\mu})}{2a} + m\psi(n) \right]. \tag{1.19}$$

The link field U_{μ} is related to the path-ordered product ($\mathcal{P}$) of the gauge field $A_{\mu}(x)$ along some curve:

$$G(x, y) = \mathcal{P} \exp\left(ig_0 \int_{\mathcal{C}} A_{\mu}(x')dx'^{\mu} \right). \tag{1.20}$$

This quantity is the so-called "gauge transporter" [16] in the continuum. It has the property of connecting in a gauge covariant manner two points in spacetime. Hence it is clear to see the analogy with the lattice link variables. From Equation (1.20) it is easy to see that for small lattice

spacing a one can expand:

$$G(n, n + \hat{\mu}) \equiv U_\mu(n) = \exp(iaA_\mu(n)) = \mathbb{1} + iaA_\mu(n) + \mathcal{O}(a^2) \tag{1.21}$$

$$G(n, n - \hat{\mu}) \equiv U_{-\mu}(n) = \exp(-iaA_\mu(n)) = \mathbb{1} - iaA_\mu(n - \hat{\mu}) + \mathcal{O}(a^2).$$

The gauge transporter is naturally gauge-invariant when evaluated on closed loops, and this property also applies to the discretized version of the theory. In particular, one can use the minimal loop, the so-called "Plaquette" (see Figure 1.7) as a building block for a discretized version of the gluonic part of the QCD action. Therefore, one can show, using Equation (1.21) that:

$$P_{\mu\nu}(n) = U_\mu(n)U_\nu(n + \hat{\mu})U_{-\mu}(n + \hat{\mu} + \hat{\nu})U_{-\nu}(n + \hat{\nu}) \tag{1.22}$$

$$= U_\mu(n)U_\nu(n + \hat{\mu})U_\mu^\dagger(n + \hat{\nu})U_\nu^\dagger(n)$$

$$\approx \exp\left[ia^2 G_{\mu\nu}(n) + \mathcal{O}(a^3)\right].$$
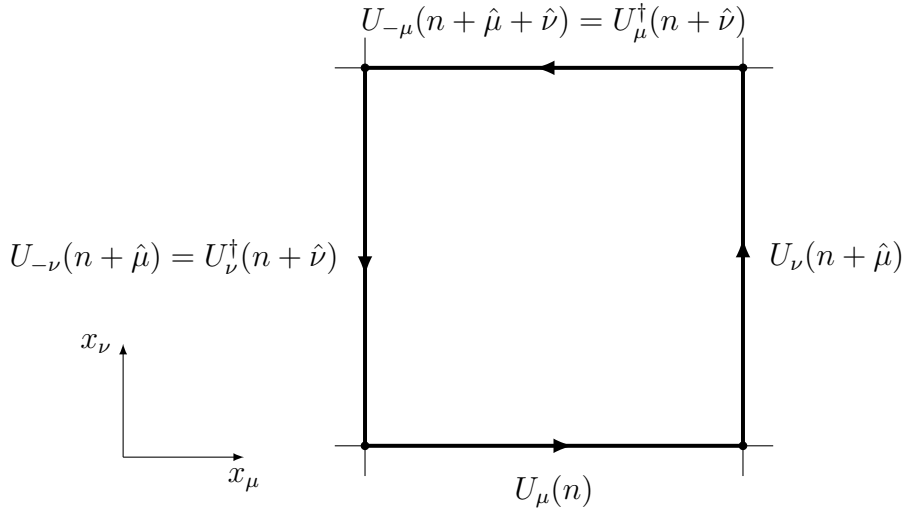
Figure 1.7: The Plaquette, as defined on the lattice, is the oriented product of the link variables of a minimal square in the plane $\mu\nu$.

It is easy then to use the result of Equation (1.22) to construct the gauge part of the action:

$$S_G = \frac{1}{4g^2} \int d^4x\, G_{\mu\nu}(x)^2 \rightarrow \frac{a^4}{2g^2} \sum_{n \in \Lambda} \sum_{\mu\nu} \mathrm{Tr}(G_{\mu\nu}(n)^2) = \frac{2}{g^2} \sum_{n \in \Lambda} \sum_{\mu < \nu} \mathrm{Re}\left[\mathrm{Tr}\left(\mathbb{1} - P_{\mu\nu}(n)\right)\right].$$

$$\tag{1.23}$$

14

Equation (1.23) is the so-called Wilson Gauge Action and is the simplest possible discretization for the Yang-Mills action on a lattice.

1.2.2 Lattice Fermions

The discretization of the fermion action given by Equation (1.19) is incomplete, as it contains unphysical results. To better understand this, one needs to look at the Fourier transform of the propagator. Introducing the lattice Dirac matrix $D_{xy}[U]$ between two lattice sites x and y:

$$S_F[\psi, \bar{\psi}] = a^4 \sum_{n \in \Lambda} \bar{\psi}(x) D_{xy}[U] \psi(y), \tag{1.24}$$

where

$$D_{xy}[U] = \sum_{\mu=1}^{4} \gamma_\mu \frac{U_\mu(x)\delta_{x,(y-\hat{\mu})} - U_{-\mu}^\dagger(x)\delta_{x,(y+\hat{\mu})}}{2a} + m\delta_{x,y}. \tag{1.25}$$

Its Fourier transform is:

$$\tilde{D}_{pq} = \delta(p-q)\tilde{D}(p) \quad \text{where} \quad \tilde{D}(p) = m\mathbb{1} + \frac{i}{a}\sum_{\mu=1}^{4} \gamma_\mu \sin(p_\mu a). \tag{1.26}$$

The inverse of $\tilde{D}_{pq}$ is related to the Dirac propagator and results in:

$$\tilde{D}(p)^{-1} = \frac{-ia\sum_\mu \gamma_\mu \sin(p_\mu a) + m}{\sum_\mu \sin^2(p_\mu a) + m^2}. \tag{1.27}$$

Going back to position space, via an inverse Fourier transform, we obtain an expression for the *quark propagator*:

$$D^{-1}(x;y) = \frac{1}{|\Lambda|}\sum_{p \in \tilde{\Lambda}} \tilde{D}(p)^{-1} e^{ip(x-y)a}. \tag{1.28}$$

Equation (1.27) has the expected pole in the limit of $m \to 0$, but also 15 other unphysical poles residing at the corners of the first Brillouin Zone (any value of p_μ where all of its components are either 0 or $\frac{\pi}{a}$). This is referred to as the "fermion doublers" problem, and a solution was already proposed by Wilson in his first paper on discretizing QCD on a lattice [2]. First, it should be noted that the problem vanishes in the continuum, when $a \to 0$, so this is a pure discretization effect. The solution is then to add additional terms to the fermion action in such a way that they cancel

the doublers without changing the theory in the continuum. The simplest such term is the Wilson term, which modifies $\tilde{D}(p)$ as:

$$\tilde{D}^W(p) = m\mathbb{1} + \frac{i}{a}\sum_{\mu=1}^{4}\gamma_\mu \sin(p_\mu a) + \mathbb{1}\frac{1}{a}\sum_{\mu=1}^{4}(1 - \cos(p_\mu a)). \tag{1.29}$$

This leaves the pole at $m = 0$ intact, but it is clear that the additional cosine piece vanishes in the continuum. In coordinate space, this becomes:

$$D_{xy}^W[U] = \frac{1}{2a}\sum_{\mu=\pm1}^{\pm4}(\mathbb{1} - \gamma_\mu)U_\mu(x)\delta_{x,(y-\hat{\mu})} + \left(m + \frac{4}{a}\right)\delta_{x,y}, \tag{1.30}$$

with the handy notation of $\gamma_{-\mu} = -\gamma_\mu$. With this definition, we can then write the final form of the Wilson fermion action:

$$S^W[\psi,\bar{\psi}] = a^4\sum_{n\in\Lambda}\bar{\psi}(x)D_{xy}^W[U]\psi(y). \tag{1.31}$$

The Wilson action, by construction, is correct up to orders of $\mathcal{O}(a)$. One can, in principle, design other terms that systematically improve this discretization by adding higher-dimensional terms that cancel exactly the $\mathcal{O}(a)$ errors. This is known as the Symanzik improvement program [22] and for the Wilson action the first order correcting term is the so-called "Clover Term", computed first by B. Sheikholeslami and R. Wohlert [23]:

$$S^{\text{SW}} = S^W - \frac{1}{2}c_{\text{SW}}a^5\sum_{n\in\Lambda}\bar{\psi}(n)\sigma_{\mu\nu}G_{\mu\nu}(n)\psi(n). \tag{1.32}$$

The Clover Term is called this way because of the way the gluon field tensor is discretized, which resembles a clover leaf, see Figure 1.8. More precisely, the $\mathcal{O}(a^2)$ improved definition of $G_{\mu\nu}$ on the lattice is:

$$G_{\mu\nu} = \frac{1}{8}\left[Q_{\mu\nu} - Q_{\nu\mu}\right] \tag{1.33}$$

with

$$Q_{\mu\nu} = P_{\mu\nu}(n) + P_{\nu-\mu}(n) + P_{-\nu\mu}(\hat{n}) + P_{-\mu-\nu}(n). \tag{1.34}$$

In Chapter 3 we will see a modified version of the clover term, part of the Stabilized Wilson Fermions (SWF) package.
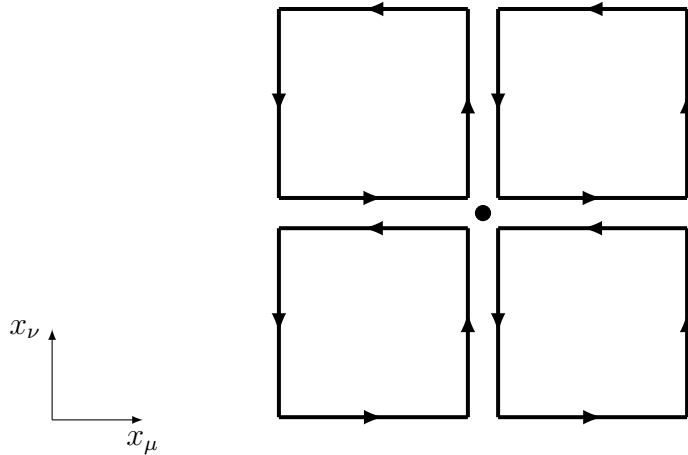
Figure 1.8: Schematic representation of the symmetric definition of the clover term $Q_{\mu\nu}(n)$ on the lattice. The field strength tensor for a plane $\mu\nu$ at a lattice site is given by the average of all the for plaquettes that start from a site n, all taken with the same orientation.

1.2.3 Different Types of Lattice Actions

The Wilson-Clover action seen in Equation (1.32) is only one of the possible choices for a lattice fermion action. As discussed in the previous section, one is free to add any term to the Dirac matrix as long as it vanishes in the continuum. Over the years, there has been active research in designing different actions, each with its own benefits and downsides. Here we will briefly mention the most important classes.

- **Wilson-Clover Fermions:** these are the ones we already discussed. It is the most straight-forward way to remove the fermion doublers problem, however, it explicitly breaks chiral symmetry. Still, clover fermions have been extensively used in LQCD calculations over the years. Some of the field's most remarkable results, such as the light hadron spectrum, which is in complete agreement with experimental data, have been obtained with this discretization, see Figure 1.9.

- **Ginsparg-Wilson Fermions:** this discretization is obtained by imposing the requirement of preserving chiral symmetry exactly on the lattice through the Ginsparg-Wilson equation [24]. However, this is done at the expense of the locality of the action. In practice, the two main types of actions of this class, Domain-wall [25–28] and overlap fermions [29, 30], have a

much higher numerical cost compared to other discretizations, but are still extensively used due to their properties.

- **Staggered Fermions:** these don't remove the fermion doublers explicitly, but instead distribute their values on multiple lattice sites, leading to 4 "tastes" (an unphysical analogous of flavor) that have to be removed afterwards. Introduced first by J. Kogut and L. Susskind [31]. Staggered fermions are widely used because of their relatively cheap computational cost and practicality, especially in finite-temperature QCD [32]. In Chapter 4 we will apply this discretization to the Schwinger Model.

- **Twisted-Mass Fermions:** these are very similar to Wilson fermions, but have an additional rotation in flavor space (the twist) that partially preserves chiral symmetry [33–35]. However, this procedure breaks isospin symmetry instead.
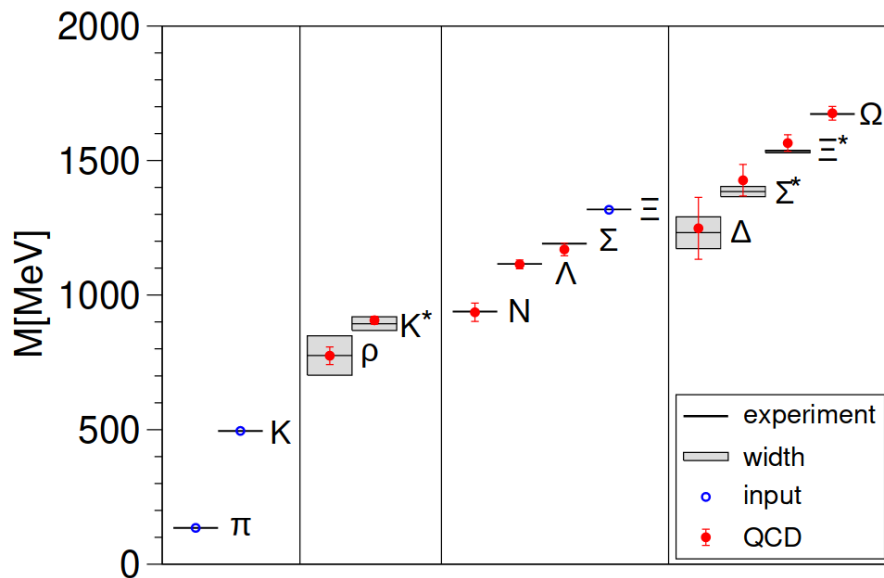
Figure 1.9: Light Hadron Masses as calculated from the Budapest-Marseille-Wuppertal (BMW) collaboration [3] using Wilson-Clover fermions. The gray lines and bands represent the experimental data, the blue circles are the input used to fix the scale, while the LQCD calculation results are the red points.

1.3 Numerical Evaluation of Path Integrals on the Lattice QCD

When performing numerical calculations of QCD on a lattice, as shown in the previous section, for a given observable O, one numerically has to evaluate the path integral:

$$\langle O \rangle = \frac{\int \mathcal{D}\psi \mathcal{D}\bar{\psi} \mathcal{D}U \, O \, e^{-S_E[\psi,\bar{\psi},U]}}{\int \mathcal{D}\psi \mathcal{D}\bar{\psi} \mathcal{D}U e^{-S_E[\psi,\bar{\psi},U]}}. \tag{1.35}$$

This is done by first selecting a discretized lattice action:

$$S_E[\psi,\bar{\psi},U] = S_G[U] + S_F[\psi,\bar{\psi},U] = S_G[U] + \bar{\psi}M[U]\psi. \tag{1.36}$$

In the path integral, at every point, there are three fields to consider, U, ψ, $\bar{\psi}$. However, the fermion fields can be integrated exactly using the properties of Grassmann numbers, in particular, for any given matrix K one has:

$$\int \mathcal{D}\Theta^* \mathcal{D}\Theta e^{-\Theta^* K \Theta} = \left(\prod_i \int d\theta_i^* d\theta_i \right) e^{-\theta_i^* K_{ij} \theta_j} = \left(\prod_i \int d\theta_i^* d\theta_i \right) e^{-\sum_i \theta_i^* k_i \theta_i} \tag{1.37}$$

$$= \prod_i k_i = \det K.$$

This simplifies the form of the denominator of Equation (1.35), the partition function, which becomes:

$$Z = \int \mathcal{D}\psi \mathcal{D}\bar{\psi} \mathcal{D}U e^{-S[\psi,\bar{\psi},U]} = \int \mathcal{D}U e^{-S_G[U]} \det M[U]. \tag{1.38}$$

For the numerator, if O does not depend on the fermion fields, we can apply the same result, but if it depends on a fermion bilinear, a more general form of Equation (1.37) is needed:

$$\int \mathcal{D}\Theta^* \mathcal{D}\Theta \theta_a^* \theta_b e^{-\Theta^* K \Theta} = \left(\prod_i \int d\theta_i^* d\theta_i \right) \theta_a^* \theta_b e^{-\theta_i^* K_{ij} \theta_j} = \left(\prod_i \int d\theta_i^* d\theta_i \right) \theta_a^* \theta_b e^{-\sum_i \theta_i^* k_i \theta_i}$$

$$\tag{1.39}$$

$$= (\det K)(K^{-1})_{ab},$$

which results in:

$$\langle O[\psi(n),\bar{\psi}(m),U] \rangle = \frac{1}{Z} \int \mathcal{D}U \, O[M[U]^{-1},U] e^{-S_G[U]} (\det M[U]). \tag{1.40}$$

If multiple fermions appear in the observable, those need to be contracted using Wick's theorem. However, their dependence still only enters as the inverse of the Dirac matrix $M[U]$, which in turn only depends on the gauge field.

The form of Equation (1.40) cannot be simplified analytically any further. However, it is very suitable for numerical integration using Monte Carlo methods. In fact one can identify the term $P[U] = \frac{e^{-S_G[U]} \det M}{Z}$ as a Probability Distribution Function (PDF), notice that it only depends on the gauge field U, and draw random samples from it in what we call an *ensemble* $\mathcal{U} = \{U_1, U_2, \ldots, U_N\}$. The expectation value of the observable then becomes the ensemble average of O computed on the *gauge field configurations* of $\mathcal{U}$:

$$\langle O \rangle = \int \mathcal{D}U \, P[U] O[\psi, \bar{\psi}, U] \approx \frac{1}{N} \sum_{U_i \in \mathcal{U}} O(\psi, \bar{\psi}, U_i). \tag{1.41}$$

The problem is now reduced to finding a suitable procedure for drawing gauge field configurations from the PDF. The most widely used method is via Markov Chain Monte Carlo methods (MCMC) like the Metropolis-Hastings, Heat Bath or Overrelaxation algorithms. In practice, today, the Hybrid Monte Carlo algorithm, which we will describe in Section 1.3.1, is the most widely used one. Recently more novel approaches have been proposed, for example, using flow-based machine-learning algorithms [36, 37].

1.3.1   The Hybrid Monte Carlo Algorithm

The Hybrid Monte Carlo (HMC) algorithm was first proposed for Lattice QCD calculations in 1987 by S. Duane et al. [38] and it is now utilized in broader contexts, such as machine learning, with the different name, but fortunately with the same acronym, of Hamiltonian Monte Carlo. The key idea is to combine microcanonical updates, performed with Hamiltonian dynamics, and a Metropolis accept-reject condition to maintain ergodicity.

The microcanonical part of the algorithm, often times called the Molecular Dynamics (MD) update for the analogy with other Hamiltonian dynamics simulations, is based on the observation that for a given bosonic field Q one can introduce a real field P that it is its conjugate momentum.

An observable that depends on Q can then be calculated as:

$$\langle O \rangle_Q = \frac{\int \mathcal{D}Q \, O[Q] \, e^{-S[Q]}}{\int \mathcal{D}Q \, e^{-S[Q]}} = \frac{\int \mathcal{D}Q \, O[Q] \, e^{-\frac{1}{2}P^2 - S[Q]}}{\int \mathcal{D}Q \, e^{-\frac{1}{2}P^2 - S[Q]}} = \langle O \rangle_{Q,P} \tag{1.42}$$

The expectation value does not change as the Gaussian integrals for P cancel. This formulation represents a non-relativistic microcanonical ensemble whose Hamiltonian is $H[Q, P] = \frac{1}{2}P^2 + S[Q]$ and its equation of motion are:

$$\dot{P} = -\frac{\partial H}{\partial Q} = -\frac{\partial S}{\partial Q}, \tag{1.43}$$
$$\dot{Q} = \frac{\partial H}{\partial P} = P.$$

One can utilize these equations to move from one lattice gauge configuration to another along the Markov Chain. To perform the Hamiltonian evolution, one uses regular algorithms developed for MD, such as the Leapfrog algorithm, which is widely used for its reversibility properties. Since the MD update carries errors coming from the numerical integration, usually of $\mathcal{O}(\epsilon^2)$ where ϵ is the time-step size, to enforce ergodicity one performs a Metropolis accept-reject check. So the transition probability between two configurations before and after the MD update is:

$$T(P', Q'|P, Q) = \min\left(1, \frac{\exp(-H[P', Q'])}{\exp(-H[P, Q])}\right) = \min\left(1, \ \exp(-\Delta H)\right). \tag{1.44}$$

This can be shown to satisfy detailed balance, hence making it suitable for MCMC calculations.

1.3.2 Hadron Correlators on the Lattice

In order to study the properties of hadrons on the lattice, like their mass, one needs to find a suitable way to construct operators out of quark fields that match the desired hadron's quantum numbers. In practice, the calculation of *hadron correlators* is what is performed on the lattice. These are two-point functions that are usually interpreted via spectral decomposition:

$$C(t) = a^3 \sum_{\mathbf{x}} \langle O(t, \mathbf{x}) \bar{O}(0, \mathbf{0}) \rangle = \sum_k \langle 0| \, \hat{O} \, |k\rangle \langle k| \, \hat{O}^\dagger \, |0\rangle e^{-tE_k}, \tag{1.45}$$

where $O(t)$ is what is called an interpolating operator. It is a combination of quark fields with the same quantum numbers of the hadron under consideration, but also of all its excited states.

These cannot be distinguished on the lattice, so all of them propagate at one. For this reason, one needs to consider the time dependence of the two-point function, as their propagation is different because of the e^{-tE_k} factor, which makes heavy excited states decay very quickly in signal on the lattice. In contrast, the stable ground state is easily recovered in the $t \to \infty$ case. We will see more of the details later in this section.

In this work, we will extensively use these interpolating operators:

$$\text{Pion: } O_{\pi^+}(x) = \bar{d}(x)_{\alpha,c}(\gamma_5)_{\alpha\beta}u(x)_{\beta,c}$$

$$\text{Kaon: } O_{K^+}(x) = \bar{s}(x)_{\alpha,c}(\gamma_5)_{\alpha\beta}u(x)_{\beta,c}$$

$$\text{Proton: } O_P(x) = \epsilon_{abc}u(x)_{\alpha,a}\left(u(x)^T_{\beta,b}C(\gamma_5)_{\beta\gamma}d(x)_{\gamma,c}\right)$$

$$\text{Omega: } O_\Omega(x) = \epsilon_{abc}us(x)_{\alpha,a}\left(s(x)^T_{\beta,b}C(\gamma_5)_{\beta\gamma}s(x)_{\gamma,c}\right). \tag{1.46}$$

The correlator is then the contraction between $O(x)$ and $\bar{O}(y)$. For pions, for example, it is:

$$\langle O_{\pi^+}(x)\bar{O}_{\pi^+}(y)\rangle =$$
$$= \langle \bar{d}(x)_{\alpha_1,c_1}(\gamma_5)_{\alpha_1\beta_1}u(x)_{\beta_1,c_1}\bar{u}(y)_{\alpha_2,c_2}(\gamma_5)_{\alpha_2\beta_2}d(y)_{\beta_2,c_2}\rangle$$
$$= -\langle(\gamma_5)_{\alpha_1\beta_1}\bar{d}(x)_{\alpha_1,c_1}d(y)_{\beta_2,c_2}(\gamma_5)_{\alpha_2\beta_2}\bar{u}(y)_{\alpha_2,c_2}u(x)_{\beta_1,c_1}\rangle$$
$$= -\langle\text{Tr}\left[\gamma_5 D_u^{-1}(y,x)\gamma_5 D_d^{-1}(x,y)\right]\rangle_G \tag{1.47}$$

for the proton and the Ω baryon the contraction is slightly more involved, as there are more possible contractions of the quarks of identical flavor. In this case, Wick's theorem is used to compute all terms that contribute to the correlator while keeping track of the relative signs coming from the anti-commutation relations, but it is diagrammatically shown in Figure 1.10.
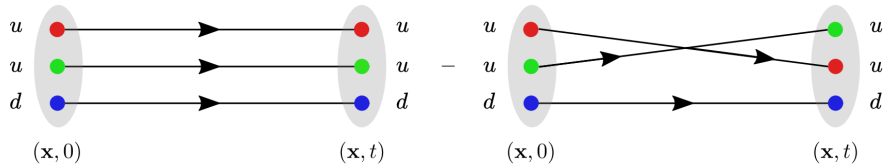
Figure 1.10: Schematic representation of the possible quark contractions for a proton interpolating operator.

The term, for a generic quark flavor q, that appears in Equations (1.46) and (1.47) is, when contracted, the equivalent of the inverse Dirac operator between the two points where the quarks fields are located:

$$\langle q(x)_{\alpha,a} \bar{q}(y)_{\beta,b} \rangle = D^{-1}(y,x)_{ab}^{\alpha\beta}. \tag{1.48}$$

From Equation (1.47) we see that in order to calculate the expectation value of the quark correlator, one needs the expectation value, computed over a gauge ensemble, of the trace of the appropriate (depending on the hadron) quark propagators $D^{-1}(y,x)$. In principle, this requires the inversion of the whole Dirac matrix, but this is not feasible numerically due to the large size of the matrix. In practice, the problem of determining the quark propagator is reformulated in terms of a *linear system* by fixing a *point source*:

$$\eta_{\beta,b}(0) = \delta_{b,c_1} \delta_{\beta,\alpha_1} \delta_{y,(\mathbf{y},0)}. \tag{1.49}$$

This source is a simple Kronecker delta in space, Dirac and color indices. The problem is then reformulated as:

$$D(0,x)_{ab}^{\alpha\beta} q(x)_{\alpha,a} = \eta_{\beta,b}(0).$$

This is equivalent to computing only one column of the inverse matrix. Since we are still interested in all possible combinations of Dirac and color indices, in practice, a total of $3 \times 4 = 12$ different source vectors are to be solved. The exact form of the Dirac operator matrix depends on the lattice action; in the case of the Wilson-Clover action, it is a very sparse matrix because fermions at one lattice site are only linked to their nearest neighbors.

This kind of linear system is usually solved using iterative methods. One of the simplest ones is the Conjugate Gradient, but many variations are used. For example, the BiCGSTAB [39] is a common choice for non-hermitian operators. These iterative solvers are terminated at convergence: when for a small parameter ϵ:

$$||Ax_n - b|| < \epsilon$$

Recently, there has been much research on faster and more efficient algorithms, such as the SAP [40] and Multi-Grid [41] algorithms. In Chapter 2 we will discuss more about the details of

the quark propagator solvers, as we will try to find a machine learning method to speed up the calculations.

1.3.3 Correlation Functions and Effective Masses

One can extract properties of hadrons propagating on the lattice from the correlation function of Equation (1.45). An obvious first choice would be that of the mass. In Figure 1.11 we can see the typical behavior of $C(t)$ for the pion and the nucleon.
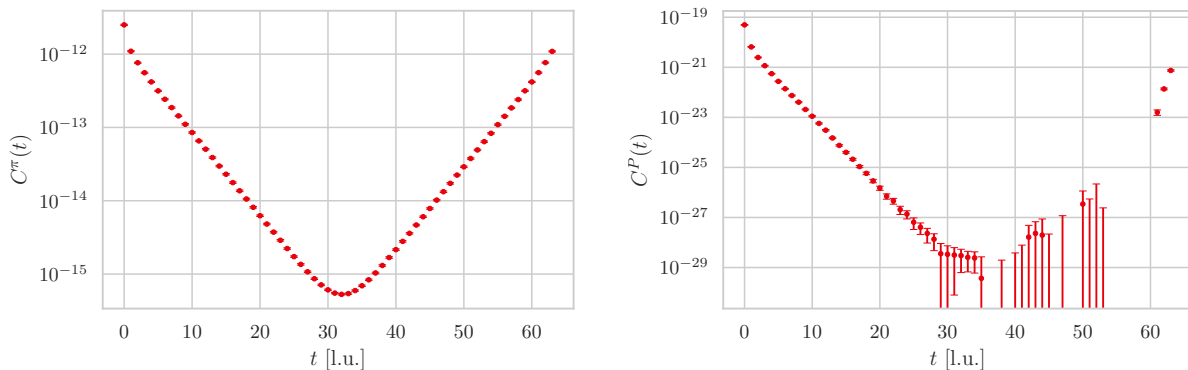
Figure 1.11: Example of pion (left) and proton (right) correlators from an ensemble of the PACS-CS collaboration [42]. The lattice size is $32^3 \times 64$, the lattice spacing is $a = 0.0907(13)$ while the pion mass $m_\pi = 567.6(3)$, on the x-axis the time is reported in lattice units, as standard practice.

As expected, there is an exponential decay in time, both for positive and negative values around the point $t = 0$ where the source is located; this behavior is given by the periodic boundary conditions. The asymmetric shape of the baryon correlators is due to the mass difference of the positive and negative parity states, which both propagate from the same interpolating operator. The degradation of the signal of the nucleon correlator is due to the Signal-to-Noise problem, which we will discuss more in Section 1.4.2.

The simplest possible way to extract hadron masses is to look at the "effective mass". It is easily obtained from taking the logarithm of Equation (1.45):

$$m_{\text{eff}}(t) = \ln \left[\frac{C(t)}{C(t+1)} \right]. \tag{1.50}$$

24

In the limit of $t \to \infty$ the function $m_{\text{eff}}(t)$ has a plateau whose value is the lowest energy of the spectrum of the hadron interpolating operator. One can then fit a constant to this plateau to get an estimate of the mass (see Figure 1.12). In Chapter 2 we will use this procedure, while in Chapter 3, a more involved analysis for the determination of hadron masses will be presented.
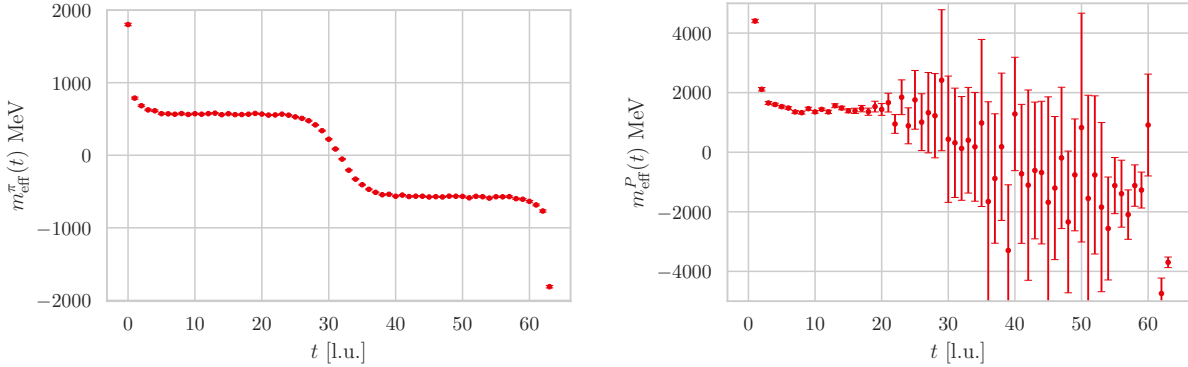
Figure 1.12: Example of pion (left) and proton (right) effective mass, computed following Equation (1.50) for the correlators in Figure 1.11 from an ensemble of the PACS-CS collaboration [42]. The lattice size is $32^3 \times 64$, the lattice spacing is $a = 0.0907(13)$ while the pion mass $m_\pi = 567.6(3)$.

1.3.4 Smearing Techniques

A common technique for improving the correlators' signal is to use gauge link *smearing*. Since, most of the time, one is interested only in the ground state of the hadron state, one is free to modify the interpolating operator freely to increase its overlap with the ground state. Many different smearing procedures have been proposed, for example APE [43], HYP [44] or Stout [45] smearing. In this work, we employ the Gaussian gauge-invariant smearing technique introduced by S. Güsken [46]. The starting ansatz is that the hadron operator can be constructed from the fermion field $\psi(x,t)$ smeared with an appropriate wave-function $F(x,x')$:

$$\tilde{\psi}(x,t) = \sum_{x'} F(x,x')\psi(x,t) = \sum_{x'} \left[\delta_{x,x'} + \alpha H(x,x') \right] \psi(x,t) \tag{1.51}$$

where $H(x, x')$ is the hopping matrix:

$$H(x, x') = \sum_{\mu=1}^{3} U_\mu(x)\delta_{x',x+\hat{\mu}} + U_\mu^\dagger(x - \hat{\mu})\delta_{x',x-\hat{\mu}} \tag{1.52}$$

the parameter α represents the strength of the coupling between neighboring links, hence the smoothness of the smearing. Equation (1.51) is one step of the smearing procedure, in practice, there are N_{smear} applications of that same equation. The scheme then is characterized by two parameters $(N_{\text{smear}}, \alpha)$. When applying this procedure to quark propagators, the algorithm is as follows:

1. Create a smeared source at z and smear it with N_{smear} applications of F: $Q(z) = F_{\text{smear}}^N(z, 0)$

2. Solve the modified linear system:

$$\sum_{y,t} M(z, t; y, t')(\tilde{q})^{-1}(y, t'; 0) = Q(z)\delta_{t,0}, \tag{1.53}$$

with

$$(\tilde{q})^{-1}(y, t'; 0) = \sum_z q^{-1}(y, t'; z, 0)F(z, 0), \tag{1.54}$$

3. Finally, multiply the smeared propagator with the wave function F again:

$$\sum_z q^{-1}(y, t'; 0) = \sum_y (F^N)^\dagger(\tilde{q})^{-1}(y, t; 0) \tag{1.55}$$

.

One thing to note is that the last step is optional, in which case the "sink" is point-like and not smeared. The propagator computed this way is not positive definite, but can be useful for methods such as the variational method [47, 48] or matrix-Prony methods [49, 50]. Throughout this work, we will label the possible configuration of point or smeared sinks with the obvious notation of PS (point sink, smeared source) and SS (smeared sink, smeared source).

1.4   Modern Lattice QCD

Lattice QCD is now a mature field with well-established practices regarding the generation of configurations, the measurement of observables and their analysis. In this section, we will

first outline how simulations in LQCD are performed and later highlight some of the numerical challenges that are currently affecting them.

1.4.1 Structure of LQCD Calculations

Current LQCD simulations are primarily performed with dynamical fermions, meaning that the PDF used for generating the Markov Chain for the gauge field ensemble includes the fermion action (as opposed to "quenched simulations"). The light quarks are usually considered to be degenerate, while all other flavors have to be simulated separately. The usual labeling nowadays is that of $2 + 1$ for simulations with degenerate u, d quarks and the s quark, and $2 + 1 + 1$ for simulations that also add the c quark.

For numerical reasons and in order to correctly estimate the systematic uncertainties, LQCD calculations are not performed on a single ensemble of gauge field configurations. In fact, we have already seen, in Section 1.2.2, that the lattice action is only correct up to a fixed order in the lattice spacing a and contains artifacts that need to be removed to recover the continuum theory. So one needs several ensembles at different values of a and perform the continuum extrapolation ($a \to 0$). Another extrapolation that is usually performed is that of the infinite volume ($L \to \infty$) to make sure that the systematics introduced by the finite size of the simulation are accounted for.

In Section 1.3 we also have noticed that the condition number of the Dirac matrix approaches zero for smaller values of the quark mass. This makes calculation at the physical value numerically expensive. A way to circumvent this problem is to measure observables at different values of the pion mass m_π and later extrapolate to the physical point. This is known as the "chiral extrapolation". Recently some calculations with physical m_π have become accessible due to improvements in computing hardware.

Since generating gauge field configurations is computationally very expensive, it is standard practice to save them to disk space, effectively separating the running of the MCMC algorithm from the measuring of observables into two distinct tasks. Here is a rough sketch of how the calculation is then performed:

1. Generate with MCMC methods a set of gauge field ensembles with different physical characteristics (lattice spacing, volume, quark masses), sampling from a PDF determined by choice of the lattice action. As already mentioned, this is numerically challenging and requires large High Performance Computing (HPC) facilities, often with several computing nodes at a time, because of the high cost of the MD updates. Therefore, several collaborations generate ensembles and share their configurations among their members or make them publicly available. In Chapter 3, we will see a new effort of this type, the Open Lattice Initiative (OPENLAT).

2. Since most of the observables require computing quark propagators, performing measurements is also a relatively expensive task from the numerical point of view. In fact, solving the linear systems is challenging because of the sheer size of the Dirac matrix and the poor convergence properties of linear solvers, especially when dealing with light quark masses. The measurement of observables, however, can be be performed on different hardware than the the one for MCMC algorithms, in particular today these are mostly performed on GPU nodes at HPC facilities, with usually only very few nodes, while the parallelization is done at the configuration level (running multiple small jobs, one per configuration, at once).

3. Once the observables are computed over a set of configurations, expectation values are evaluated, often using resampling methods such as bootstrap [51] or jackknife, see Appendix 1A.2 for more details, and the extrapolations to infinite volume, continuum and physical quark mass are performed. All of this procedure is done to obtain reliable estimates of the statistical and systematic uncertainties. In doing this, also the autocorrelation of the MCMC needs to be taken into account, see Appendix 1A.1

4. In order to compare results coming from LQCD to experimental values, one needs to determine a physical scale for the lattice. In fact, all quantities computed on the lattice are expressed in terms of the lattice spacing a, so a physical quantity has to be used to set its value to the proper units. A similar procedure has to be performed for the quark masses as

well. In general, one needs $1 + N_f$ physical quantities to fix the scale.

Scale setting is a research field on its own and many different possibilities have been explored, such as using the static quark potential (from which the Sommer scale r_0 is obtained) [52], heavy baryons like the Ω [53], or the gradient flow [54]; the latter is the one that will be used throughout this work, more details can be found in Appendix 1C.

1.4.2   Current Numerical Challenges in LQCD

The approach presented in the previous section has been very successful, however, there are some limitations to the current approach that prevent, for example, large-scale simulations at the physical point with small lattice spacing. In this section, we will briefly present some of them.

Critical Slowing Down   An important quantity in MCMC algorithms is the autocorrelation time. It is a measure of how much the samples of a given chain are correlated with each other. Ideally one would want all the samples to be independent from each other. In LQCD calculations, since the typical size of the ensembles is rather small, it is critical to ensure the independence of the gauge field configurations or at least account for their autocorrelation as a systematic error. For a more detailed description, see Appendix 1A.1.

When approaching the critical point of a theory the autocorrelation time increases. In the case of LQCD, this happens when the lattice spacing approaches zero. Calculations at small a are already more expensive as the total volume in physical units is usually kept constant, so the lattices tend to have more points towards the continuum limit. However, due to the phenomenon of "critical slowing down", the autocorrelation time also increases, further augmenting the cost of these simulations.

Lattice QCD has, as opposed to the continuum theory, a non-fixed topology. The topological charge of the configurations in an ensemble fluctuates and its distribution is supposed to be Gaussian. When approaching the continuum, different topological sectors become increasingly separated and the MCMC algorithms, such as the HMC, struggle to sample all the sectors properly.

This is the phenomenological explanation for the critical slowing down. For a more detailed explanation of topological effects in LQCD, see Appendix 1B.

Dirac Matrix Condition Number As discussed in Section 1.3, the linear solver for quark propagators is solved iteratively, using Krylov-subspace linear system solvers. This means that if after n iterations the solver has satisfied the inequality

$$\|r_n\| < \epsilon \|\eta\|, \tag{1.56}$$

one says that the quark propagator is obtained with accuracy ϵ and terminated the algorithm. In general, ϵ is chosen to be small enough that any effects coming from the linear solver truncation are smaller than the stochastic uncertainties coming from ensemble averages of observables. Rewriting the above relation as

$$\|DS - D\psi_n\| < \epsilon \|DS\|, \tag{1.57}$$

and using standard bounds from the matrix norm inherited from the L_2 vector norm, one deduces that the relative deviation of the solution after n solver iterations, ψ_n and the real solution S

$$\|S - \psi_n\| < \epsilon \kappa(D) \|S\|, \tag{1.58}$$

is controlled by ϵ and the condition number of the matrix D

$$\kappa(D) = \|D\| \cdot \|D^{-1}\| = \sqrt{\frac{\lambda_{\max}(D^\dagger D)}{\lambda_{\min}(D^\dagger D)}}, \tag{1.59}$$

where we have used the matrix norm induced by the spectral radius of the matrix D.

In lattice QCD the lowest eigenvalue of $D^\dagger D$ is proportional to the quark mass squared and the largest eigenvalue is proportional to the inverse lattice spacing squared. The implication is that as the quark mass is decreased, the linear system becomes increasingly ill-conditioned, leading to longer iteration times and possible failures in the solver algorithm.
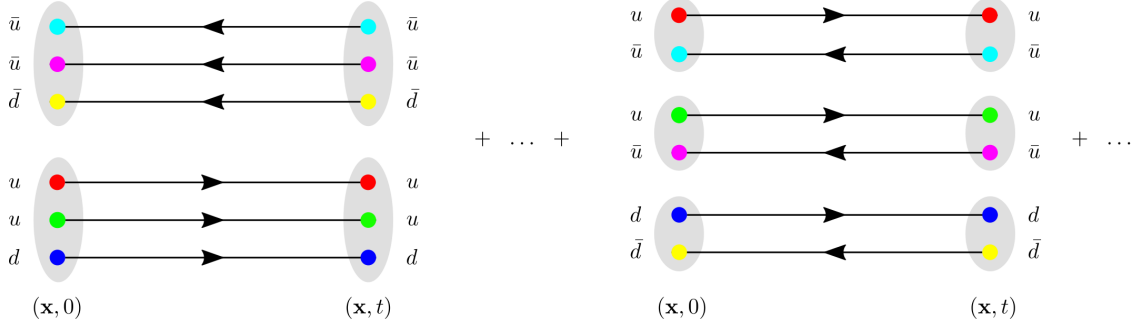
Figure 1.13: Contractions of the square of a proton interpolating operator.

Signal to Noise When calculating baryonic observables in LQCD, an insurmountable problem is that of the Signal-to-Noise. The variance of an observable is defined as $\sigma^2(\mathcal{O}) = \langle O^2 \rangle - \langle O \rangle^2$. In Figure 1.10, we showed the contractions for a single proton, but the possible contractions for the squared of the operator would be as in Figure 1.13. As shown pictorially, there are more possible contractions: one is the expected two baryon system, while another possible one is that of three pions propagating.

Because on the lattice, in the large t limit, the lightest state dominates, the above picture is problematic, as the variance will be dominated by the three pion state and not by the two nucleons one. In detail, the signal-to-noise can be given by:

$$StN = \frac{\langle \mathcal{O} \rangle}{\sigma_{\mathcal{O}}} \propto \frac{e^{-tm_p}}{e^{-\frac{3}{2}tm_\pi}} = \exp{-t(m_p - \frac{3}{2}m_\pi)} \tag{1.60}$$

This poses a challenge in extracting the ground state of such hadrons, as for small t the correlator is contaminated by excited states, but for large t the signal is lost due to noise. This effect becomes stronger as the mass gap between the nucleon and the pion increases, i.e. when approaching the physical point. Strong evidence of this problem can be seen in Figure 1.12 where the pion correlator and effective mass have a very smooth behavior and constant signal-to-noise. In contrast, the proton data clearly shows degradation of signal towards the center of the lattice.

To summarize, Lattice QCD is a mature field of research that is now a crucial component in our understanding of the SM. Thanks to the advances in computing power and numerical methods it can now be used for precision calculations of non-perturbative phenomena. There are, however,

some challenges that need to be addressed either with small incremental improvements to more traditional approaches, like new lattice discretizations or better algorithms, or even with entirely novel methods such as Machine Learning or Quantum Computing among others.

APPENDICES

1A.1 Autocorrelation of Data

One inherent characteristic of Markov Chain Monte Carlo methods is the correlation between samples along the chain. This effect, called autocorrelation, is rather intuitive since the algorithms work by modifying the latest element on the chain by some random value, which naturally implies that the new element has some relation with the previous one if the change is not sufficiently large. This phenomenon leads to an effective decrease of the sample size as not all configurations are statistically independent. It is thus crucial to consider autocorrelation effects in MCMC calculations.

A measure of autocorrelation is given by the *integrated autocorrelation time*, denoted with τ_{int}, which, given a set of ordered variables $\{x_1, x_2, \ldots x_N\}$ indexed by a time coordinate t is defined as:

$$\tau_{\text{int}} = \frac{1}{2} \sum_{t=1}^{\infty} \frac{\Gamma(t)}{\Gamma(0)} = \frac{1}{2} \sum_{t=1}^{\infty} \rho(t), \tag{1A.1}$$

where $\Gamma(t)$ is the autocorrelation function of points at a distance t in the chain:

$$\Gamma(t) = \Gamma(-t) = \langle (x_i - \bar{x})(x_{i+t} - \bar{x}) \rangle \approx \frac{1}{N-t} \sum_{i=1}^{N-t} (x_i - \bar{x})(x_{i+t} - \bar{x}). \tag{1A.2}$$

With this definition, in the case of large N, the true variance of the data is:

$$\tilde{\sigma}^2 = 2\tau_{\text{int}} \sigma^2, \tag{1A.3}$$

this factor can be estimated numerically for a given Markov chain and used to correct the results a posteriori. It is easy to see that if $\tau_{\text{int}} = \frac{1}{2}$ then the variance is unchanged, meaning there is no autocorrelation effect, and this can happen only if $\Gamma(t) = 0$ for all $t \neq 0$, i.e. all samples are uncorrelated.

Unfortunately, on a finite sample of size N the ratio $\rho(t)$ becomes noisy for large t, making it hard to determine τ_{int}. However, the noise must sum to 0 in the large N limit, so that one

can truncate the summation in Equation (1A.1) to a suitable window of size W. Following the procedure in [55] the square of the deviations of $\rho(t)$ can be computed approximately as:

$$\langle \delta\rho(t)^2 \rangle \approx \frac{1}{N} \sum_{k=1}^{\infty} \left[\rho(k+t) + \rho(k-t) - 2\rho(k)\rho(t) \right]^2. \tag{1A.4}$$

The terms in this equation all vanish for a sufficiently large value of k, which means one can introduce a cut-off Λ, usually of order $\mathcal{O}(10-100)$ and only sum for $k = 1, \dots, \Lambda$. One can then define the width of the summation window for τ_{int} to be first value of t where $\rho(t)$ is smaller than its fluctuations, i.e. where $\rho(t) < \sqrt{\langle \delta\rho(t)^2 \rangle}$. One can also define an approximate error formula for the integrated autocorrelation time, which reads:

$$\sigma^2(\tau_{\mathrm{int}}) \approx \frac{2(2W+1)}{N} \tau_{\mathrm{int}}^2. \tag{1A.5}$$

Throughout this work, the autocorrelation of data computed on lattice gauge field configurations has been estimated using this procedure.

1A.1.1 Statistical Blocking

A straightforward way to handle autocorrelated data is to use the method of Blocking [56]. It works by creating blocks of data of size B out of a data set by averaging B points along the Markov chain, creating a new chain of length N/B. The block-averages are then used as the new variables for the analysis, with the new data points being less autocorrelated than the unblocked ones.

To estimate the optimal block size B one should calculate expectation values and their variance for several increasing values of B, until the variance reaches a plateau. It can be shown that the variance tends to the same value as the one corrected with the integrated autocorrelation time from Equation (1A.3). In fact, blocking can be used as a cheap method to estimate τ_{int}.

1A.2 The Bootstrap Resampling Method

Resampling methods in statistics can be beneficial to correctly estimating the population mean and its uncertainty by utilizing subsets of a given sample of data. In the case of LQCD, since

lattice ensembles are usually small in size, resampling is beneficial to improve the precision of the calculations.

A first estimate of the expectation value and its uncertainty, on a set of N uncorrelated measurements, for an observable quantity O, is given by the mean and standard error of the mean:

$$\hat{O} = \frac{1}{N} \sum_{i=1}^{N} O_i, \tag{1A.6}$$

$$\sigma_{\hat{O}} = \sqrt{\left\langle \left(\hat{O} - \langle O \rangle \right)^2 \right\rangle} = \sqrt{\frac{1}{N^2} \sum_{i,j=1}^{N} \left\langle \left(\langle O_i \rangle - \langle O \rangle \right) \left(\langle O_j \rangle - \langle O \rangle \right) \right\rangle} = \frac{\sigma_O}{\sqrt{N}},$$

where O_i are the measurements for the observable O.

The Bootstrap method [51] relies on building a set of N_b *bootstrap samples* by choosing N points at random from the the original data O_i with repetition. It then uses the samples' mean to obtain an estimator of the population mean. In particular, given the mean of a bootstrap sample:

$$\hat{O}_b = \frac{1}{N} \sum_{i=1}^{N} O_{r(N)}, \tag{1A.7}$$

where $r(N)$ signifies a random number drawn between 1 and N, then the population mean and its variance are given by:

$$\tilde{O} = \frac{1}{N_b} \sum_{b=1}^{N_b} \hat{O}_b \qquad \tilde{\sigma}_{\tilde{O}}^2 = \frac{1}{N_b} \sum_{b=1}^{N_b} \left(O_b - \tilde{O} \right)^2. \tag{1A.8}$$

The expectation value and the uncertainty of O are then given $\langle O \rangle = \tilde{O}$ and its associated uncertainty is $\tilde{\sigma}_{\tilde{O}}$. It should be noted that the results are not unbiased, in fact, $\tilde{O} \neq \hat{O}$ and their difference gives the order of the possible deviations of $\tilde{O}$ from the true value $\langle O \rangle$.

The Bootstrap method is numerically inexpensive and can also be used for derived quantities of the observable O. Furthermore, in the case of LQCD, if one computes multiple observables on the same ensemble, the bootstrap method can be applied to estimate the population means of composite quantities, as long as the random samples are drawn simultaneously, thus preserving the correlation of the data.

APPENDIX 1B

OVERVIEW OF TOPOLOGICAL EFFECTS IN LATTICE QCD

The QCD vacuum exhibits particular topological properties that related to important physical phenomena [57, 58]. One important property is the topological charge, which can intuitively be understood as the winding number of the quantum field. In the continuum theory, fields with different topological charges are separated as there is no unitary transformation that can connect them; we often refer to these as *topological sectors*. Instead, when QCD is discretized on a lattice, the MCMC simulations have access to all sectors, thanks to non-zero instanton modes [59], and, in fact, observables are only well defined if all the sectors are sampled correctly. A side-effect of applying boundary periodic conditions to the gauge field is, however, that when approaching the continuum limit, the topological sectors become increasingly separated, i.e. the MCMC algorithms have lower probabilities of tunneling from one sector to another. This, in practice, means that to achieve proper sampling of an observable requires longer runs the closer we get to the continuum [60].

On the lattice, there are different ways to define the topological charge: one is fermionic and is based on the Atiyah-Singer index theorem [61] and relates the charge to the distribution of the spectrum of the Dirac operator [62]; in contrast, the other definition, which is the one we will discuss now, only comes from gauge field effects. The two are identical in the continuum, and purely topological observables, like the topological susceptibility, should be independent of the definition used [63, 64].

The topological charge can be written as the the integral over all spacetime of the topological charge density:

$$Q = \int d^4x\, q(x), \tag{1B.1}$$

where

$$q(x) = \frac{1}{32\pi^2} \text{Tr}(G_{\mu\nu}(x)\tilde{G}^{\mu\nu}(x)). \tag{1B.2}$$

The term $\tilde{G}^{\mu\nu}$ represents the dual of the gauge field tensor:

$$\tilde{G}^{\mu\nu}(x) = \frac{1}{2}\epsilon^{\mu\nu\rho\sigma}G^{\rho\sigma}(x), \tag{1B.3}$$

with $\epsilon^{\mu\nu\rho\sigma}$ being the totally anti-symmetric Levi-Civita symbol. On the lattice, the topological charge in Equation (1B.1) can be computed using a suitable discretized definition of the gauge tensor. As we have seen in Section 1.2.2, this can be done using the Clover term, so the topological charge becomes:

$$Q[U] = \frac{1}{64\pi^2}\sum_{n\in\Lambda}\sum_{\mu<\nu}\epsilon^{\mu\nu\rho\sigma}\mathrm{Tr}[G^{(\text{clover})}_{\mu\nu}(n)G^{(\text{clover})}_{\rho\sigma}(n)]. \tag{1B.4}$$

This definition is susceptible to short-distance fluctuations of the gauge field and is therefore very noisy, in fact, it is divergent in the continuum. A tool to improve its determination is the Gradient Flow, which is discussed more in detail in Appendix 1C, as it dampens the high-frequency modes, only leaving the long-scale signal without needing any additional renormalization as the topological charge is a purely gluonic observable.

A quantity of interest is the topological susceptibility χ, that is related to the second cumulant of the distribution of the topological charge and is defined as:

$$\chi = \frac{1}{a^4|\Lambda|}\langle Q^2\rangle. \tag{1B.5}$$

This quantity is crucial in understanding essential properties of instantons on the lattice and, through the Witten-Veneziano formula [57], it can be shown to be proportional to the mass of the η' meson.

The Gradient Flow, or Wilson Flow, is a method for studying non-perturbative effects of QCD, first proposed by M. Lüscher in the context of trivializing maps [65], but later explored more in depths in [54, 66] and following papers. It works by modifying the fields according to a gauge-covariant "diffusion-like" flow equation which evolves them towards the stationary points of the action. For $SU(3)$ this equation reads:

$$\partial_{t_f} B_\mu = D_\nu G_{\nu\mu}, \tag{1C.1}$$

where t_f is the so-called *flow-time*, the parameter that controls the evolution, and $B_\mu(t_f, x)$ is the flowed version of the original gauge field with the initial condition at zero flow-time set to be: $B_\mu|_{t_f=0} = A_\mu$. The other terms in the equation are just the generalization to the B_μ field of the covariant derivative and the field tensor:

$$G_{\mu\nu} = \partial_\mu B_\nu - \partial_\nu B_\mu + [B_\mu, B_\nu], \tag{1C.2}$$

$$D_\mu = \partial_\mu + [B_\mu, \cdot]. \tag{1C.3}$$

From Equation (1C.1) we can see that the right-hand side is proportional to the gradient of the action; hence the name. Its effect is intuitively that of moving the field B_μ along the steepest descent path of the action. This, in turn, corresponds to a "Gaussian-like" smearing of the gauge fields with a smearing root-mean-square radius of size $\sqrt{8t_f}$.

On the lattice, one can rewrite the flow equations in terms of the field $V_{t_f}(x, \mu)$ which is the flow generalization of the link filed $U(x, \mu)$:

$$\partial_{t_f} V_{t_f}(x, \mu) = -g_0^2 [\partial_{x,\mu} S_G(V_{t_f})] V_{t_f}(x, \mu), \tag{1C.4}$$

where S_G is the Wilson plaquette action, see Equation (1.23), computed with the $V_{t_f}(x, \mu)$. This equation can be integrated numerically on the lattice using Runge-Kutta Munthe-Kaas methods [67, 68].

One crucial fact about the gradient flow is that the flowed gauge fields do not need any renormalization; therefore, correlation functions containing them have a well-defined continuum limit [66]. For example, in the case of the topological charge that we presented in Appendix 1B, the gradient flow serves to dampen the high-frequency modes removing noise but leaving the topological charge unaltered. It is, therefore, an advantageous method to accurately determine the topological charge on the lattice. In fig. 1C.1, we show an example of the smearing effect given by the gradient flow on the topological charge density. One can observe that the highly fluctuating values of the density as t_f increases are removed, leaving only the long-distance effects that are the ones that truly contribute to the topological charge.
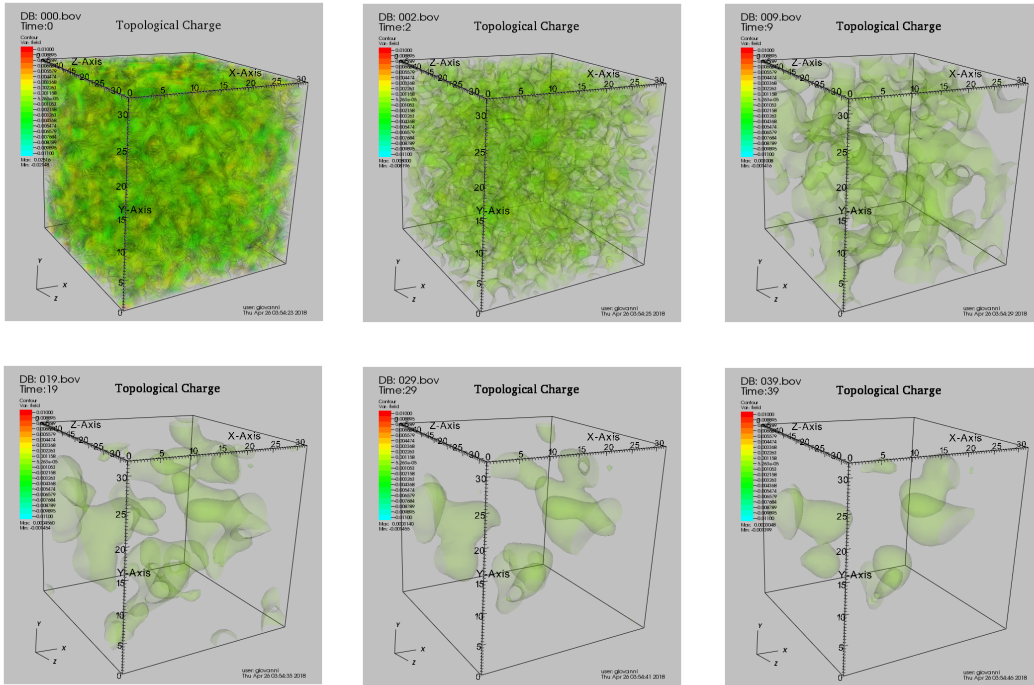
Figure 1C.1: Topological charge computed at fixed euclidean time on a pure gauge lattice of size $32^3 \times 64$ with spacing $a = 0.06793$ fm. The different figures represent different values of the flow time with radii $\sqrt{8t_\mathrm{f}} = 0, 0.14, 0.30, 0.43, 0.52, 0.60$ fm. The visualization has been performed using the software called "LatViz", developed together with H.M. Vege interface based on the open source visualization tool by LLNL VisIt [69].

The topological charge at $\sqrt{8t_\mathrm{f}} = 0$ fm (the non-flowed field configuration) is highly affected by short-distance over the lattice sites. One can notice that the effect of the flow equation is to

remove these fluctuations, that at $\sqrt{8t_f} = 0.60$ fm are almost entirely gone. For larger smearing radii, the emergence of well-separated topological sectors becomes visible.

1C.1 Scale Fixing with the Gradient Flow

In one of the first articles about the gradient flow, it was suggested that the method could be used for scale setting in Lattice QCD [54]. In particular, this is done by considering the dimensionless quantity $t_f^2 \langle E \rangle$, in the region where $\sqrt{8t_f}$ is at a hadronic scale. In that case, the quantity has a very weak dependence on the lattice spacing, and thus, it is, also considering that it can be computed with high numerical precision on the lattice, suitable for scale setting. It was proposed to consider the value t_0, defined as the point where:

$$t_0^2 \langle E \rangle = 0.3 \tag{1C.5}$$

as scale. In practice this means that all lattice observables are then expressed in terms of t_0/a^2, while the value of t_0 needs to be fixed using a physical observable, for example as done in [70], where they find the value $\sqrt{t_0} = 0.1464(18)$ fm. Conversely, given an ensemble and the physical value of t_0, one only needs to find the value of t_f/a^2 where Equation (1C.5) is satisfied to find t_0 and then obtain the lattice spacing a, see Figure 3.10 for example.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] D. Galbraith and C. Burgard. "Standard model of the standard model." (Nov. 12, 2013), [Online]. Available: https://texample.net/tikz/examples/model-physics/.

[2] K. G. Wilson, "Confinement of quarks," *Physical Review D*, vol. 10, no. 8, pp. 2445–2459, Oct. 15, 1974, ISSN: 0556-2821. DOI: 10.1103/PhysRevD.10.2445.

[3] S. Dürr *et al.*, "Ab initio determination of light hadron masses," *Science*, vol. 322, no. 5905, pp. 1224–1227, Nov. 21, 2008, ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.1163233.

[4] A. Bazavov, "An overview of (selected) recent results in finite-temperature lattice QCD," *Journal of Physics: Conference Series*, vol. 446, p. 012 011, Sep. 2013, ISSN: 1742-6596. DOI: 10.1088/1742-6596/446/1/012011.

[5] H.-W. Lin *et al.*, "Parton distributions and lattice QCD calculations: A community white paper," *Progress in Particle and Nuclear Physics*, vol. 100, pp. 107–160, May 1, 2018, ISSN: 0146-6410. DOI: 10.1016/j.ppnp.2018.01.007.

[6] S. R. Beane, W. Detmold, K. Orginos, and M. J. Savage, "Nuclear physics from lattice QCD," *Progress in Particle and Nuclear Physics*, vol. 66, no. 1, pp. 1–40, Jan. 1, 2011, ISSN: 0146-6410. DOI: 10.1016/j.ppnp.2010.08.002.

[7] J. Dragos, T. Luu, A. Shindler, J. de Vries, and A. Yousif, "Confirming the existence of the strong CP problem in lattice QCD with the gradient flow," *Physical Review C*, vol. 103, no. 1, p. 015 202, Jan. 19, 2021. DOI: 10.1103/PhysRevC.103.015202.

[8] S. Borsanyi *et al.*, "Leading hadronic contribution to the muon magnetic moment from lattice QCD," *Nature*, vol. 593, no. 7857, pp. 51–55, May 2021, ISSN: 1476-4687. DOI: 10.1038/s41586-021-03418-1.

[9] V. Cirigliano, W. Detmold, A. Nicholson, and P. Shanahan, "Lattice QCD inputs for nuclear double beta decay," *Progress in Particle and Nuclear Physics*, vol. 112, p. 103 771, May 1, 2020, ISSN: 0146-6410. DOI: 10.1016/j.ppnp.2020.103771.

[10] M. Gell-Mann, "The eightfold way: A theory of strong interaction symmetry," TID-12608, CTSL-20, 4008239, Mar. 15, 1961, TID−12 608, CTSL−20, 4 008 239. DOI: 10.2172/4008239.

[11] Y. Ne'eman, "Derivation of strong interactions from a gauge invariance," *Nuclear Physics*, vol. 26, no. 2, pp. 222–229, Aug. 1961, ISSN: 00295582. DOI: 10.1016/0029-5582(61)90134-1.

[12] O. W. Greenberg, "Spin and unitary-spin independence in a paraquark model of baryons and mesons," *Physical Review Letters*, vol. 13, no. 20, pp. 598–602, Nov. 16, 1964, ISSN: 0031-9007.

DOI: 10.1103/PhysRevLett.13.598.

[13]   M. Y. Han and Y. Nambu, "Three-triplet model with double SU (3) symmetry," *Physical Review*, vol. 139, no. 4, B1006–B1010, Aug. 23, 1965, ISSN: 0031-899X. DOI: 10.1103/PhysRev.139.B1006.

[14]   H. Fritzsch, M. Gell-Mann, and H. Leutwyler, "Advantages of the color octet gluon picture," *Physics Letters B*, vol. 47, no. 4, pp. 365–368, Nov. 1973, ISSN: 03702693. DOI: 10.1016/0370-2693(73)90625-4.

[15]   Particle Data Group *et al.*, "Review of particle physics," *Progress of Theoretical and Experimental Physics*, vol. 2022, no. 8, p. 083C01, Aug. 8, 2022, ISSN: 2050-3911. DOI: 10.1093/ptep/ptac097.

[16]   M. E. Peskin, *An Introduction To Quantum Field Theory*. Boca Raton: CRC Press, Jan. 30, 2018, 866 pp., ISBN: 978-0-429-50355-9. DOI: 10.1201/9780429503559.

[17]   Y. Aoki *et al.*, "FLAG review 2021," *The European Physical Journal C*, vol. 82, no. 10, p. 869, Oct. 4, 2022, ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-022-10536-1.

[18]   S. Bethke, "Experimental tests of asymptotic freedom," *Progress in Particle and Nuclear Physics*, vol. 58, no. 2, pp. 351–386, Apr. 2007, ISSN: 01466410. DOI: 10.1016/j.ppnp.2006.06.001.

[19]   D. J. Gross and F. Wilczek, "Ultraviolet behavior of non-abelian gauge theories," *Physical Review Letters*, vol. 30, no. 26, pp. 1343–1346, Jun. 25, 1973, ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.30.1343.

[20]   H. D. Politzer, "Reliable perturbative results for strong interactions?" *Physical Review Letters*, vol. 30, no. 26, pp. 1346–1349, Jun. 25, 1973, ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.30.1346.

[21]   C. Gattringer and C. B. Lang, *Quantum Chromodynamics on the Lattice* (Lecture Notes in Physics). Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, vol. 788, ISBN: 978-3-642-01849-7. DOI: 10.1007/978-3-642-01850-3.

[22]   K. Symanzik, "Continuum limit and improved action in lattice theories," *Nuclear Physics B*, vol. 226, no. 1, pp. 187–204, Sep. 1983, ISSN: 05503213. DOI: 10.1016/0550-3213(83)90468-6.

[23]   B. Sheikholeslami and R. Wohlert, "Improved continuum limit lattice action for QCD with wilson fermions," *Nuclear Physics B*, vol. 259, no. 4, pp. 572–596, Sep. 30, 1985, ISSN: 0550-3213. DOI: 10.1016/0550-3213(85)90002-1.

[24]   P. H. Ginsparg and K. G. Wilson, "A remnant of chiral symmetry on the lattice," *Physical*

Review D, vol. 25, no. 10, pp. 2649–2657, May 15, 1982. DOI: 10.1103/PhysRevD.25.2649.

[25] D. B. Kaplan, "A method for simulating chiral fermions on the lattice," *Physics Letters B*, vol. 288, no. 3, pp. 342–347, Aug. 27, 1992, ISSN: 0370-2693. DOI: 10.1016/0370-2693(92)9 1112-M.

[26] D. B. Kaplan, "Chiral fermions on the lattice," *Nuclear Physics B - Proceedings Supplements*, Proceedings of the International Symposium on, vol. 30, pp. 597–600, Mar. 1, 1993, ISSN: 0920-5632. DOI: 10.1016/0920-5632(93)90282-B.

[27] Y. Shamir, "Chiral fermions from lattice boundaries," *Nuclear Physics B*, vol. 406, no. 1, pp. 90–106, Sep. 27, 1993, ISSN: 0550-3213. DOI: 10.1016/0550-3213(93)90162-I.

[28] V. Furman and Y. Shamir, "Axial symmetries in lattice QCD with kaplan fermions," *Nuclear Physics B*, vol. 439, no. 1, pp. 54–78, Apr. 10, 1995, ISSN: 0550-3213. DOI: 10.1016/0550-321 3(95)00031-M.

[29] R. Narayanan and H. Neuberger, "A construction of lattice chiral gauge theories," *Nuclear Physics B*, vol. 443, no. 1, pp. 305–385, Jun. 5, 1995, ISSN: 0550-3213. DOI: 10.1016/0550-3 213(95)00111-5.

[30] H. Neuberger, "Exactly massless quarks on the lattice," *Physics Letters B*, vol. 417, no. 1, pp. 141–144, Jan. 15, 1998, ISSN: 0370-2693. DOI: 10.1016/S0370-2693(97)01368-3.

[31] J. Kogut and L. Susskind, "Hamiltonian formulation of wilson's lattice gauge theories," *Physical Review D*, vol. 11, no. 2, pp. 395–408, Jan. 15, 1975. DOI: 10.1103/PhysRevD.11.395.

[32] A. Bazavov *et al.*, "Equation of state in (2 + 1)-flavor QCD," *Physical Review D*, vol. 90, no. 9, p. 094 503, Nov. 4, 2014, ISSN: 1550-7998, 1550-2368. DOI: 10.1103/PhysRevD.90.094503.

[33] S. Aoki, "New phase structure for lattice QCD with wilson fermions," *Physical Review D*, vol. 30, no. 12, pp. 2653–2663, Dec. 15, 1984. DOI: 10.1103/PhysRevD.30.2653.

[34] R. Frezzotti, P. A. Grassi, S. Sint, and P. Weisz, "Lattice QCD with a chirally twisted mass term," *Journal of High Energy Physics*, vol. 2001, no. 8, pp. 058–058, Aug. 29, 2001, ISSN: 1029-8479. DOI: 10.1088/1126-6708/2001/08/058.

[35] A. Shindler, "Twisted mass lattice QCD," *Physics Reports*, vol. 461, no. 2, pp. 37–110, May 2008, ISSN: 03701573. DOI: 10.1016/j.physrep.2008.03.001.

[36] R. Abbott *et al.*, "Sampling QCD field configurations with gauge-equivariant flow models," 2022, Publisher: arXiv Version Number: 2. DOI: 10.48550/ARXIV.2208.03832.

[37] S. Foreman, T. Izubuchi, L. Jin, X.-Y. Jin, J. C. Osborn, and A. Tomiya, "HMC with normalizing

flows," 2021, Publisher: arXiv Version Number: 1. DOI: 10.48550/ARXIV.2112.01586.

[38]   S. Duane, A. Kennedy, B. J. Pendleton, and D. Roweth, "Hybrid monte carlo," *Physics Letters B*, vol. 195, no. 2, pp. 216–222, Sep. 1987, ISSN: 03702693. DOI: 10.1016/0370-2693(87)91197-X.

[39]   H. A. van der Vorst, "Bi-CGSTAB: A fast and smoothly converging variant of bi-CG for the solution of nonsymmetric linear systems," *SIAM Journal on Scientific and Statistical Computing*, vol. 13, no. 2, pp. 631–644, Mar. 1992, ISSN: 0196-5204, 2168-3417. DOI: 10.1137/0913035.

[40]   M. Lüscher, "Solution of the dirac equation in lattice QCD using a domain decomposition method," *Computer Physics Communications*, vol. 156, no. 3, pp. 209–220, Jan. 15, 2004, ISSN: 0010-4655. DOI: 10.1016/S0010-4655(03)00486-7.

[41]   J. Brannick, R. C. Brower, M. A. Clark, J. C. Osborn, and C. Rebbi, "Adaptive multigrid algorithm for lattice QCD," *Physical Review Letters*, vol. 100, no. 4, p. 041601, Jan. 28, 2008, ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.100.041601.

[42]   S. Aoki *et al.*, "Non-perturbative renormalization of quark mass in nf= 2 + 1 QCD with the schrödinger functional scheme," *Journal of High Energy Physics*, vol. 2010, no. 8, p. 101, Aug. 23, 2010, ISSN: 1029-8479. DOI: 10.1007/JHEP08(2010)101.

[43]   M. Albanese *et al.*, "Glueball masses and string tension in lattice QCD," *Physics Letters B*, vol. 192, no. 1, pp. 163–169, Jun. 25, 1987, ISSN: 0370-2693. DOI: 10.1016/0370-2693(87)91160-9.

[44]   A. Hasenfratz and F. Knechtli, "Flavor symmetry and the static potential with hypercubic blocking," *Physical Review D*, vol. 64, no. 3, p. 034504, Jul. 2, 2001, ISSN: 0556-2821, 1089-4918. DOI: 10.1103/PhysRevD.64.034504.

[45]   C. Morningstar and M. Peardon, "Analytic smearing of SU (3) link variables in lattice QCD," *Physical Review D*, vol. 69, no. 5, p. 054501, Mar. 23, 2004, ISSN: 1550-7998, 1550-2368. DOI: 10.1103/PhysRevD.69.054501.

[46]   S. Güsken, "A study of smearing techniques for hadron correlation functions," *Nuclear Physics B - Proceedings Supplements*, vol. 17, pp. 361–364, Sep. 1, 1990, ISSN: 0920-5632. DOI: 10.1016/0920-5632(90)90273-W.

[47]   C. Michael, "Adjoint sources in lattice gauge theory," *Nuclear Physics B*, vol. 259, no. 1, pp. 58–76, Sep. 16, 1985, ISSN: 0550-3213. DOI: 10.1016/0550-3213(85)90297-4.

[48]   M. Lüscher and U. Wolff, "How to calculate the elastic scattering matrix in two-dimensional quantum field theories by numerical simulation," *Nuclear Physics B*, vol. 339, no. 1, pp. 222–252, Jul. 23, 1990, ISSN: 0550-3213. DOI: 10.1016/0550-3213(90)90540-T.

[49] G. T. Fleming, S. D. Cohen, H.-W. Lin, and V. Pereyra, "Excited-state effective masses in lattice QCD," *Physical Review D*, vol. 80, no. 7, p. 074 506, Oct. 13, 2009, ISSN: 1550-7998, 1550-2368. DOI: 10.1103/PhysRevD.80.074506.

[50] G. Plonka and M. Tasche, "Prony methods for recovery of structured functions," *GAMM-Mitteilungen*, vol. 37, no. 2, pp. 239–258, 2014, ISSN: 1522-2608. DOI: 10.1002/gamm.201410011.

[51] B. Efron, "Bootstrap methods: Another look at the jackknife," *The Annals of Statistics*, vol. 7, no. 1, Jan. 1, 1979, ISSN: 0090-5364. DOI: 10.1214/aos/1176344552.

[52] R. Sommer, "A new way to set the energy scale in lattice gauge theories and its application to the static force and coupling constant in SU (2) yang-mills theory," *Nuclear Physics B*, vol. 411, no. 2, pp. 839–854, Jan. 1994, ISSN: 05503213. DOI: 10.1016/0550-3213(94)90473-1.

[53] S. Capitani, M. Della Morte, G. von Hippel, B. Knippschild, and H. Wittig, "Scale setting via the omega baryon mass," 2011, Publisher: arXiv Version Number: 1. DOI: 10.48550/ARXIV.1110.6365.

[54] M. Lüscher, "Properties and uses of the wilson flow in lattice QCD," *Journal of High Energy Physics*, vol. 2010, no. 8, p. 71, Aug. 2010, ISSN: 1029-8479. DOI: 10.1007/JHEP08(2010)071.

[55] A. Sokal, "Monte carlo methods in statistical mechanics: Foundations and new algorithms," in *Functional Integration*, C. DeWitt-Morette, P. Cartier, and A. Folacci, Eds., vol. 361, Series Title: NATO ASI Series, Boston, MA: Springer US, 1997, pp. 131–192, ISBN: 978-1-4899-0321-1. DOI: 10.1007/978-1-4899-0319-8_6.

[56] H. Flyvbjerg and H. G. Petersen, "Error estimates on averages of correlated data," *The Journal of Chemical Physics*, vol. 91, no. 1, pp. 461–466, Jul. 1989, ISSN: 0021-9606, 1089-7690. DOI: 10.1063/1.457480.

[57] E. Witten, "Current algebra theorems for the u(1) "goldstone boson"," *Nuclear Physics B*, vol. 156, no. 2, pp. 269–283, Sep. 1979, ISSN: 05503213. DOI: 10.1016/0550-3213(79)90031-2.

[58] A. Di Giacomo, "Topology in lattice QCD," 1997, Publisher: arXiv Version Number: 1. DOI: 10.48550/ARXIV.HEP-LAT/9711034.

[59] T. Schäfer and E. V. Shuryak, "Instantons in QCD," *Reviews of Modern Physics*, vol. 70, no. 2, pp. 323–425, Apr. 1, 1998, ISSN: 0034-6861, 1539-0756. DOI: 10.1103/RevModPhys.70.323.

[60] M. Lüscher and S. Schaefer, "Lattice QCD without topology barriers," *Journal of High Energy Physics*, vol. 2011, no. 7, p. 36, Jul. 7, 2011, ISSN: 1029-8479. DOI: 10.1007/JHEP07(2011)036.

[61] M. F. Atiyah and I. M. Singer, "The index of elliptic operators on compact manifolds," *Bulletin of the American Mathematical Society*, vol. 69, no. 3, pp. 422–433, 1963, ISSN: 0002-9904, 1936-881X. DOI: 10.1090/S0002-9904-1963-10957-X.

[62] H. Leutwyler and A. Smilga, "Spectrum of dirac operator and role of winding number in QCD," *Physical Review D*, vol. 46, no. 12, pp. 5607–5632, Dec. 15, 1992, ISSN: 0556-2821. DOI: 10.1103/PhysRevD.46.5607.

[63] C. Alexandrou *et al.*, "Comparison of topological charge definitions in lattice QCD," 2017, Publisher: arXiv Version Number: 2. DOI: 10.48550/ARXIV.1708.00696.

[64] D. J. Gross, R. D. Pisarski, and L. G. Yaffe, "QCD and instantons at finite temperature," *Reviews of Modern Physics*, vol. 53, no. 1, pp. 43–80, Jan. 1, 1981, ISSN: 0034-6861. DOI: 10.1103/RevModPhys.53.43.

[65] M. Lüscher, "Trivializing maps, the wilson flow and the HMC algorithm," *Communications in Mathematical Physics*, vol. 293, no. 3, pp. 899–919, Feb. 2010, ISSN: 0010-3616, 1432-0916. DOI: 10.1007/s00220-009-0953-7.

[66] M. Lüscher and P. Weisz, "Perturbative analysis of the gradient flow in non-abelian gauge theories," *Journal of High Energy Physics*, vol. 2011, no. 2, p. 51, Feb. 2011, ISSN: 1029-8479. DOI: 10.1007/JHEP02(2011)051.

[67] H. Munthe-Kaas, "Lie-butcher theory for runge-kutta methods," *BIT Numerical Mathematics*, vol. 35, no. 4, pp. 572–587, Dec. 1, 1995, ISSN: 1572-9125. DOI: 10.1007/BF01739828.

[68] H. Munthe-Kaas, "Runge-kutta methods on lie groups," *BIT Numerical Mathematics*, vol. 38, no. 1, pp. 92–111, Mar. 1, 1998, ISSN: 1572-9125. DOI: 10.1007/BF02510919.

[69] H. Childs *et al.*, "VisIt: An end-user tool for visualizing and analyzing very large data," Nov. 2012.

[70] M. Bruno, T. Korzec, and S. Schaefer, "Setting the scale for the CLS 2 + 1 flavor ensembles," *Physical Review D*, vol. 95, no. 7, p. 074 504, Apr. 12, 2017, ISSN: 2470-0010, 2470-0029. DOI: 10.1103/PhysRevD.95.074504.

CHAPTER 2

MACHINE LEARNING METHODS FOR HADRON CORRELATORS FROM LATTICE QCD

Despite the great hardware, software and algorithmic developments, the computational cost of a realistic lattice QCD calculation is still among the most demanding numerical calculations performed regularly in High-Performance Computing (HPC) facilities around the world [1–3]. As an example, four out of five of the largest awards of the 24th call of the Partnership for Advances Computing in Europe in 2021 were for LQCD [4].

The majority of the time spent by a lattice QCD calculation is in the computation of the quark propagator, which not only provides the building block for the determination of any fermionic correlation function, but it is needed repeatedly during the Markov Chain Monte Carlo (MCMC) used to generate the gauge ensemble with dynamical quarks, as discussed in Section 1.3. The calculation of the quark propagator is achieved by inverting a very large sparse matrix representing the lattice Dirac operator, see Section 1.4. The problem becomes a sparse linear system once we fix the location of the source, however, it can still contain $\mathcal{O}(10^9)$ unknowns because the quark propagator is effectively a vector of the size of the number of lattice sites times a factor 12 coming from color and Dirac indices.

These types of linear systems are usually solved by adopting Krylov Subspace (KS) methods, such as the Conjugate Gradient (CG) or its variations like the Biconjugate Gradient Stabilized (BiCGSTAB) method[5]. However, due to the computational advance in recent years, calculations started to approach the continuum limit or the physical value of the light quark masses, but this leads to a tremendous increase in the solver iteration count [6]. Several preconditioners can be used to speed-up these type of calculations such as even-odd preconditioning, domain-decomposition [7] and deflation [8]. A successful way to precondition KS solvers is based on several variants of so-called *multigrid* algorithms such as inexact deflation [9, 10], the Multigrid Generalized Conjugate Residual (MG-GCR) [11] and multigrid approaches which are adaptive

aggregation based such as the DD-αAMG [12].

Recently, some studies on how Machine Learning (ML) algorithms could be used to speed up lattice calculations have been released. Since these methods have matured considerably over the past decade, there is a lot of interest in understanding if and how they can be employed in LQCD calculations. ML algorithms have already been applied in [13] to try to reconstruct the dependence of more complicated observables on simpler ones, or, for example, in [14]. Other studies have focused on so-called normalizing flows for generating configurations from direct sampling [15, 16], attempting to find a replacement, full or partial, of the HMC using generative ML models.

At the same time, there have been studies in Lattice QCD for methods of noise reduction [17], mainly in the context of disconnected diagrams, such as the hopping parameter expansion technique, the truncated solver method and the truncated eigenmode acceleration. One of these, the truncated solver method (TSM), relies on the rapid convergence of solvers to the solution within very few iterations, with most of the time being spent afterwards in gaining only little precision more. In more precise terms, the logarithm of the residue of the quark propagator linear system has a roughly linear dependency on the number of iterations, as we will see in Section 2.1. This well-known fact can be used to reduce the variance if one computes a set of N propagators with stochastic sources on a gauge field configuration with very low precision, then on a subset $M \ll N$ a high precision propagator is computed as well. The high precision data is then used to compute the possible bias of the low precision estimates.

In this chapter, we propose a different strategy, in a similar spirit to the TSM: by taking advantage of recent developments in Machine Learning algorithms, in particular in the field of supervised learning, we apply them to the problem of speeding up the calculation of the hadron correlation functions. While ultimately the goal is to accelerate the determination of the quark propagator, in this work, we focus on derived quantities, hadron correlation functions and effective masses to reduce the dimensionality of the ML problem. As a first step in using ML to determine hadron correlators, we propose to combine the information contained in correlation functions

determined with quark propagators calculated with different precisions from the linear system solver. The analysis presented in this work shows that quark propagators calculated with sloppier precision than the standard one contain enough information to let the ML algorithm reconstruct the hadron correlators calculated with more precise quark propagators. To train the ML algorithm, one makes use of the ensemble generated with the MCMC and subsets of it.

This chapter is organized as follows: in Section 2.1 we thoroughly introduce the problem with some numerical lattice details and study the dependence on the solver precision and its impact on two-point functions. In Section 2.2 we introduce a few supervised ML algorithms and discuss their application to our calculation. In Section 2.3, we study the correlation of the data from our ensembles and propose a tentative algorithm, while in Section 2.4 we will present the results of the ML procedure. Finally, in Section 2.5 we summarize our conclusions.

2.1   Residue analysis of the BiCGSTAB Linear Solver

The convergence properties of the BiCGSTAB algorithm on a quark propagator problem are non-trivial. In particular, there is no guarantee that the residue of the linear system as a function of iteration counts is monotonically decreasing and, in fact, that is usually not the case. A description of the algorithm can be found in Appendix 2B. In this section, we investigate the convergence of the algorithm for quark propagator inversions for the LQCD ensembles we will use for our ML procedure from the traditional perspective.

In this work, we consider QCD with $N_f = 2 + 1$ dynamical quark flavors. The theory is regulated on a Euclidean lattice of spacing a with an Iwasaki gauge action [18] and a non-perturbative $\mathcal{O}(a)$ improved fermion action [19, 20] (clover fermions). The details of the ensembles we have used can be found in [21–23] and for completeness we summarize them in Table 2.1. The ensembles span a range of masses $m_\pi = 400 - 700$ MeV at a fixed lattice spacing (ensembles M_1, M_2, M_3), and three different lattice spacing while keeping the total volume and pion mass approximately fixed (ensembles A_1, A_2, M_1). It is conceivable that, in the range of masses and lattice spacings we have investigated in this work, the results we obtain are largely independent of

the fermion lattice action used. It would still be very interesting to test the algorithm we propose with different types of lattice actions such as Ginsparg-Wilson fermions [24].

	β	κ_l	κ_s	L	T	$a\,[fm]$	$m_\pi\,[MeV]$	N
M_1	1.90	0.13700	0.1364	32	64	0.0907(13)	699.0(3)	399
M_2	1.90	0.13727	0.1364	32	64	0.0907(13)	567.6(3)	400
M_3	1.90	0.13754	0.1364	32	64	0.0907(13)	409.7(7)	450
A_1	1.83	0.13825	0.1371	16	32	0.1095(25)	710(1)	800
A_2	1.90	0.13700	0.1364	20	40	0.0936(33)	676.3(7)	790

Table 2.1: Ensembles from the PACS-CS collaboration [22], with clover fermion action. Physical quantities (lattice spacing a and pion masses m_π) have been calculated for another work [23].

2.1.1 Krylov Space Solvers

To fix the notation, we briefly repeat some well-known facts about the determination of quark propagators with Krylov space solvers. For completeness, a few additional details can be found in Appendix 2A, while an exhaustive treatment of iterative methods for sparse linear systems can be found in [25]. The quark propagator $q(x)$ is obtained by solving the equation

$$Dq(x) = \eta(x)\,, \tag{2.1}$$

where D represents the lattice Dirac operator of choice at non-zero mass and $\eta(x)$ is the source field, as explained in Section 1.4. In our case, the lattice operator D is given by the non-perturbative improved Wilson fermion operator. The equation satisfied by the quark propagator Equation (2.1) is a very large and sparse linear system that can be solved by generating a sequence of solutions q_n of increasing precision. The accuracy of the solution after n iterations is measured by the norm of the residue

$$r_n = \eta - Dq_n\,. \tag{2.2}$$

In the following, when we refer generically to the norm, the L_2^n normed vector space and the corresponding induced norm on the vector space of matrices are to be intended. Definitions and selected properties of the norms used in this work can be found in Appendix 2A. As presented in

Section 1.4.2, the iterations of the KS algorithms are terminated when the inequality

$$\|r_n\| < \epsilon \|\eta\|, \tag{2.3}$$

is satisfied and thus the quark propagator is obtained with accuracy ϵ; we will also refer this parameter as *stopping criterion* and it is a crucial element in the discussion throughout this chapter.

2.1.2 BiCGSTAB Solver History

As an example of a Krylov space solver behavior for different accuracies, in the left plot of Figure 2.1 we show the average time needed to compute the quark propagator as a function of the stopping criterion for all the ensembles listed in Table 2.1. Calculations are usually run with very small values of ϵ to ensure that the solver precision is always smaller than the statistical uncertainty coming from the ensemble averages. However, we notice that reducing the precision of the calculation from $\epsilon = 10^{-8}$ to the range $\epsilon = 10^{-1} - 10^{-3}$ would imply a significant reduction in the overall computational cost, which is what we aim to use ML for.
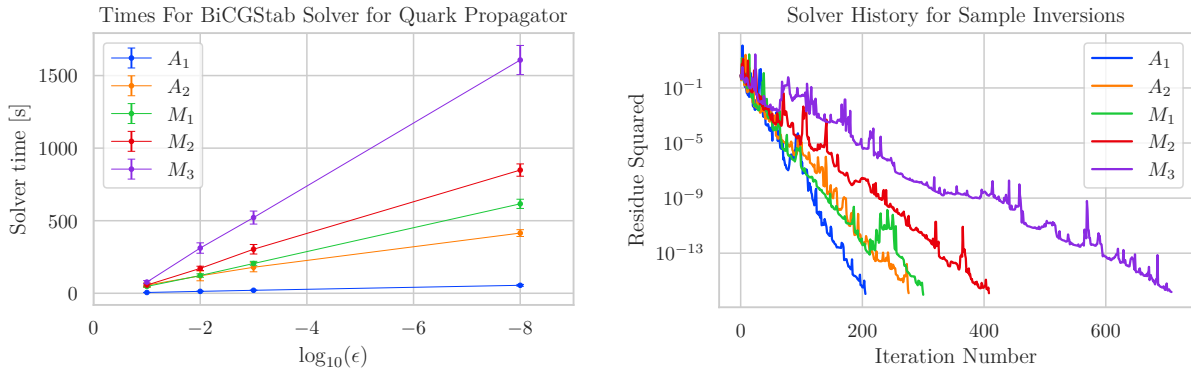
Figure 2.1: Left Plot: Average time to compute the quark propagator as a function of the stopping criterion for all the ensembles (see Table 2.1). To make the comparison fair, all the timings have been obtained using the same computational setup, i.e. the same number of nodes, cores per node and geometry. The error bar represents the standard deviation of the timings of the configurations in the ensemble. Right plot: Residuum as a function of the number of iterations for one sample configuration from each ensemble. Results shown in both plots are obtained using a BiCGSTAB algorithm.

For completeness, in the right plot, we show the iterated residuum as a function of the number of iterations for a representative gauge of all the ensembles listed in Table 2.1. It is clear to see that the trajectory of the solution is not monotonic with the iteration count, meaning that different local minima are explored by the solver, however, there is a general linear trend with the log of the precision.

The main idea of this work is to try to reduce the computational cost of the linear system for the quark propagator by using some information about the solver history on a subset of the gauge ensemble and then using supervised ML methods to speedup the computations on the rest of the ensemble. As a first try, we use numerical data for different stopping parameters ϵ as training and prediction data sets. For example, using a precise measurement of the propagator ($\epsilon = 10^{-8}$) on a subset of the ensemble and a less precise one ($\epsilon = 10^{-1}, 10^{-2}, 10^{-3}$) on the whole ensemble. This would dramatically reduce the time required to compute observables.

2.2   Machine Learning Regression for Two-point Functions

The field of Supervised Machine Learning is based upon the idea that for an arbitrary class of functions $\mathcal{F} = \{f(x|\theta) : \theta \in \mathbb{R}^l\}$, with $f(x|\theta) : \mathbb{R}^n \to \mathbb{R}^m$, parameterized by θ, and given a finite set of pairs $T = \{x_i, y_i\}_{i=1}^N$, one can find the best function $f(x|\theta)$ in the class that approximates the probability distribution that generated the pairs in T. This might remind the reader of regular fitting, and indeed supervised learning is a superset of fitting procedures. The generalization comes from the possibility of using very generic function classes $\mathcal{F}$ which can be very useful in cases where the function $F(x) : \mathbb{R}^n \to \mathbb{R}^m$ that governs a process of interest is unknown and maybe hard or even impossible to define rigorously. In such cases, if a sample of input-output data from such a function or process is available (this is called the *training set*), one can use supervised ML to at least get an approximation of $F(x)$ [26].

The procedure to obtain a valid approximation for $F(x)$ is divided into two parts: the choice of the class of function $\mathcal{F}$ or *model*, these are what are commonly known as ML methods or algorithms, and the optimization of the parameters θ, known in the jargon as *weights*.

The optimization is usually done with gradient descent or derived methods, mainly Stochastic Gradient Descent (SDG) [26] and related algorithms such as Adam [27] or AdaGrad [28]. All these optimization methods require a *loss function* to estimate the distance of the current approximation candidate $f(x|\theta*)$ from the known value for the training set. A simple choice is the *squared error loss*, or L_2-norm, $L_2 = \sum_i^N (y_i - f(x_i|\theta*))^2$, but other loss functions can be defined.

While the choice of the minimization algorithm, its input and parameters can affect the end results, the most important choice that impacts the model is the class of functions $\mathcal{F}$. For example, one can choose the set of linear functions from $\mathbb{R}^n \to \mathbb{R}^m$ and an appropriate number of coefficients represented in θ. If the L_2-norm is chosen while optimizing, then, in fact, the whole procedure is the familiar case of linear regression. Other types of functions can be used and there is a very active field of research in approximation theory to define the convergence properties of different function classes.

2.2.1 Artificial Neural Networks

An Artificial Neural Network (ANN or simply NN) is a function or model composed of a series of linear transformations alternated by non-linear operations. It has a set of unknown parameters, the *weights*, that schematically link sets of *nodes* that make up what are called *layers*. A very common subclass is that of *feedforward* NN, which means that layers are only connected in order, with their output being the input of only the following layer. Among these, the most popular one is the Multi-Layer Perceptron (MLP), which is NN made of a series of fully connected layers, see Figure 2.2.

It has been proven that MLPs are universal approximants [29, 30], meaning that any function can be approximated arbitrarily well with such NN, however, the number of layers and their sizes are not determined uniquely, so no information is known exactly on how to find the optimal approximant, just its existence.

The inner layers are called *hidden* because they do not represent any meaningful state on their own and hence are difficult to access and investigate them. Each node in a layer is based
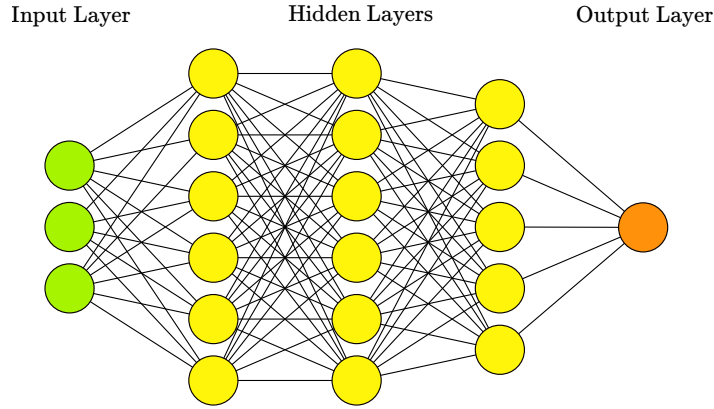
Figure 2.2: Example structure of a Neural Network for regression. The size of the input and output layers are fixed by the function that needs to be fit. The black lines represent the weights connecting the different nodes.

on the perceptron model, schematically drawn in Figure 2.3 Each node takes N-input values
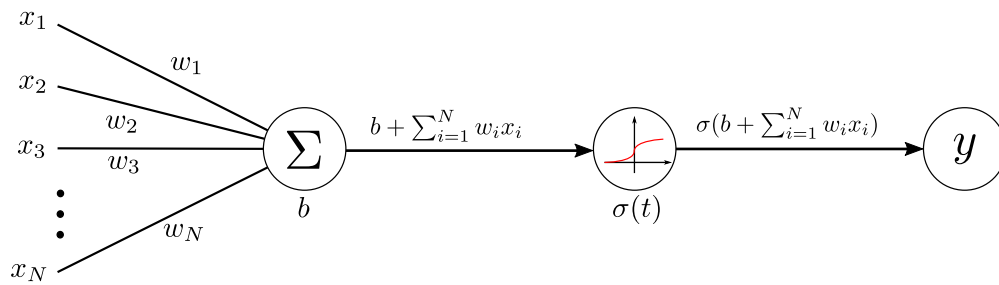
Figure 2.3: Neural Network unit used in this work. First a linear combination of the N inputs x_i with weights w_i is summed to a bias b, then to the result t is passes to a non-linear activation function, a sigmoid in this case.

$\{x_1, \ldots, x_N\}$ and models the final target y based on an N-sized vector of weights $\mathbf{w}$ and a *bias* b:

$$y = \sigma(b + \sum_{k=1}^{N} w_i x_i), \tag{2.4}$$

where the σ is the *activation function*, which is a non-linear function, some simple example of which are the sigmoid:

$$\sigma(x) = \frac{1}{1 + e^{-x}}, \tag{2.5}$$

or the Rectified Linear Unit (ReLU):

$$\sigma(x) = \begin{cases} 0 \text{ if } x < 0 \\ x \text{ if } x \geq 0. \end{cases} \tag{2.6}$$

56

A layer is then composed of a set of such perceptrons and we can then formulate the problem for a single layer in matrix notation:

$$\mathbf{y} = \sigma(\mathbf{b} + W \cdot \mathbf{x}), \tag{2.7}$$

Now we can generalize the result to more layers. For example a 3-layer MLP would be:

$$f(\mathbf{x}) = \sigma_3(\mathbf{b}_3 + W_{23} \cdot \sigma_2(\mathbf{b}_2 + W_{12} \cdot \sigma_1(\mathbf{b}_1 + W_{01} \cdot \mathbf{x}))), \tag{2.8}$$

which is now clearly a series of alternating linear and non-linear transformations on the input vector $\mathbf{x}$. Often, but in particular for the case of regression NN, the outermost function σ is taken to be an identity.

The number of layers and their sizes are not fixed and have to be chosen wisely: these are examples of what are called *hyperparameters*. Other empirical choices are, for example, the type of activation function for each layer, the initial values for the weights and biases before the minimization is run or the minimization algorithm and its parameters, see Appendix 2C.

To find the optimal weights and the biases of our NN we use the backpropagation algorithm, which is nowadays the workhorse for fast learning in NNs. The backpropagation algorithm became popular after the article by Rumelhart et al. [31], but it has its origin in the Master Thesis results of S. Linnainmaa [32].

The backpropagation algorithm is the crucial element of all Gradient Descent methods as it allows the calculation of the partial derivatives of the loss function of all the weight and biases without having to store any data or having an explicit functional for the derivatives. It is one of the most important examples of dynamic programming.

In general, the optimization of the parameters of a NN is a highly non-trivial task of convex optimization. One of the main issues is the presence of several local minima of the loss function. In practice, one can only test on some sample data whether the fitted parameters give a good representation of the data or not. This is made even worse by the risk of over-fitting the model to the training data, which is exacerbated by the usually very large number of parameters. To mitigate this problem, often, a subset of the training data is not used for fitting the parameters

but for *validation* or *bias correction.* Another way to mitigate over-fitting is by imposing some constraints on the values of the parameters, by, for example, modifying the loss function, see Appendix 2C.

2.2.2 Partitioning of the data

As previously stated, in this work, we want to train a NN to model the trajectory of the BiCGSTAB solver from one or more low precisions to a high one. In order to do this, we compute the hadron two-point functions $C(t)$ on all the N configurations of each ensemble in Table 2.1. These correlation functions are obtained from propagators that have been determined with four different stopping criterion: $\epsilon = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-8}$.

Out of the N configurations, we choose a fraction $f = \frac{N_L}{N}$ that will be used for training our ML model. We will call this partition the *labeled fraction* because for each configuration, we know both the low precision value of the correlation function and the *target* value, $\epsilon = 10^{-8}$ that we want to reproduce during the training.

The test set, from which the ML estimator for the correlator is computed, is chosen as the remaining $(1 - f)N$ configurations of the ensemble. The data for the precise correlator ($\epsilon = 10^{-8}$) on this set is *never* used in the ML process. It is only used to compare the ML predictions with a full-length traditional calculation and thus to test the efficiency of the model discussed.

The labeled configurations are further partitioned into two sets of size N_T and N_B. The first are the ones used for training the NN model, while the second is used for bias correction, as done for example in [13] and shown more in detail in Section 2.3.1. Figure 2.4 shows a simple diagram of the data partitioning for clarity.

2.2.3 Single Point Correlator Models

The set of ML models considered for this study all focuses on finding relations between a hadronic correlator $C(t)$ at low precisions and the precise one. In what we will call *Single Point Neural Network* models (SPNN), the relation is found for every source-sink separation time independently.
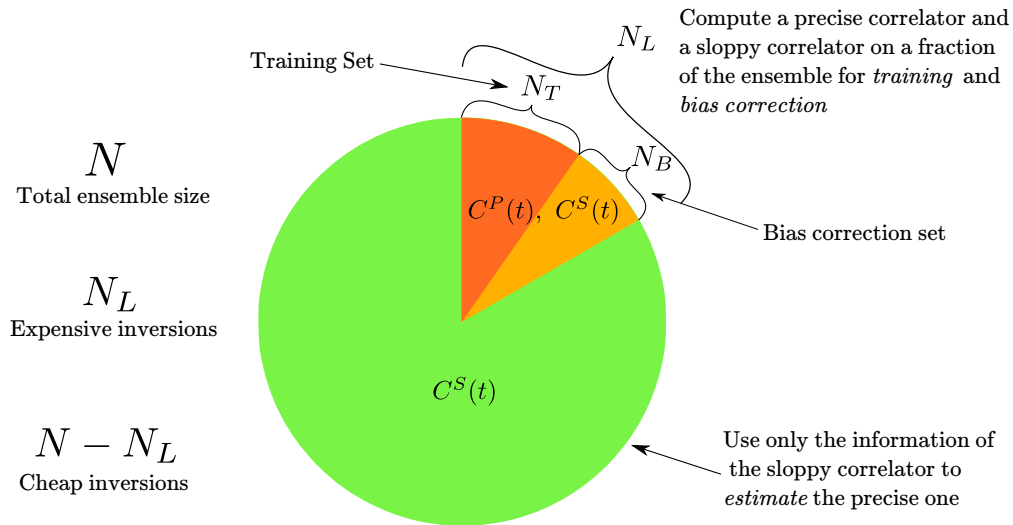
Figure 2.4: Simple diagram on how the data is partitioned. On the whole ensemble the "sloppy" correlator $C^S(t)$ is computed from propagators with stopping criterion $\epsilon = 10^{-1}, 10^{-2}, 10^{-3}$. On just N_L configurations the "precise" correlator $C^P(t)$ is computed with $\epsilon = 10^{-8}$.

It effectively corresponds to training T different ML models, one for each value of t, for every hadronic correlator. There is much freedom in the choice of the model, as one can choose different input variables while keeping the target variable fixed. However, one has to consider that increasing the dimensionality D of the problem can affect the convergence.
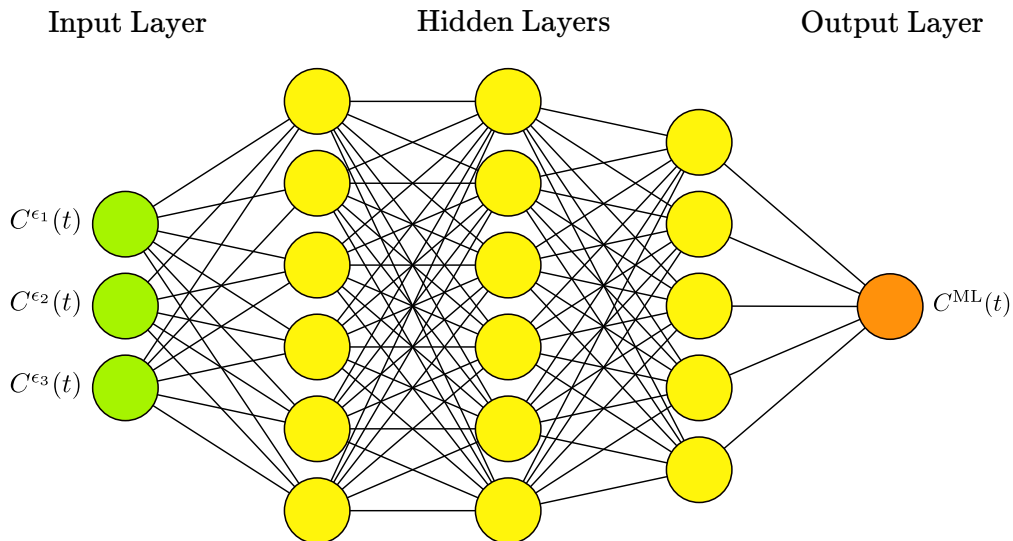
Figure 2.5: Schematic representation of a NN for the model with multiple ϵ, SPNN2. It is important to note that all layers in this model are rather small.

Here is the list of single-point models used in this work:

- **SPNN2**: a Neural Network using $\epsilon = 10^{-1}, 10^{-2}$ as input and $\epsilon = 10^{-8}$ as target. This is done independently for every source-sink separation time t, i.e. a neural network is created and trained for every t.

- **SPNN3**: same as SPNN2 but using $\epsilon = 10^{-1}, 10^{-2}, 10^{-3}$ as input and $\epsilon = 10^{-8}$ as target.

- **SPNN2t1/SPNN3t1**: a Neural Network using $\epsilon = 10^{-1}, 10^{-2}$ (and 10^{-3}) at time t and $t \pm 1$ as input and $\epsilon = 10^{-8}$ as target. The difference with SPNN2/SPNN3 is that the input given to the NN also contains data from neighboring source-sink separation times t.

- **SPNN2t2/SPNN3t2**: a Neural Network using $\epsilon = 10^{-1}, 10^{-2}$ (and 10^{-3}) at time t, $t \pm 1$ and $t \pm 2$, as input and $\epsilon = 10^{-8}$ as target.

It is clear that the dimensionality of each model space is then $D = (2n + 1)N_\epsilon$, where N_ϵ is the number of stopping parameters used as input and n is the number of neighboring time points used. In Figure 2.5 we show the example structure of the NN corresponding to the SPNN3 model. To fix the size of the hidden layers, we followed a semi-empirical method that gradually reduces the size of the network until its performance decays. More details are in Appendix 2C.

2.2.4   Global Correlator Models

A more radical approach is to create a model to directly relate a low precision correlator, considered as a vector of size T, or $T \times N_\epsilon$, if multiple stopping parameter data is used as input at once, to a high precision one. Clearly, the dimensionality of the problem is much larger than the ones in the previous section, as now the ML model represents a function $f : \mathbb{R}^{T \times N_\epsilon} \to \mathbb{R}^T$. Depending on the maximum ϵ used, we name these models **GNN2** and **GNN3** in analogy with the single point models. In Figure 2.6 we show the example of a NN corresponding to the GNN2 model. The size of the hidden layers has been set following the procedure in Appendix 2C.
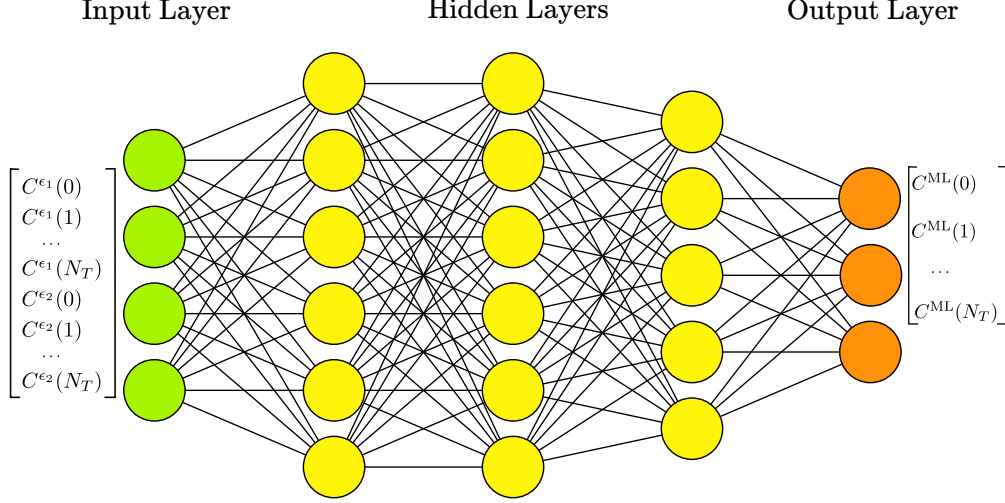
Figure 2.6: Schematic representation of a NN for the global correlator model. This particular NN corresponds to the GNN2 model. The input layer is of size $T \times N_\epsilon$ while the output layer is of size T. The size and number of intermediate hidden layers are parameters of the model differ from the ones in the picture.

2.3 Accelerating Correlator Calculations with ML

The first consideration to ensure the feasibility of our ML procedure is to check if there is some degree of correlation between the hadron correlators at different precisions. For notation's sake, given a gauge configuration U_i and a propagator evaluated with accuracy $\epsilon = 10^{-n}$, we label the correlator as as $C_n(t; U_i) = C_n^{(i)}(t)$. The average over the ensemble is denoted as simply $C_n(t) = \frac{1}{N} \sum_{i=1}^{N} C_n^{(i)}(t)$.

To determine the correlation among the 2-point functions evaluated at different precisions we define the usual sample correlation coefficient as:

$$r_{mn} = \frac{\sum_{i=1}^{N} \left(C_m^{(i)}(t) - C_m(t) \right) \left(C_n^{(i)}(t) - C_n(t) \right)}{\sqrt{\sum_{i=1}^{N} \left(C_m^{(i)}(t) - C_m(t) \right)^2} \sqrt{\sum_{i=1}^{N} \left(C_n^{(i)}(t) - C_n(t) \right)^2}}. \tag{2.9}$$

In the top plot of Figure 2.7 we show a heat map of r_{28} for the ensemble M_2 as a function of the Euclidean time separation between the two interpolating operators t/a for selected hadrons. In the bottom plots, we plot the actual value of $C_2(t; U_i)$ against $C_8(t; U_i)$ for two different values of t for the pion: a small value, where the excited states contribution should still be large (bottom left), and an asymptotic one where the ground state should be dominant (bottom right). We

61

clearly observe a high level of correlation for most of the 2-point functions. The correlation though, decreases towards the center of the lattice, where the signal-to-noise decreases. One could phenomenologically conclude that one starts to lose the signal when the accurate solution of the quark propagator is statistically uncorrelated to the quark propagator obtained during the first steps of the iterations. If we consider the iterative nature of the KS solvers, somehow, the loss of correlation further away from the source location is expected. The higher the accuracy requested, the higher the number of iterations and the larger the distance from the source where the solution is accurate. Nevertheless, the clear correlation, which to first approximation is linear even in the large t case, can be exploited to extract information about the precise correlation starting from the sloppy one.
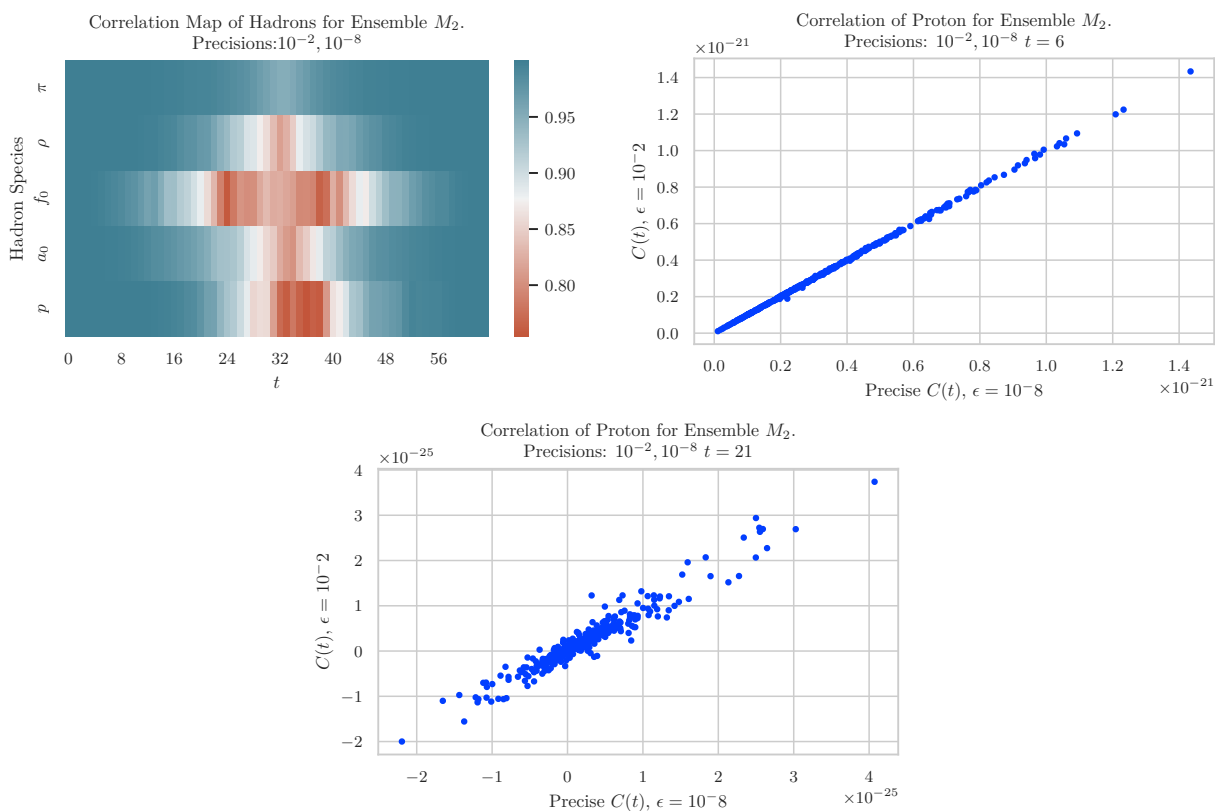
Figure 2.7: Top: The correlation coefficient between the correlator calculated at $\epsilon = 10^{-2}$ and $\epsilon = 10^{-8}$ for various hadrons as a function of the source-sink separation time. Bottom: the proton correlator at two precisions ($\epsilon = 10^{-2}, 10^{-8}$) plotted against each other for two different values of t.

In Figure 2.8, we show all the data that is used as input for our calculations, both in the form

of the correlator and as the effective mass plot, where central values and error bars are calculated using the information of all configurations of the ensemble utilizing the bootstrap method, see Appendix 1A.2. Our ML procedure intuitively has the aim of finding a mapping from lower ϵ data to the high precision one (the red data in the plots) at the configuration level, exploiting the correlation between different precisions.

One thing to notice is that the data for $\epsilon = 10^{-3}$ is overlapping very closely with the precise results of $\epsilon = 10^{-8}$. This is the reason why it has been decided to consider the pairs of models SPNN2, SPNN3 and GNN2, GNN3 to check if the results are significantly different when the $\epsilon = 10^{-3}$ data is included. One has to consider, however, that there is a computational cost difference between $\epsilon = 10^{-3}$ and $\epsilon = 10^{-2}$, so this, as we will see in Section 2.4.1, needs to be accounted for when comparing the results.
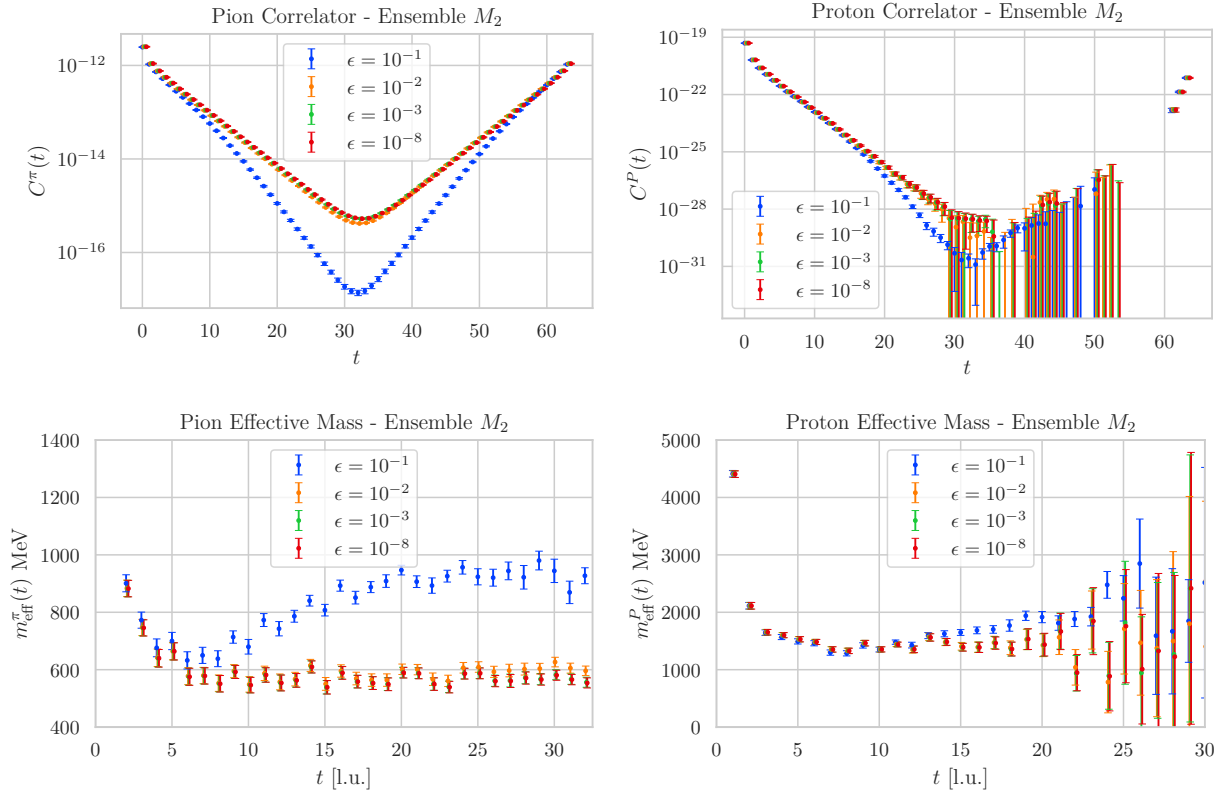
Figure 2.8: Top row: Two-point correlator for the pion (left) and proton (right). Bottom row: effective mass for the pion (left) and proton (right). Different data series represent the varying stopping criterion ϵ. All data are for the ensemble M_2.

2.3.1 Bias Correcting Procedure

Throughout this analysis, we employed a technique to remove the bias of the high precision ML estimates as outlined in [33], by considering the chance that the low-precision data for an observable O^{LP} and the high-precision O^{HP} one could have different expectation values. Calling O^{*HP} the high precision estimate from the ML model using the O^{LP}, one then considers the expectation value for O^{HP}:

$$\langle O^{HP} \rangle = \langle O^{*HP} \rangle + \langle O^{HP} - O^{LP} \rangle. \tag{2.10}$$

. The second term is what we call the *bias correcting term.* It can be estimated with high precision from a small sample of data when the fluctuations of O^{HP} and O^{LP} are highly correlated, which we assume is the case for the problem we have at hand because the underlying gauge dynamics affect the correlator results at different precisions in a similar way. The unbiased estimator for the observable then becomes:

$$\bar{O}^{HP} = \frac{1}{N - N_L} \sum_{i=0}^{N-N_L} O_i^{*HP} + \frac{1}{N_B} \sum_{j=1}^{N_B} (O_j^{HP} - O_j^{LP}). \tag{2.11}$$

It is important to note that two sums run on different partitions of the ensemble data, as shown in Figure 2.4. The first one is for the $N - N_L$ unlabeled configurations on which only the low precision data is known, and the value of O^{*HP} is calculated using the ML model. The second sum runs on the bias correction subset of the labeled fraction of the ensemble, for which both O^{HP} and O^{LP} are known.

This bias correcting procedure is the same as in the TSM method of [17], but here it is done at the ensemble level and not for a single configuration.

2.3.2 Error determination

A popular method to improve the statistical analysis of lattice QCD data is the bootstrap resampling method [34]. Since, in this work, we present both the results coming from both a "traditional", or "fully precise", calculation and ones coming from a ML-aided method, it is important to specify how the bootstrapping of data is performed in the two cases.

The data of the traditional calculation come from bootstrapping the results of the correlator $C(t)$ from each configuration. As usual, in order to preserve the correlations between the data, one needs to apply the same sampling to all values of t. Throughout this work, we used $N_{\text{boot}} = 1000$ samples of size N (the number of configurations in the ensemble) for this case. The values for the variance of all the expectation values presented come from the variance of the bootstrap sample averages. More details are found in Appendix 1A.2

The determination of the uncertainty coming from the ML procedure is more involved. Crucially one need not mix the labeled data with the unlabeled data. Our first step was then to create this partition, even though in an actual application of this method, this would not be necessary as the precise calculation data would not be available for all configurations. Each subset of the data is then bootstrapped independently, but within the same partition, all the data for different the precisions, including the precise one, are sampled using the same random draws for the bootstrap, which are also maintained across different values of t, to preserve the correlations between the data. The bootstrap samples for the labeled data are then used to train N_{boot} neural networks (all with the same structure, obviously) and to calculate the bias correcting term. To each of these NN a bootstrap sample of unlabeled data is assigned and the ML estimator for each NN is computed, so the sample expectation value of an observable is then is the average of the bias-corrected NN predictions on the sampled variables. Finally, as is standard for the bootstrap method, the final estimator and its variance are the average of the sample expectation values and their variance respectively.

2.4 Numerical Results

We tested all of our models on the ensembles from the PACS-CS collaboration presented in Table 2.1 to understand the feasibility of our ML procedure for hadron correlators. Each model was trained using bootstrap, as presented in Appendix 1A.2, so that the results presented below are all bootstrap averages of different NN.

2.4.1 Performance Metrics

To get a quantitative estimate of the performance of the algorithms, we define a *computational-time gain factor* Γ^{ML} for every model. It depends on the fraction f of the ensemble used for training the NN and on the minimum ϵ used as input for the model. If we denote with t_P the total time that one would need to determine the quark propagator with the highest accuracy, $\epsilon = 10^{-8}$ (the traditional calculation), the relative time gain factor is

$$\Gamma^{ML}(f) = \frac{t_{\epsilon_{\min}}}{t_P}(1-f) + f \, . \tag{2.12}$$

This is the sum of the relative total time required to obtain the correlator with the highest low precision $\epsilon_{\min}$, either 10^{-2} or 10^{-3} depending on the model, on the $N - N_T$ configurations where only the low precision is needed, $\frac{t_{\epsilon_{\min}}}{t_P}(1-f)$, added to the training fraction f itself. Γ^{ML} is not a time but rather a ratio of times representing a gain; it is bounded by 1, when the time to compute the low precision correlator t_ϵ is equal to the precise time t_p or when $f = 1$, but since in general $t_\epsilon \leq t_p$, then $\Gamma^{ML} < 1$. The smaller Γ^{ML}, the larger the computational gain. In computing $\Gamma(f)$ we neglected the time spent for training the NN, which is negligible when compared to the propagator calculations: it is of the order of ≈ 10 minutes on a regular laptop.

In order to assess the quality of the results, we computed three quantities for each model:

- the ratio of the variances $\frac{\sigma^2_{ML}}{\sigma^2_E}$ between the ML results and the high precision ones

- the compatibility of the results $\frac{|\langle O \rangle_{ML} - \langle O \rangle_E|}{\sqrt{\sigma^2_{ML} + \sigma^2_E}}$

- the *algorithmic efficiency*, what we define as $AE(f) = \frac{1}{\Gamma^{ML}(f)} \frac{\sigma^2_E}{\sigma^2_{ML}}$. This combines the time gain factor and the loss of precision in the variance. The higher this quantity is, the more efficient it is to use the ML algorithm as opposed to perform a traditional calculation.

These have been computed for multiple observables, namely for the correlator $C(t_1)$ at a fixed small euclidean time $t_1 \approx N_T/10$, the correlator at a fixed asymptotic time $C(t_2) \approx N_T/3$ and for the effective mass calculated using a $\cosh$ fit in the case of mesons and a constant fit for baryons.

2.4.2 Single Point Models

In the following pages, we show the results for the variance ratio, the compatibility of the results and the algorithmic efficiency for all the ML models we presented for both the pion and proton data on the ensemble M_2. There are three plots for each quantity as it is measured on t_1, t_2 and for the effective mass. For clarity, with "fully precise" (FP) results, we mean the observables computed over all configurations of the ensemble with a high precision propagator.

The first observation to make is that all single-point models behave very similarly. The discrepancy between the SPNN3 class of models and the SPNN2 class is more evident, while increasing the number of time points in the model, so going from SPNN2 to SPNN2t1 and SPNN2t2 or the equivalents for SPNN3, does improve the results but not as much as increasing the number of precisions used in the model. This is very clear at large t and for the effective mass. There appears to be a larger gain in adding time points to the SPNN2 model than in the SPNN3 model.

One thing that is apparent from Figure 2.9 is that the ML results are always compatible with the FP results, with compatibility consistently less than 0.8, so the central value obtained from ML is always well within 1σ of the precise one.

A more interesting observation can come from looking at the variance ratio, Figure 2.10: we observe the variance for single points being always very close to the FP for sufficiently large f, but it is much higher for the effective mass. The fact that for low f the variance ratio is large is indicative of the inability of the ML models to capture the relation between the data fully. At large f, one would expect the variance ratio to rise, as less information can be extracted from the data and the effect of increasing f only results in reducing the sample size. We observe this behavior clearly for the pion data at t_1, where in fact, we expect the convergence of the solver to be extremely fast as it is close to the source. At the same time, for other quantities and for the proton, the effect is small, but however the variance ratio seem to plateau on a fixed value.
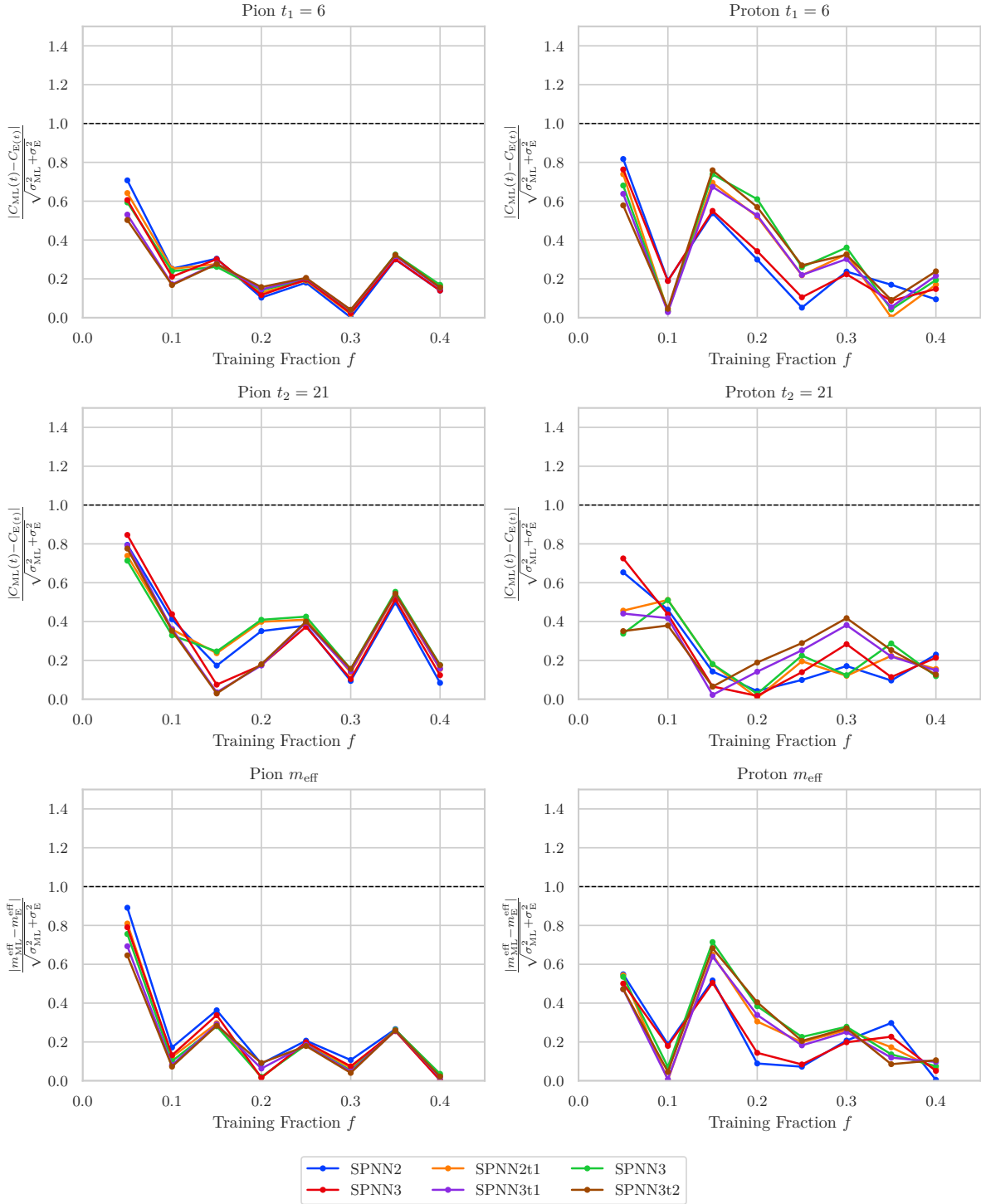
Figure 2.9: Compatibility between the ML results, using SPNN methods, and the fully precise results for the ensemble M_2 as a function of the training fraction. The top row is the correlator at time $t_1 = T/10$, the middle row is the correlator at time $t_2 = T/3$ and the lower row is the results of a constant fit for the effective mass.
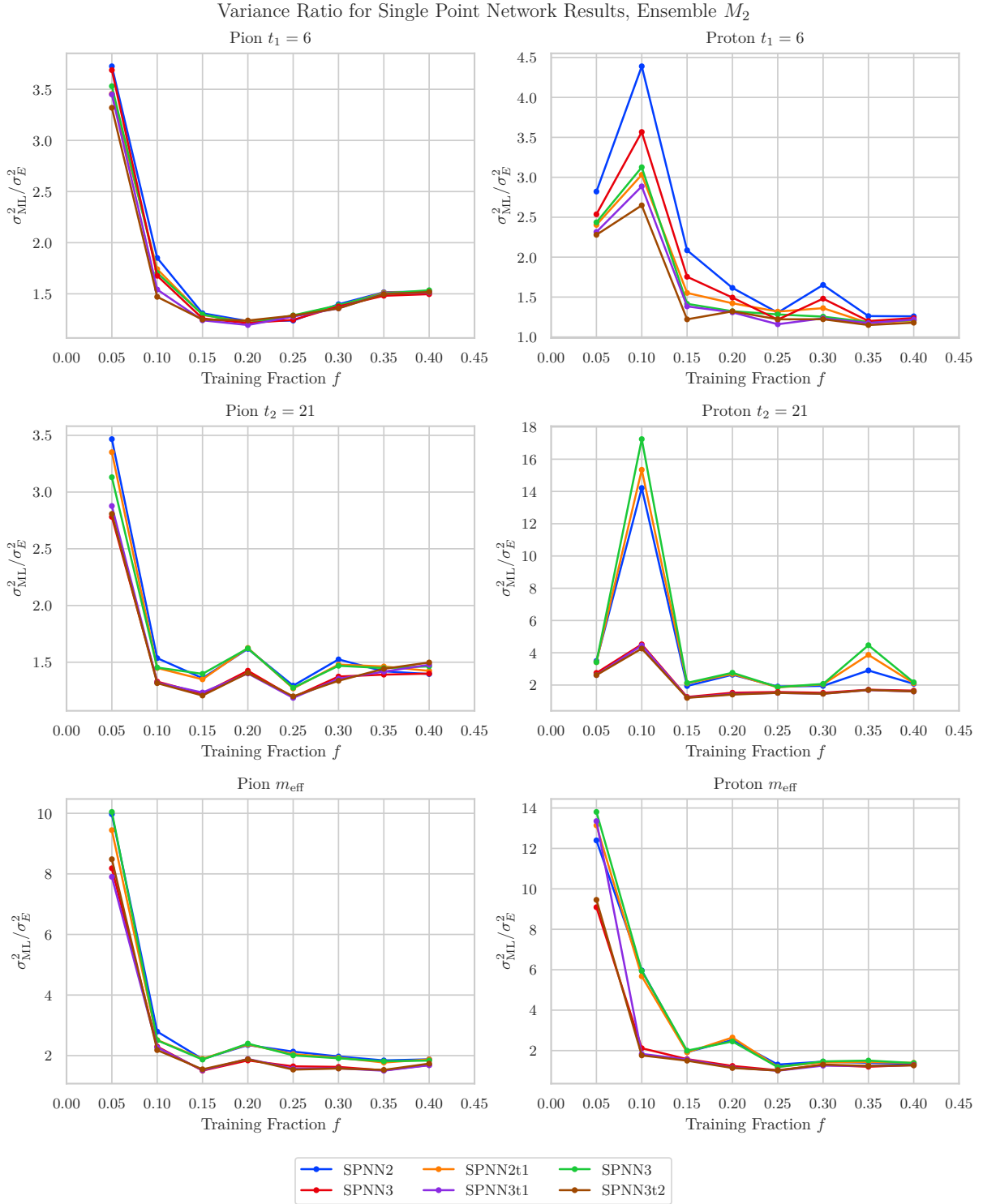
Figure 2.10: Variance ratio between the ML results, using SPNN methods, and the fully precise results for the ensemble M_2 as a function of the training fraction. The top row is the correlator at time $t_1 = T/10$, the middle row is the correlator at time $t_2 = T/3$ and the lower row is the results of a constant fit for the effective mass.
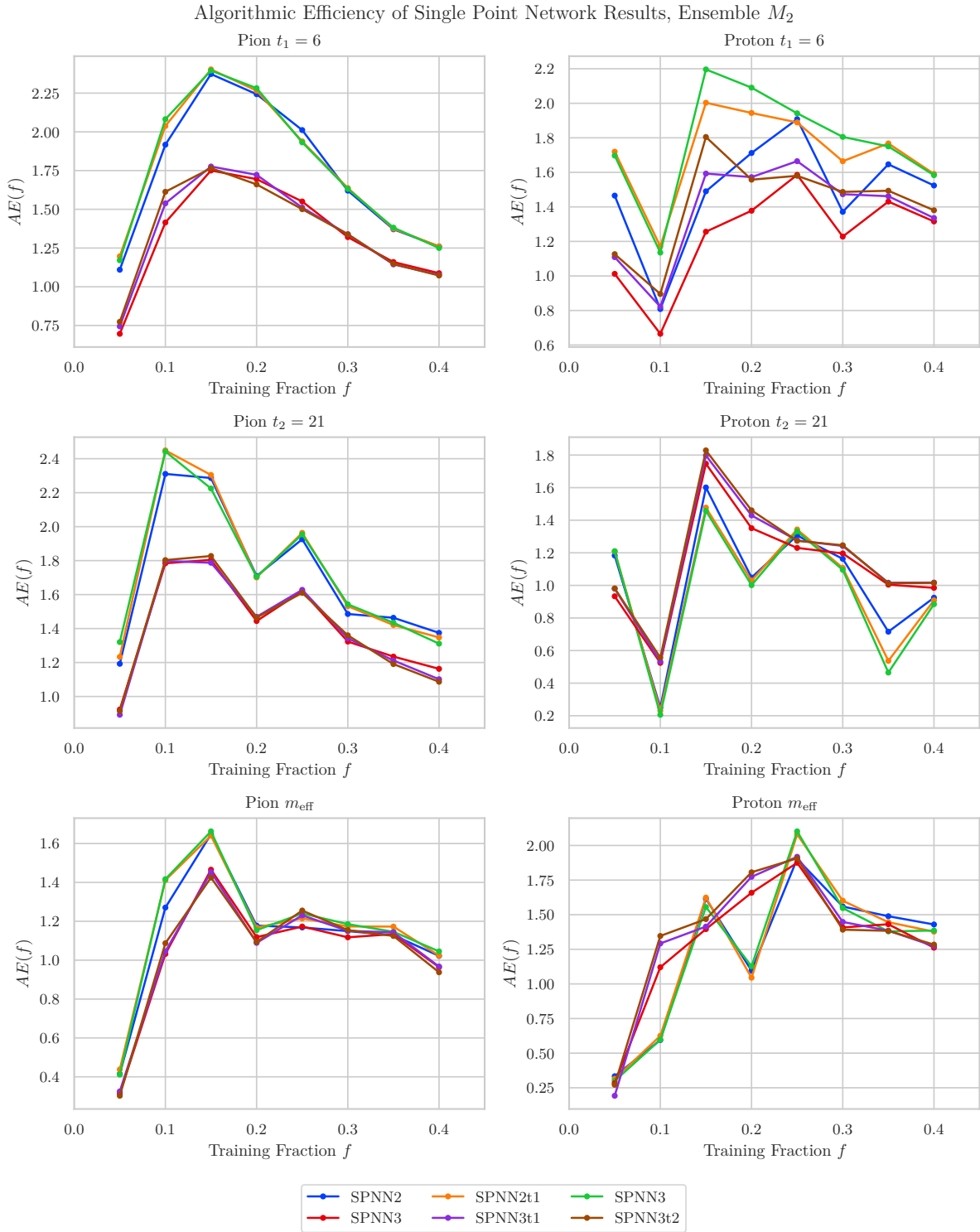
Figure 2.11: Algorithmic efficiency of the ML results, using SPNN methods, and the fully precise results for the ensemble M_2 as a function of the training fraction. The top row is the correlator at time $t_1 = T/10$, the middle row is the correlator at time $t_2 = T/3$ and the lower row is the results of a constant fit for the effective mass.

When looking at Figure 2.11, the AE, however, it is clear that SPNN2 models lead to better efficiency, as their Γ, the time gain factor, is much smaller because they use propagators only up to $\epsilon = 10^{-2}$ which are cheaper. The AE is also an effective measure to discern the optimal training fraction; it is clear that larger training fractions do improve the variance slightly but come at a higher computational cost. From our data, one can infer that for the pion, the optimal range for f is somewhere in the interval $0.10 - 0.25$, while for the proton, a larger training fraction is needed, in the range $0.15 - 0.30$. It is also interesting to see that for t_2 and for the effective mass of the proton, the AE is higher for SPNN3 than for SPNN2 and that the AE also is smaller than 1 for some combinations of model and f. This could indicate that the relation in the region where the signal-to-noise starts to degrade, one is forced to use higher precision solvers.

In general, we find that there are wide ranges of f where the algorithmic efficiency is larger than 1, meaning that the ML learning method is an effective way to speed up the calculation of hadron two-point correlators and the effective mass, with the best case being the pion for which a computational gain of more than a factor 2 can be consistently gained.

There are, however, some critical challenges to this ML approach. The first is the wild fluctuations of the results shown in the figures above. The ML training procedure works well on average, but it seems to fail catastrophically for some setups, for example, one can look at the proton variance ratio for $f = 0.10$ for t_2 in Figure 2.10. For all SPNN2 and SPNN3 models, there is a sharp spike, also present moderately at t_1 but not for the effective mass. This could be due to the sample of data used for training since it is present irrespective of the model chosen. One possible source of the issue can be hypothesized by the compatibility, which for the same point is strikingly low. This means that the ML model is fully capturing the relation between the low and high precision data, but it is failing to reproduce its correct distribution; hence this could be a case of over-fitting. However, given the small sample size and the small size of the ensemble in general, this can be a hard problem to overcome.

The second issue comes from the difference in the variance ratio of the single point of the

correlator $C(t_1)$ and $C(t_2)$ and the effective mass. For the SPNN models, the variance is always much larger for the effective mass than it is for the correlator for sufficiently large f, where there the ML models have converged. This potentially indicates a loss of correlation between the data at different t in the final ML results, which is problematic when, for example, performing fits to the data like in the case of the effective mass. This is naturally given by the fact that we are training independent models for each value of t, but while the input data is prepared in such a way that correlations between the data are preserved, in practice, the bootstrap samples used for training at different values of t all resample the data with the same random sequence, the output data depends on the training and the inner workings of the NN. This makes it so that the predicted values of the correlator coming from the ML model on a single configuration might be decorrelated at different t. We can see, however, that this effect is mitigated by adding the data for more values of t as input to the NN, which is the reason why SPNN2t1 and SPNN2t2 have better performance than SPNN2, and the same for the SPNN3 class. Adding the additional t increases the information about the time correlation in the NN, but from our results we can conclude that this is not enough to estimate m_{eff} correctly, for example.

In Figure 2.12 we show the results on different ensembles for the SPNN3t2 model, which is the one that uses the most data and, in general, performs best, even though it has a lower time gain factor than the SPNN2 models. We observe no significative difference in behavior as masses, volumes and lattice spacings are changed, which is a good indication that this ML procedure can be applied safely for a wide range of physical parameters for the lattice gauge field ensembles.

In general, the SPNN models are an effective method for reconstructing the hadron correlators and their distribution from low precision data. However, there are concerns that should be addressed for they practical use for calculations of secondary quantities like the effective mass.
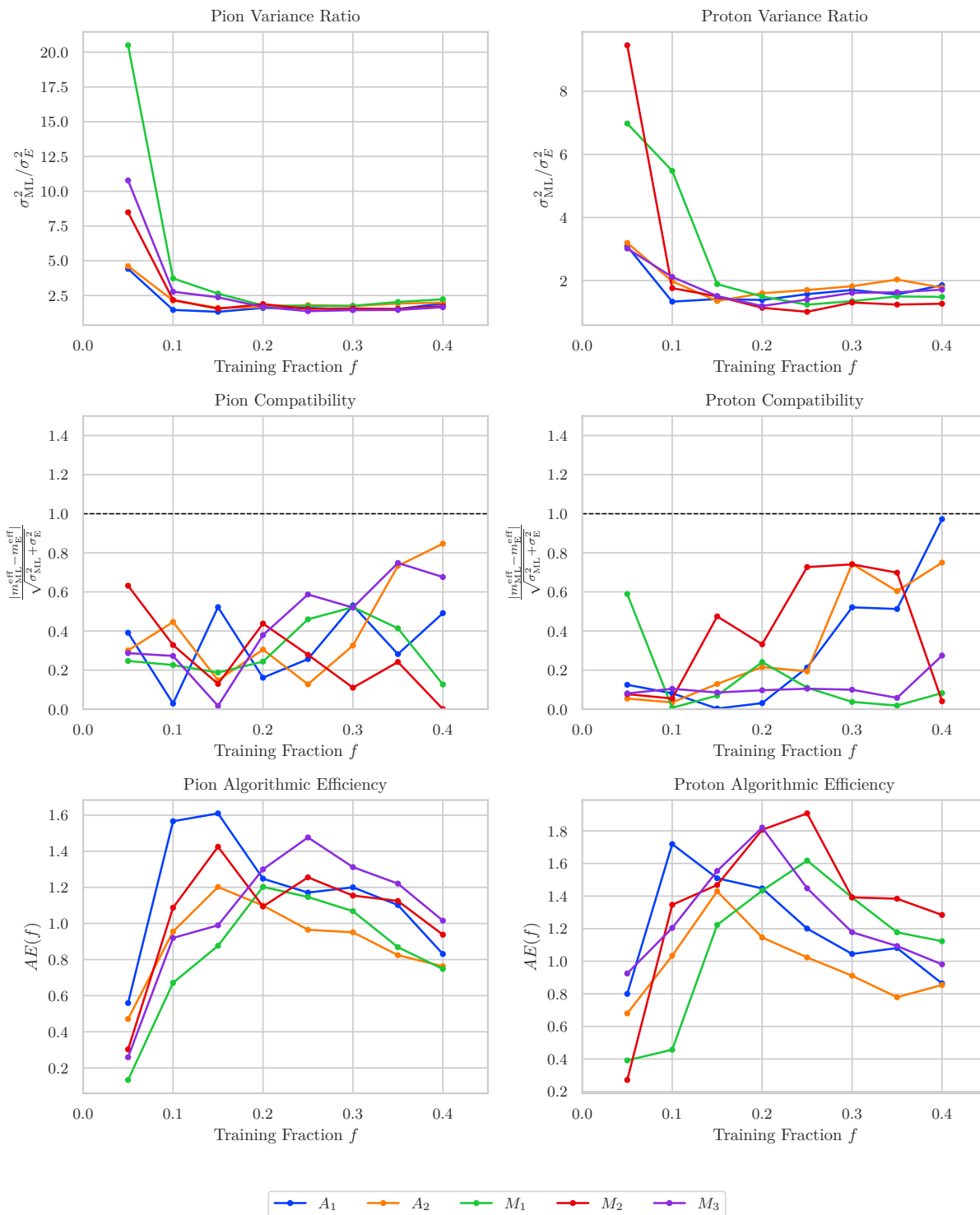
Figure 2.12: Variance Ratio (top), Compatibility (middle) and Algorithmic efficiency (bottom) of the ML effective mass results, using the SPNN3t2 method, and the fully precise results for the ensembles of Table 2.1 as a function of the training fraction.

2.4.3 Global Network Models

The global network models are clearly much more complicated to train than the single-point networks. For a start, their input and output layers are of the order of the number of lattice sites in the temporal dimension, but for ML models, this is not an unprecedented size. What is potentially problematic is the nature of the data itself, in particular, the exponential decay of the correlator as t increases: the data spans several orders of magnitude and can, in exceptional cases, be negative, which prevents a logarithmic treatment of the data. As explained in Appendix 2C, the way to solve the issue is to scale all the data at a given time slice to have unit variance and zero mean. While this procedure has no consequence in the SPNN case, in GNN models, this effectively means that the network is only left to learn the correlations between the data and not the general behavior of the function. In the following figures, we show the results for the effective mass from GNN3 on all ensembles; the results for GNN2 are very similar, so they have been omitted for clarity of the figures.

The most striking difference between SPNN and GNN models is the inversion of the roles of the variance and of the compatibility. The results have, in fact, a very small variance ratio when compared to the fully precise calculation but end up being a few σ away from the true results. The low variance is a clear sign of the network's inability to fully capture the distribution of the correlator at every point t, while the low compatibility of the results implies that also the central value of the distribution is incorrect, at least in the region where the effective mass fit is performed. This also points to another problem coming from the scaling of the data: GNN models in no way can distinguish points, in the case of the proton, where the signal-to-noise ratio is large (small t) or small ($t \approx T/2$). So the effectiveness of the training is significantly reduced. This is, however, not an exhaustive explanation for the effect as the pion is also affected.

An apparent limitation of the GNN models is the extreme fluctuations of the results, especially of the compatibility. From the bottom two panels of Figure 2.13, we observe that, for example, the compatibility of the m_{eff} for the pion on the A_2 ensemble oscillates from ≈ 7 to almost 0, when changing f from 0.1 to 0.2. This is a clear indication of the instability of the method.

One last consideration to make is that the GNN does give a small variance for the effective mass, as opposed to the SPNN models, implying that the correlations between different t are indeed preserved, although around the wrong central value and with a too small variance.

Overall, GNN models do not seem suitable to the problem at hand, as they fail at reconstructing the hadron correlator distribution properly.
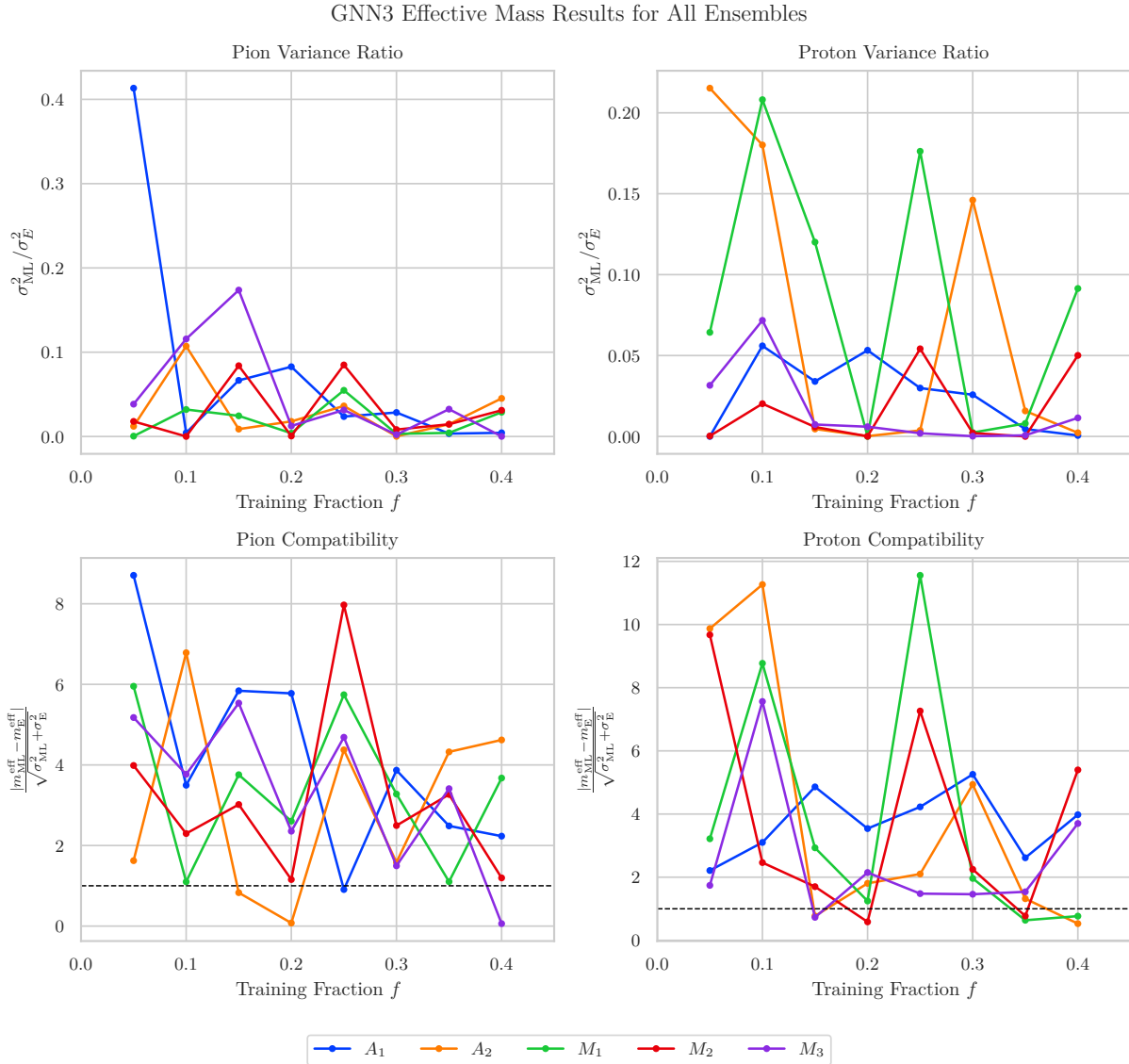
Figure 2.13: Variance Ratio (top) and Compatibility (bottom) of the ML effective mass results, using the GNN3 method, and the fully precise results for the ensembles of Table 2.1 as a function of the training fraction.

2.5   Outlook

Currently, the feasibility of this ML procedure is being studied. Some preliminary results seem encouraging, however, there is the question of how reliable the proposed algorithm is. We believe that there is certainly some information to be extracted from the solver history in the determination of quark propagators and this can be used to speed up the calculation. What we have found is that applying ML to derived quantities like the two-point correlation function or the effective mass is not sufficient to reach a degree of confidence in the results that is satisfactory for us. This is especially true as, in practice, there is always the possibility of performing the fully precise calculation, which is "only" more expensive numerically, but certainly more stable and reliable. However, if we want to adopt this ML routine, we would have to be confident that the computational gain comes at no expense of the final precision of our results.

One possible way forward would be to test the limits of validity of this ML procedure further, i.e. going to larger volumes or to smaller pion masses. In those ranges, the numerical cost increases dramatically, which might make the algorithmic efficiency steadily large. This would effectively mean that the time that is saved from calculating only low precision propagators could be spent on sampling more sources on the same gauge field.

This last point also suggests another possible improvement for our derived quantities approach, which is to use multiple stochastic sources per configuration and perform the ML training on the average of those or us them to effectively enlarge the training set size. However, this would shift the usefulness of the algorithm towards the possibility of only being applied for relatively simple derived quantities, as most of the information on the single linear system solve would be averaged out. This approach would then effectively be a variation of the TSM.

A more interesting development would probably come from accessing the propagator directly during the iterative solver and training a model on the full quark propagator as opposed to the correlator. However, this is very hard to implement in practice for two reasons. The first is that the current LQCD software that computes propagators is highly optimized (mostly GPU) code, which makes it hard to modify efficiently. The second hurdle to overcome would be the large

size of the network needed for training, as the quark propagator is a very large object in terms of numerical size (it has a size proportional to the lattice size), which would make the training of such model very expensive and challenging.

One last attempt would also be to try to apply different ML strategies, like analytic continuation or unsupervised learning, but those would have to be radically different approaches from what we have presented.

Nevertheless, utilizing the solver trajectory to gain information on the quark propagator and speed up its calculation is something that should be explored further, not just for lattice QCD applications. The potential impact of this procedure to compute hadronic observables could also possibly be adapted to the speed of the calculation for pseudo-fermions in the Markov Chain Monte Carlo algorithms and, more in general, it could enable the calculations on large lattices, where the Dirac operator is prohibitively large for current computing architectures.

APPENDICES

LINEAR ALGEBRA CONVENTIONS AND DEFINITIONS

Most of the conventions regarding linear algebra are taken from [25]. Here we report them for clarity and to fix the notation.

2A.1 Conventions on Norms

We consider a vector space $\mathcal{V}$ over the complex field $\mathbb{C}$ endowed with an inner product $(\cdot, \cdot)$ satisfying the properties of positivity, linearity and Hermitian symmetry. The norm on any element of $\mathcal{V}$ is an application from the vector space to the real field $\mathbb{R}$ satisfying the properties of positivity, homogeneity and the triangle inequality. Given $v \in \mathcal{V}$ examples of norms on $\mathbb{C}^n$ are

$$||v||_\infty = \max_{i \in [1:n]} |v_i|\,, \tag{2A.1}$$

$$||v||_p = \left[\sum_{i=1}^n |v_i|^p \right]^{1/p}\,, \quad p \geq 1\,. \tag{2A.2}$$

The corresponding vector spaces are denoted respectively as L_∞^n and L_p^n. In the $p = 2$ case the vector space L_2^n is endowed with an inner product and the Euclidean norm is defined to be

$$||v||_2 = \sqrt{(v, v)}. \tag{2A.3}$$

We consider now a vector space $\mathcal{M}_n$ of $n \times n$ matrices. Also in this case a matrix norm can be defined satisfying the usual properties and additionally the property of submultiplicativity, i.e. $||AB|| \leq ||A||\,||B||\,, \ \ \forall A, B \in \mathcal{M}_n$. A matrix norm can be induced by a vector norm defined on $\mathbb{C}^n$ as follows

$$||A|| \equiv \max_{||v||=1} ||Av||\,, \tag{2A.4}$$

where it should be clear when we talk of a matrix norm or vector norm. The matrix norm induced by the norm on L_2^n is related to the spectral radius

$$\rho(A) \equiv \max \left\{ |\lambda|, \lambda \text{ ev. of } A \right\}\,, \tag{2A.5}$$

79

as follows

$$||A||_2 = \sqrt{\rho(A^\dagger A)}\,. \tag{2A.6}$$

2A.2 Preconditioned Krylov space solver

Krylov-space iterative solvers are based on the idea of constructing more and more accurate solutions from the complex linear space defined by a Krylov sub-space. A Krylov space of dimension n is defined by

$$\mathcal{K}_n(A, \eta) \equiv \mathrm{span}\big\{\eta, D\eta, \ldots, D^{n-1}\eta\big\}\,. \tag{2A.7}$$

Many algorithms use the Krylov sub-space to generate the proper affine vector space where to search for the solution of the iterative problem.

For simplicity, we consider here the operator $Q^2 = D^\dagger D$, real symmetric and positive definite, since this is the form of the operator most often used in LQCD calculations. We assume here that in a finite volume v and at finite lattice spacing a, the lattice Dirac operator is a sparse large $N \times N$ matrix. We define the Q^2-norm as

$$||v||_{Q^2} = \sqrt{(v, Q^2 v)}\,. \tag{2A.8}$$

The CG algorithm is an iterative procedure that aims to find an approximate solution after n iterations, $\psi_n \in \mathcal{K}_n(D, \eta)$ minimizing the error

$$||e_n||_{Q^2} = ||S - \psi_n||_{Q^2} = ||DS - D\psi_n||_2\,, \tag{2A.9}$$

at step n. The rate of convergence of the CG depends on the spectrum of the lattice Dirac operator. If we make the assumption that all eigenvalues λ are well separated by the origin, and we only know the condition number $\kappa(D)$:

$$\kappa(D) = ||D|| \cdot ||D^{-1}|| = \sqrt{\frac{\lambda_{\mathrm{max}}(D^\dagger D)}{\lambda_{\mathrm{min}}(D^\dagger D)}}\,, \tag{2A.10}$$

then the Q^2-norm of the error after n iterations satisfies

$$\frac{||e_n||_{Q^2}}{||e_0||_{Q^2}} \leq 2\left(\frac{\sqrt{\kappa(D)} - 1}{\sqrt{\kappa(D)} + 1}\right)^n\,. \tag{2A.11}$$

This result guarantees that, barring too large values of $\kappa(D)$, the CG converges to a given accuracy ϵ with $\mathcal{O}(\sqrt{k})$ iterations.

THE CONJUGATE GRADIENT AND BICGSTAB ALGORITHMS

Since the analysis presented in this chapter focused on applications to the BiCGSTAB algorithm, we present the pseudo-code for both the simpler Conjugate Gradient algorithm and the BiCGSTAB.

2B.1   The Conjugate Gradient Method

Given a linear system $\mathbf{Ax} = \mathbf{b}$ and an initial guess vector $\mathbf{x_0}$ and a stopping parameter ϵ the algorithm proceeds as follows [25]:

Algorithm 1 Conjugate Gradient algorithm

$\mathbf{r}_0 \leftarrow \mathbf{b} - \mathbf{Ax}_0$
$k \leftarrow 0$
while $\mathbf{r}_k > \epsilon$ **do**
  $\alpha_k \leftarrow \frac{\mathbf{r}_k^\mathrm{T}\mathbf{r}_k}{\mathbf{p}_k^\mathrm{T}\mathbf{Ap}_k}$
  $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \alpha_k\mathbf{p}_k$
  $\mathbf{r}_{k+1} \leftarrow \mathbf{r}_k - \alpha_k\mathbf{Ap}_k$
  $\beta_k \leftarrow \frac{\mathbf{r}_{k+1}^\mathrm{T}\mathbf{r}_{k+1}}{\mathbf{r}_k^\mathrm{T}\mathbf{r}_k}$
  $\mathbf{p}_{k+1} \leftarrow \mathbf{r}_{k+1} + \beta_k\mathbf{p}_k$
  $k \leftarrow k + 1$
end while

The CG algorithm only requires the direct application of the linear operator $\mathbf{A}$ and two temporary vectors $\mathbf{r}_k$ and $\mathbf{p}_k$. By analysis, once can deduct that $\mathbf{r}_k$ and $\mathbf{p}_k$ span the same Krylov subspace and form two orthogonal basis, one with respect to the inner product, the other with respect to the inner product induced by $\mathbf{A}$. The vectors $\mathbf{x}_k$ represent the projection of the real solution $\mathbf{x}$ onto the Krylov subspace.

2B.2   The BiCGSTAB Algorithm

The BiConjugate Gradient Stabilized method was first introduced by H. A. van der Vorst [5] in 1992 and, as its predecessor BiCG, relies on two vectors, instead of one in the CG, two construct the Krylov subspace which result in smoother and faster convergence properties. The generalization

to the case of a complex linear operator, which is what is needed for the Dirac operator in LQCD, was first published by M. Gutknecht [35] and reads:

Algorithm 2 BiConjugate Gradient Stabilized algorithm

$\mathbf{r}_0 \leftarrow \mathbf{b} - \mathbf{A}\mathbf{x}_0$
$\mathbf{s}_0 \leftarrow \mathbf{r}_0$
$\mathbf{y}_0 \leftarrow \mathbf{r}_0$
$\delta_0 \leftarrow \mathbf{y}_0^\dagger \mathbf{r}_0$
$\phi_0 \leftarrow \frac{\mathbf{y}_0^\dagger \mathbf{A}\mathbf{s}_0}{\delta_0}$
$k \leftarrow 0$
while $\mathbf{r}_k > \epsilon$ **do**
    $\omega_k \leftarrow 1/\phi_k$
    $\mathbf{w}_{k+1} \leftarrow \mathbf{r}_k - \omega_k \mathbf{A}\mathbf{s}_k$
    $\chi_k \leftarrow \frac{(\mathbf{A}\mathbf{w}_{k+1})^\dagger \mathbf{w}_{k+1}}{||\mathbf{A}\mathbf{w}_{k+1}||^2}$
    $\mathbf{r}_{k+1} \leftarrow \mathbf{w}_{k+1} - \chi_k \mathbf{A}\mathbf{w}_{k+1}$
    $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \omega_k \mathbf{s}_k + \chi_k \mathbf{w}_{k+1}$
    $\delta_{k+1} \leftarrow \mathbf{y}_0^\dagger \mathbf{r}_{k+1}$
    $\psi_{k+1} \leftarrow -\frac{\omega_k \delta_{k+1}}{\chi_k \delta_k}$
    $\mathbf{s}_{k+1} \leftarrow \mathbf{r}_{k+1} - \psi_{k+1}(\mathbf{s}_k - \chi_k \mathbf{A}\mathbf{s}_k)$
    $\phi_{k+1} \leftarrow \frac{\mathbf{y}_0^\dagger \mathbf{A}\mathbf{s}_{k+1}}{\delta_{k+1}}$
    $k \leftarrow k + 1$
end while

In this algorithm, there are three auxiliary vectors needed for the iterations ($\mathbf{w}_k$, $\mathbf{r}_k$ and $\mathbf{s}_k$) and the stabilizing vector $\mathbf{y}_0$. Also, this algorithm only requires the direct application of the linear operator.

APPENDIX 2C

NEURAL NETWORK PARAMETERS IN DETAIL

When dealing with ML problems, and NN in particular, there are a lot of non-physical parameters and choices involved in the training. We briefly mentioned some of them in Section 2.2; here we present the choices for the values of hyper-parameters for reproducibility. We also mention that all models have been trained using the `scikit-learn` [36] package from scipy [37], in python which is the de facto standard language for ML, using the Multi-Layer Perceptron Regressor method.

Starting Values   When training a NN one can choose a starting point for the optimization algorithm. In practice, this is equivalent to setting starting values for all the weights and biases of the network. We have chosen to set them with a random uniform distribution of unit variance, following [38]. This is the most common choice and the default behavior in `scikit-learn`.

Over-fitting and Cross Validation   In order to prevent over-fitting, a standard L2-norm regulator has been used. This has the effect of penalizing networks with too large weights or biases, which can result from over-fitting, by adding a term to the loss function that is the L2-norm of all the weights and biases.

Cross-validation is a common technique in ML training to lessen the bias of the model with respect to the training data sample. It acts by reserving a subsample of the training dataset on which the model is not fit, but it is instead tested during training. This is very effective a reducing over-fitting, especially when the sample dataset size is small. The cross-validation sample was set to 10% of the training data.

Choice of the Optimizer   There are several possible optimizers for training a ML model that are readily available to use in most frameworks. The main difference between them is the speed and the ability to distinguish local minima from the global minimum of the parameter space. The optimizer we used in this work is the Stochastic Gradient Descent algorithm (SDG).

Choice of the Activation Function   The choice of the activation function, as for the optimizer, can be easily made in most ML frameworks like `scikit-learn` as they are readily available. The sigmoid is a very well-studied activation function, but it is rather expensive to compute as it requires exponentiation. A simpler function is the Rectified Linear Unit (ReLU), see Equation (2.6), which requires just an if statement, so it is much cheaper than the sigmoid. We chose the ReLU both for performance and because it more easily captures the chance of two nodes being completely uncorrelated, i.e. the weight between them is 0, while the sigmoid always has a non-zero contribution.

Input Scaling   To prevent numerical issues in the SDG minimizer, all the data that is used in the NN should have roughly the same magnitude and variance. We scaled all the data at every source-sink separation time to have zero mean and unit variance. This is a rather standard procedure and guarantees that all weights, biases and input data have the same magnitude. This is especially important in our case as the correlator data is exponentially decreasing with increasing t, meaning that the input to the NN would be extremely small compared to the biases and weights, which would make training almost impossible.

Number and Sizes of Hidden Layers   One of the most important choices to make when dealing with NN models is the structure of the NN itself. In order to fix the appropriate size of the layers of the network, we performed a semi-quantitative analysis of the results for varying sizes of the layers. We fixed the maximum number of layers to 3 and the sizes of the layers in the range from 1000 to 10. We then started from the largest network and used the SPNN3t2 model for the single point network and the GNN3 for the global one (they have the largest input layer size, all the models in the two classes have the same output layer size). By decreasing the sizes of the layers gradually, we found the point where results started to diverge significantly, meaning that the network became unable to represent the relation between the data properly. The final layer sizes were fixed to be at least 1.5 times larger than the found point. Note that the sizes of the layers were allowed to be different between the same network, in particular, the last hidden layer was

kept smaller than the first one or two.

The final sizes of the hidden layers of the NN that we used are $(150, 150, 100)$ for the single point models and $(600, 600, 200)$ for the global network models. It should be noted that during training, some of the NN did not converge, i.e. their loss function was never smaller than the predefined criterion, which we took as default, of 10^{-4}. This happened only for values of $t \approx T/2$, where the signal is weaker. However, we did not see any impact on the final results because of the bootstrap procedure we used.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] B. Joó *et al.*, "Status and future perspectives for lattice gauge theory calculations to the exascale and beyond," *The European Physical Journal A*, vol. 55, no. 11, p. 199, Nov. 14, 2019, ISSN: 1434-601X. DOI: 10.1140/epja/i2019-12919-7.

[2] R. Brower, N. Christ, C. DeTar, R. Edwards, and P. Mackenzie, "Lattice QCD application development within the US DOE exascale computing project," *EPJ Web of Conferences*, vol. 175, M. Della Morte, P. Fritzsch, E. Gámiz Sánchez, and C. Pena Ruano, Eds., p. 09 010, 2018, ISSN: 2100-014X. DOI: 10.1051/epjconf/201817509010.

[3] C. Drischler *et al.*, "Towards grounding nuclear physics in QCD," *Progress in Particle and Nuclear Physics*, vol. 121, p. 103 888, Nov. 1, 2021, ISSN: 0146-6410. DOI: 10.1016/j.ppnp.2021.103888.

[4] S. Erotokritou. "PRACE 24th call for proposals supports high-gain scientific research and industrial innovations," PRACE. (May 3, 2022), [Online]. Available: https://prace-ri.eu/prace-24th-call-for-proposals-supports-high-gain-scientific-research-and-industrial-innovations/.

[5] H. A. van der Vorst, "Bi-CGSTAB: A fast and smoothly converging variant of bi-CG for the solution of nonsymmetric linear systems," *SIAM Journal on Scientific and Statistical Computing*, vol. 13, no. 2, pp. 631–644, Mar. 1992, ISSN: 0196-5204, 2168-3417. DOI: 10.1137/0913035.

[6] J. Brannick, R. C. Brower, M. A. Clark, J. C. Osborn, and C. Rebbi, "Adaptive multigrid algorithm for lattice QCD," *Physical Review Letters*, vol. 100, no. 4, p. 041 601, Jan. 28, 2008, ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.100.041601.

[7] M. Lüscher, "Solution of the dirac equation in lattice QCD using a domain decomposition method," *Computer Physics Communications*, vol. 156, no. 3, pp. 209–220, Jan. 15, 2004, ISSN: 0010-4655. DOI: 10.1016/S0010-4655(03)00486-7.

[8] M. Lüscher, "Local coherence and deflation of the low quark modes in lattice QCD," *Journal of High Energy Physics*, vol. 2007, no. 7, pp. 081–081, Jul. 30, 2007, ISSN: 1029-8479. DOI: 10.1088/1126-6708/2007/07/081.

[9] J. M. Tang, R. Nabben, C. Vuik, and Y. A. Erlangga, "Comparison of two-level preconditioners derived from deflation, domain decomposition and multigrid methods," *Journal of Scientific Computing*, vol. 39, no. 3, pp. 340–370, Jun. 1, 2009, ISSN: 1573-7691. DOI: 10.1007/s10915-009-9272-6.

[10] E. Romero, A. Stathopoulos, and K. Orginos, "Multigrid deflation for lattice QCD," *Journal*

of Computational Physics, vol. 409, p. 109 356, May 15, 2020, ISSN: 0021-9991. DOI: 10.1016/j.jcp.2020.109356.

[11] R. Babich *et al.*, "Adaptive multigrid algorithm for the lattice wilson-dirac operator," *Physical Review Letters*, vol. 105, no. 20, p. 201 602, Nov. 11, 2010. DOI: 10.1103/PhysRevLett.105.201602.

[12] A. Frommer, K. Kahl, S. Krieg, B. Leder, and M. Rottmann, "Adaptive aggregation-based domain decomposition multigrid for the lattice wilson–dirac operator," *SIAM Journal on Scientific Computing*, vol. 36, no. 4, A1581–A1608, Jan. 2014, ISSN: 1064-8275, 1095-7197. DOI: 10.1137/130919507.

[13] B. Yoon, "A machine learning approach for efficient multi-dimensional integration," *Scientific Reports*, vol. 11, no. 1, p. 18 965, Sep. 23, 2021, ISSN: 2045-2322. DOI: 10.1038/s41598-021-98392-z.

[14] T. Lechien and D. Dudal, "Neural network approach to reconstructing spectral functions and complex poles of confined particles," 2022, Publisher: arXiv Version Number: 1. DOI: 10.48550/ARXIV.2203.03293.

[15] R. Abbott *et al.*, "Sampling QCD field configurations with gauge-equivariant flow models," 2022, Publisher: arXiv Version Number: 2. DOI: 10.48550/ARXIV.2208.03832.

[16] S. Foreman, T. Izubuchi, L. Jin, X.-Y. Jin, J. C. Osborn, and A. Tomiya, "HMC with normalizing flows," 2021, Publisher: arXiv Version Number: 1. DOI: 10.48550/ARXIV.2112.01586.

[17] G. S. Bali, S. Collins, and A. Schäfer, "Effective noise reduction techniques for disconnected loops in lattice QCD," *Computer Physics Communications*, vol. 181, no. 9, pp. 1570–1583, Sep. 1, 2010, ISSN: 0010-4655. DOI: 10.1016/j.cpc.2010.05.008.

[18] Y. Iwasaki, "Renormalization group analysis of lattice theories and improved lattice action: Two-dimensional non-linear o(n) sigma model," *Nuclear Physics B*, vol. 258, pp. 141–156, Jan. 1, 1985, ISSN: 0550-3213. DOI: 10.1016/0550-3213(85)90606-6.

[19] B. Sheikholeslami and R. Wohlert, "Improved continuum limit lattice action for QCD with wilson fermions," *Nuclear Physics B*, vol. 259, no. 4, pp. 572–596, Sep. 30, 1985, ISSN: 0550-3213. DOI: 10.1016/0550-3213(85)90002-1.

[20] M. Lüscher, S. Sint, R. Sommer, P. Weisz, and U. Wolff, "Non-perturbative o(a) improvement of lattice QCD," *Nuclear Physics B*, vol. 491, no. 1, pp. 323–343, Apr. 28, 1997, ISSN: 0550-3213. DOI: 10.1016/S0550-3213(97)00080-1.

[21] R. Horsley *et al.*, "The electric dipole moment of the nucleon from simulations at imaginary vacuum angle theta," 2008, Publisher: arXiv Version Number: 2. DOI: 10.48550/ARXIV.0808.1428.

[22] S. Aoki *et al.*, "Non-perturbative renormalization of quark mass in nf= 2 + 1 QCD with the schrödinger functional scheme," *Journal of High Energy Physics*, vol. 2010, no. 8, p. 101, Aug. 23, 2010, ISSN: 1029-8479. DOI: 10.1007/JHEP08(2010)101.

[23] J. Dragos, T. Luu, A. Shindler, J. de Vries, and A. Yousif, "Confirming the existence of the strong CP problem in lattice QCD with the gradient flow," *Physical Review C*, vol. 103, no. 1, p. 015 202, Jan. 19, 2021. DOI: 10.1103/PhysRevC.103.015202.

[24] P. H. Ginsparg and K. G. Wilson, "A remnant of chiral symmetry on the lattice," *Physical Review D*, vol. 25, no. 10, pp. 2649–2657, May 15, 1982. DOI: 10.1103/PhysRevD.25.2649.

[25] Y. Saad, *Iterative Methods for Sparse Linear Systems*, Second. Society for Industrial and Applied Mathematics, Jan. 2003, ISBN: 978-0-89871-534-7. DOI: 10.1137/1.9780898718003.

[26] T. Hastie, J. Friedman, and R. Tibshirani, *The Elements of Statistical Learning* (Springer Series in Statistics). New York, NY: Springer, 2001, ISBN: 978-0-387-21606-5. DOI: 10.1007/978-0-387-21606-5.

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, Publisher: arXiv Version Number: 9. DOI: 10.48550/ARXIV.1412.6980.

[28] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization.," *Journal of machine learning research*, vol. 12, no. 7, 2011.

[29] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems*, vol. 2, no. 4, pp. 303–314, Dec. 1989, ISSN: 0932-4194, 1435-568X. DOI: 10.1007/BF02551274.

[30] V. Maiorov and A. Pinkus, "Lower bounds for approximation by MLP neural networks," *Neurocomputing*, vol. 25, no. 1, pp. 81–91, Apr. 1999, ISSN: 09252312. DOI: 10.1016/S0925-2312(98)00111-8.

[31] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986, ISSN: 0028-0836, 1476-4687. DOI: 10.1038/323533a0.

[32] S. Linnainmaa, "Analysis of some known methods of improving the accuracy of floating-point sums," *BIT Numerical Mathematics*, vol. 14, no. 2, pp. 167–202, Jun. 1, 1974, ISSN: 1572-9125. DOI: 10.1007/BF01932946.

[33] B. Yoon, T. Bhattacharya, and R. Gupta, "Machine learning estimators for lattice QCD observables," *Physical Review D*, vol. 100, no. 1, p. 014 504, Jul. 9, 2019, ISSN: 2470-0010, 2470-0029. DOI: 10.1103/PhysRevD.100.014504.

[34] B. Efron, "Bootstrap methods: Another look at the jackknife," *The Annals of Statistics*, vol. 7,

no. 1, Jan. 1, 1979, ISSN: 0090-5364. DOI: 10.1214/aos/1176344552.

[35]   M. H. Gutknecht, "Variants of BICGSTAB for matrices with complex spectrum," *SIAM Journal on Scientific Computing*, vol. 14, no. 5, pp. 1020–1033, Sep. 1993, ISSN: 1064-8275, 1095-7197. DOI: 10.1137/0914062.

[36]   F. Pedregosa *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[37]   P. Virtanen *et al.*, "SciPy 1.0: Fundamental algorithms for scientific computing in python," *Nature Methods*, vol. 17, no. 3, pp. 261–272, Mar. 2, 2020, ISSN: 1548-7091, 1548-7105. DOI: 10.1038/s41592-019-0686-2.

[38]   X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *Journal of Machine Learning Research - Proceedings Track*, vol. 9, pp. 249–256, Jan. 2010.

CHAPTER 3

FIRST RESULTS OF HADRON SPECTROSCOPY WITH STABILIZED WILSON FERMIONS

Modern Lattice QCD calculations require ever-improving procedures and techniques for the generation of gauge field ensembles. In this chapter, we present a new effort in this direction started by the OPEN LATtice Initiative.

As was briefly discussed in Section 1.4, Wilson-Clover fermions (WCF) [1] are a very popular choice in LQCD among other discretizations because of their straightforward interpretation and the very limited restrictions they pose on possible observables. This also leads to them being among the simplest action to implement numerically and analyze. The discretization of regular Wilson fermions is of order $\mathcal{O}(a)$, but this can be rigorously improved to higher order, as, for example, through the clover term if the coefficient c_{SW} is determined non-perturbatively. Thus WCF over the years enabled several precision calculations for a variety of physical observables, for example, several quantities that are used in the biyearly Flavor Lattice Averaging Group (FLAG) report [2], which summarizes the most important results from LQCD.

There are, however, some limitations on the use of Wilson-Clover fermions. One of the most striking is the breaking of chiral symmetry, which in the context of gauge field generation has the impact of allowing the Dirac operator to take arbitrarily small values, especially in the case of small quark masses and coarse lattice spacing, an effect that grows worse with the larger lattice volumes. This, in turn, means that to generate ensembles at low pion masses, one needs to have very fine lattice spacings, but as we discussed in Section 1.4, small lattice spacings lead to topological freezing and critical slowing down in the MCMC. This effectively poses some bounds on the parameter space, in terms of lattice spacing, volume and quark mass, where simulations can be performed successfully. One should, however, note that these limits are however very similar for other lattice discretizations.

In recent years there has been a growing interest in master-field simulations [3], which by

definition require very large volume simulations. This leads to an effort to improve the volume limitations of Wilson-Clover fermions by introducing a set of techniques for numerical stabilization and a new definition of the fermion action [4]. This "package" of both algorithmic and analytical improvements is what we call *Stabilized Wilson Fermions* (SWF).

The SWF package is of interest on its own, not just in the context of master-field simulations. Even at traditional volumes, SWF could have great benefits for single-baryon and light nuclear physics applications due to them enabling coarse lattice spacings at quark masses closer to the physical point, where WCF have issues due to stability [5]. For this reason, a new research collaboration has been formed to study and generate ensembles with SWF at the state-of-the-art level.

3.1   The OPEN LATtice initiative

The study of Stabilized Wilson fermions only began in 2018 and many questions with regards to their applicability and properties in the parameters space need to be answered. In the context of master-field simulations there are encouraging results from [6, 7], but more evidence is still needed, especially in the case of coarse lattice spacing. The very first study [4], in which the non-perturbative tuning of the Sheikholeslami-Wohlert coefficient c_{SW} was begun, showed early indication of a very good scaling behavior towards the continuum.

To enable further investigations on SWF the OPEN LATtice initiative was founded [8]. This is, at the time of this writing, composed by seven principal investigators from the USA, Europe and Asia and one doctoral researcher, the author of this thesis. The goal of the collaboration is to generate state-of-the-art QCD ensembles using SWF and share them according to the principles of the open science philosophy with all of the LQCD community. Most of the results presented in this chapter have been produced as part of this effort.

3.1.1 Open science policy

Since generating QCD gauge field ensembles is an extremely costly procedure from the numerical point of view, it became apparent in the early stages of Lattice QCD that storing and sharing the ensembles was of particular importance. One key aspect is to enable research groups with limited access to HPC facilities to participate in the scientific effort without having to start from scratch, but by using community-generated ensembles.

Some remarkable community-wide efforts were made, for example in the establishment in 2002 of the International Lattice Data Grid (ILDG) [9, 10] and all the previous smaller regional Lattice Data Grid. This proposed a standard format for the metadata and the gauge field configuration data and provided an interface where collaboration could upload and share ensembles. However, over the years, its use and relevance decreased. Only in the past two years, an effort to revise, improve and revive the ILDG has been set in progress.

The OPENLat collaboration strongly believes in the values of open science as a means to facilitate the advance of Lattice QCD. Here are summarized the main points that we stand by:

- *Define and uphold quality:* we propose to define quality standards for the ensembles we generate. In particular, some control observables are routinely checked to assess the quality of the produced gauge configurations and the stability of the Markov Chain. These quality standards are themselves continuously researched and improved upon.

- *Share and maintain repository:* a crucial aspect when generating gauge field ensembles is to maintain an accessible long-term storage space for all the data that is produced. This requires data-management facilities in HPC centers, so it is also important to provide strategies for easy access to the data once it is generated and to control the integrity of the data itself. Once the ILDG should become active again, we are strongly in favor of complying with the upcoming standard of formats and sharing procedures.

- *Community boosting:* the collaboration is open to the possibility of new members of interested parties joining our efforts with new resources to expand our sets of ensembles.

- *Grant and enable access:* we will ensure that when the ensembles are completed, which is a three-stage process (see the following section), all the gauge field configurations will be made open access. This means that no restriction will be imposed by the collaboration on which research groups/projects can use the ensembles.

3.1.2 SWF Ensembles from the OPENLAT Initiative

The goal of the OPENLAT initiative is to generate a set of lattice gauge field ensembles using SWF that span a multitude of parameters. This is done obviously to allow taking all the limits required to handle the systematic uncertainties properly for rigorous precision results. In particular, we span three parameters: the lattice spacing a for continuum extrapolation, the pion mass m_π for chiral extrapolations[1] and the total lattice size L to control the finite volume effects. It should be noted that the coarsest lattice spacing, especially with small pion masses, which are, for example, useful for nuclear physics studies, is of particular interest since they represent the regime where SWF are supposedly better than regular WCF.

The target is to generate 500 independent configurations per ensemble, to later be extended to 1000. This is to be done in three stages:

- Stage I: ensembles at the $\mathrm{SU}(3)$ flavor symmetric point, i.e. $m_\pi = m_K = 412$ MeV

- Stage II: ensembles with $m_\pi = 300, 200$ MeV

- Stage III: ensembles at the physical point $m_\pi = m_K = 135$ MeV

At the end of each stage, a publication presenting basic observables computed on the ensembles will be released as reference, together with the public availability of all the gauge field configurations.

The choices of lattice spacings to be simulated are $a = 0.064, 0.077, 0.094$ and 0.12 fm, and all the ensembles have standard periodic boundary conditions. However, an additional set of

---

[1]Chiral extrapolations are extremely useful even when ensembles at the physical point are available, as they can help constrain fits in cases where a functional form for the m_π dependence is known, for example from chiral perturbation theory (χPT) to improve the signal at the physical pion mass by means of statistical fitting.
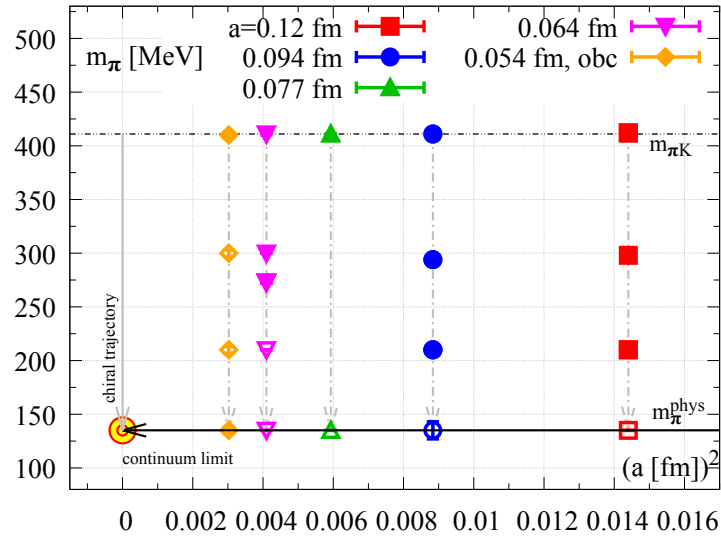
Figure 3.1: Planned combinations of pion mass m_π and lattice spacing a for the OPENLAT ensembles.
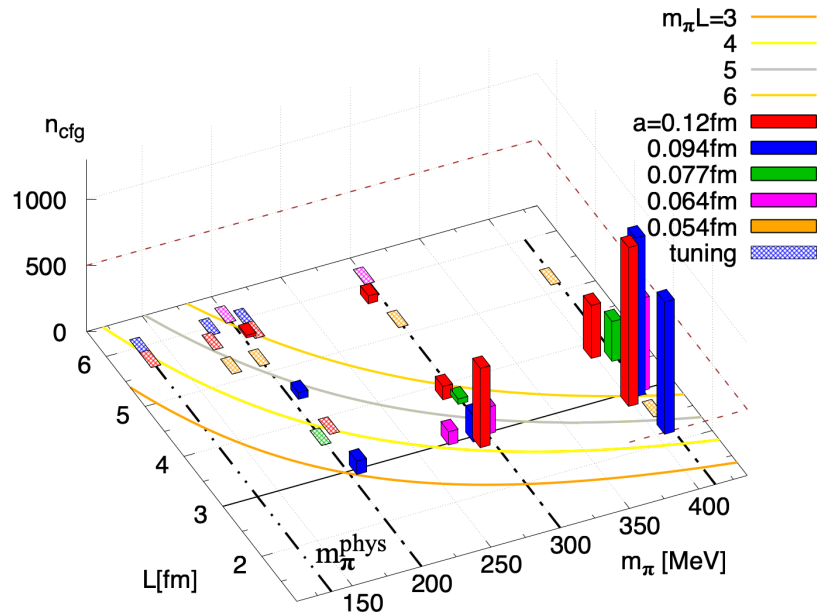
Figure 3.2: Overview of the OPENLAT ensembles presented here as parametrized by the lattice spacing a and $m_\pi L$. The height of the bars in the histograms represents the number of independent configurations that have been generated. Shaded boxes are ensembles in the tuning state.

ensembles with $a = 0.055$ fm will be generated with open boundary conditions to further improve critical slowing down and circumvent topological freezing.

In Figure 3.1 the selected combinations of m_π and a are presented. At the time of this writing, the OPENLAT Initiative is approaching the completion of Stage I, in Figure 3.2, there is the most recent available status update on the ensemble generation. The shaded squares represent ensembles that are in the "tuning" stage, where the MC algorithm for generation is adjusted and the Markov chain is thermalized. In this chapter, we will present some of the procedures used for computing the reference observables for the upcoming Stage I publication.

3.2 Stabilized Wilson Fermion Package

The SWF package comprises of a set of tools aimed at improving the stability of Wilson fermions simulations. While some of them had already been established previously, others are entirely new and were proposed in [11]. We can, however, categorize the improvements into two sets; one is aimed at better stability of the generation algorithms and other is improving the fermion discretization. The simulations are performed using the openQCD2.0+ software package[12]. As stated in the previous section, to generate the state-of-the-art ensembles a plethora of well-established techniques that improve the performance of the MCMC is also used, such as mass-preconditioning through Hasenbusch chains [13, 14], twisted mass reweighting for light quarks [15, 16] and the improved linear solvers such as the SAP [17, 18].

3.2.1 Algorithmic Stability Improvements

There are three components to the SWF in this category. The first is the Stochastic Molecular Dynamics (SMD) algorithm [19–22]. It is a modification of the traditional HMC algorithm aimed at preventing the problem arising from large jumps in the phase space trajectory. These can, for example, be due to an accumulation of errors coming from the numerical integration of the MD trajectory. The issue is that when such a jump occurs, the algorithm needs to spend several cycles to re-thermalize towards the target distribution. The key difference in SMD is that the momenta

are not chosen randomly at every step but are instead randomly rotated from one update to the other.

The algorithm is schematically presented below:

---

1. Refresh $\pi(x, \mu)$ and $\phi(x)$ by a random field rotation: $\quad \pi \to c_1 \pi + c_2 v$

$$\phi \to c_1 \phi + c_2 D^\dagger \eta$$

   (v and η normal distributed)

   $c_1^2 + c_2^2 = 1, c_1 = e^{-\epsilon\gamma}, \quad \epsilon = \text{MD integration time}, \quad \gamma = \text{friction parameter}$

2. short MD evolution

3. Accept/Reject-step $\hspace{6cm}$ (exact algorithm)

4. Repeat $\circlearrowleft$

Where $\pi(x, \mu)$, $\phi(x)$ are the momenta of the pseudo-fermion and gauge field, respectively. With a little work, it can be shown that the SMD and the HMC are equivalent in the case of fixed ϵ and large γ. In Figure 3.3, we show a simple schematic representation of the SMD as compared to the HMC algorithm.

Some interesting features of the SMD are the proven ergodicity and its convergence to the exact stationary solution [23], the significant reduction of unbounded energy violations $|\Delta H| \gg 1$ and the decrease autocorrelation at fixed cost. The effect is due to the algorithm being a "slower" than the HMC in terms of updates, but sampling the phase space in a more efficient way that more than compensates for the difference. Here ΔH is the difference in the energy before and after a full microcanonical update, which enters the accept-reject step of the algorithm in the exact manner as in the HMC, see Equation (1.44). One final feature, which only appears when using the SMD in conjunction with deflation algorithms for the pseudo-fermions, is that the smoother updates in the gauge field and its associated momenta $\phi(x)$ lead to improved updates of the deflation subspace as well.
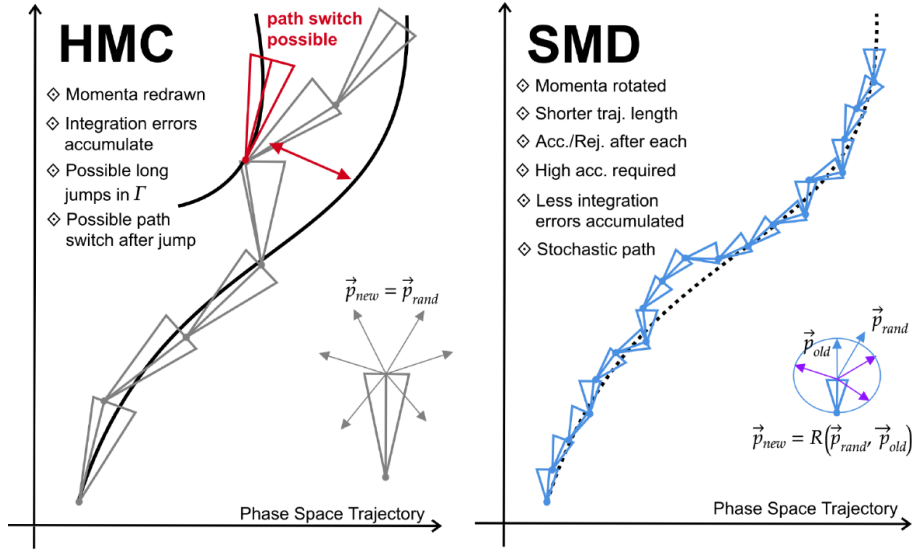
Figure 3.3: Pictorial representation of the HMC and SMD algorithms for generation of gauge field ensembles. Image courtesy of A. Francis.

The next tool for improving the algorithmic stability in SWF that we present is the use of a volume-independent norm as a stopping criterion in the linear system solve for pseudo-fermions. As shown in Section 1.4, it is usual practice to stop the algorithm when:

$$\|r_n\| = \|\eta - Dq_i\| < \epsilon \|\eta\|. \tag{3.1}$$

However, this equation hides the fact that one is free to choose any vector norm. Most commonly, the L_2 norm is chosen, but this leads to the criterion to be volume dependent. Hence, in the SWF package, the supremum norm is used. This norm helps against local effects that could lead to loss of precision.

L_2 norm	$\|\eta\|_2 = \left(\sum_x (\eta(x), \eta(x)) \right)^{1/2}$	$\rightarrow$	$\propto \sqrt{V}$
supremum norm:	$\|\eta\|_\infty = \sup_x \|\eta\|_2$	$\rightarrow$	$V-\text{independent}$

The last algorithmic stability improvement is the modification of the accept/reject global sum implementation, in `openQCD2.0` in our case, to handle quadruple precision to prevent the accumulation of error, especially in the case of very large volumes. This is relevant primarily for master-field simulations, but it is nonetheless a part of the SWF package.

3.2.2 The Exponential Clover Action

Probably the most striking feature of SWF is the use of a new discretization for the fermion action. This makes SWF fundamentally different from WCF, even though, as we will see, the new discretization is very similar to the original one by Sheikholeslami and Wohlert [1]. To start, we recall, from the discussion in Section 1.2, the full form of the WCF operator:

$$D^{\mathrm{SW}}(x, y) = \frac{1}{2a} \sum_{\mu=\pm 1}^{\pm 4} (\mathbb{1} - \gamma_\mu) U_\mu(x) \delta_{x,(y-\hat{\mu})} + \left(m + \frac{4}{a} \right) \delta_{x,y} + \frac{i}{4} c_{\mathrm{SW}} \sigma_{\mu\nu} G_{\mu\nu}(x) \delta_{x,y}. \qquad (3.2)$$

When applying even-odd preconditioning, as is standard practice since the field at a lattice site is only connected through the derivative to its nearest neighbors, which all have the opposite "parity", the operator is takes the blocked form:

$$D = \begin{pmatrix} D_{ee} & D_{eo} \\ D_{oe} & D_{oo} \end{pmatrix}. \qquad (3.3)$$

The hopping term is only in the even-odd off-diagonal component, while the diagonal blocks are diagonal matrices themselves. Their sum is the diagonal component of the original D^{SW} operator:

$$D_{ee} + D_{oo} = M + c_{\mathrm{SW}} \frac{i}{4} \sigma_{\mu\nu} G_{\mu\nu}, \qquad (3.4)$$

where we have used the shorthand notation of $M = \frac{4}{a} + m$. By inspecting this diagonal term, one can conclude that the SW Dirac operator is not protected from having arbitrarily small eigenvalues coming from the sum of M with the clover term. Furthermore, in this formulation, the positive and negative eigenvalues are distributed equally. The probability of D_{oo} having a very small eigenvalue scales with the number of eigenvalues itself, which is determined by the volume, making such simulations pathological.

In order to fix this problem, when Stabilized Wilson Fermions were proposed in [11], the WCF discretization was replaced with an exponentiated version of the clover term:

$$D_{ee} + D_{oo} = M + c_{\mathrm{SW}} \frac{i}{4} \sigma_{\mu\nu} \hat{F}_{\mu\nu} \quad \rightarrow \quad M \exp\left[\frac{c'_{\mathrm{SW}}}{M} \frac{i}{4} \sigma_{\mu\nu} \hat{F}_{\mu\nu} \right]. \qquad (3.5)$$

It is trivial to see that the first-order expansion of the exponential version of the operator is identical to the traditional Wilson Clover. This new variant is, by definition, positive definite and

easily invertible. These two features make it suitable for even-odd preconditioning, as now D_{oo} is fully protected from small eigenvalues. It should also be noted that the exponential clover term is a valid Symanzik improvement term, which is crucial for renormalizability. One should take care that the c'_{SW} coefficient is different in the exponential version than the one in the ordinary clover action, so it needs to be re-tuned non-perturbatively.

It should, however, be noted that exceptionally small eigenvalues could still be present as in the traditional WCF, but these exceptional configurations would not be given by the clover term anymore, but from other sources. SWF do not entirely solve this problem, but they mitigate it in a parameter region that is inaccessible to regular WCF simulations.

3.2.3   A Quenched Example of the Exponential Clover Action

To show the difference between the WCF action and the Exponential Clover one, we report the test done in [24] of a quenched calculation of the pion correlator. By quenched, we mean that the fermion action that was used for generating the ensemble is not the same action that is used when calculating the quark propagator for the fermion contractions. In particular, this test was performed with 1000 configurations with the gauge coupling $\beta = 6.0$ ($a = 0.093$fm) and volume $L^4 = 48^4$. The coefficients c_{SW} for the two actions were tuned separately and the values of the quark masses were fixed to give the same pion masses $m_\pi^{\mathrm{val}} = 320$ MeV.

The results, which can be seen in Figure 3.4, show no spikes in the correlator MC history, implying a much stronger stability for the exponential clover term. In turn, for all Euclidean separation times t (right plot), the correlator is well behaved, while it is clear that for times $t/a = 20$ the standard clover results are dominated by the noise coming from exceptional configurations.

3.3   Simulating Dynamical QCD with $2 + 1$ Quark Flavors and SWF

The goal of the OPENLAT Initiative is to generate QCD ensembles with $2 + 1$ dynamical fermions, that is, a light degenerate isospin doublet of light quarks and a heavier strange quark. In this section, we first briefly describe the tuning procedure for c_{SW} (Section 3.3.1), then explain how
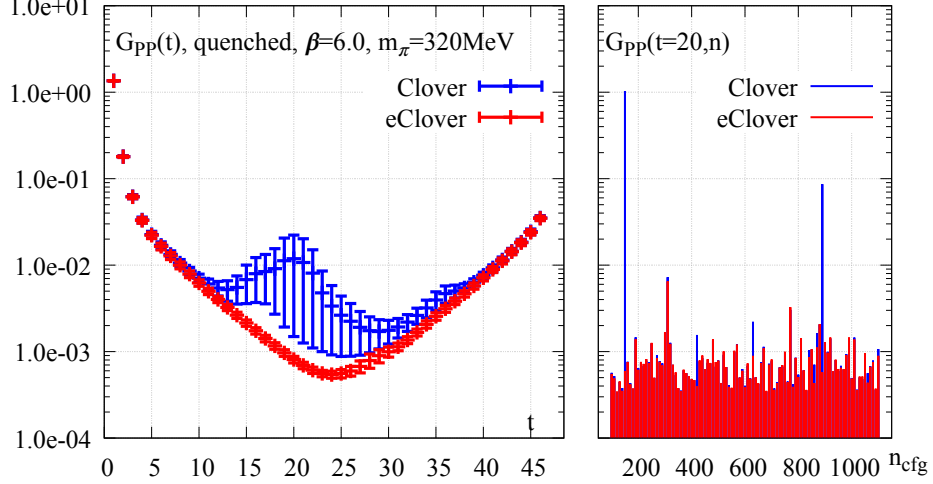
Figure 3.4: Left: The pion correlator determined using both the standard and exponentiated clover terms. Right: The Monte Carlo history of the pion correlator at fixed Euclidean time $t/a = 20$, towards the center of the lattice. Plots courtesy of A. Francis.

the quark masses are fixed (Section 3.3.2) and finally, we'll give some details on the ensemble generation process (Section 3.3.3).

3.3.1 Non-perturbative Tuning of c_{SW}

Following the procedure of Lüscher et al. [25] the tuning of c_{SW} was performed in [11] for the range $\beta \geq 3.8$ for three quark flavors. Later in [24] results were presented that extended to $\beta = 3.685$ to enable even coarser lattice spacings. This tuning is performed using the Schrödinger functional boundary conditions [26], determining c_{SW} by imposing $\mathcal{O}(a)$-improvement on small volumes, where $L = 8$ and $T = 16$. For the $\beta = 3.685$ point a value of $L = 16$ was used instead. In Figure 3.5 on the left, we report a comparison of the tuned values of c_{SW} for WCF and our new exponential clover action. On the right there is a comparison of the fits between [11] and [24].

The results of this procedure lead to the following interpolation for the Sheikholeslami-Wohlert coefficient for the exponential clover action:

$$c_{\mathrm{SW}}(g_0^2) = \frac{1 - 0.325022 g_0^2 - 0.0167274 g_0^4}{1 - 0.489157 g_0^2}. \tag{3.6}$$
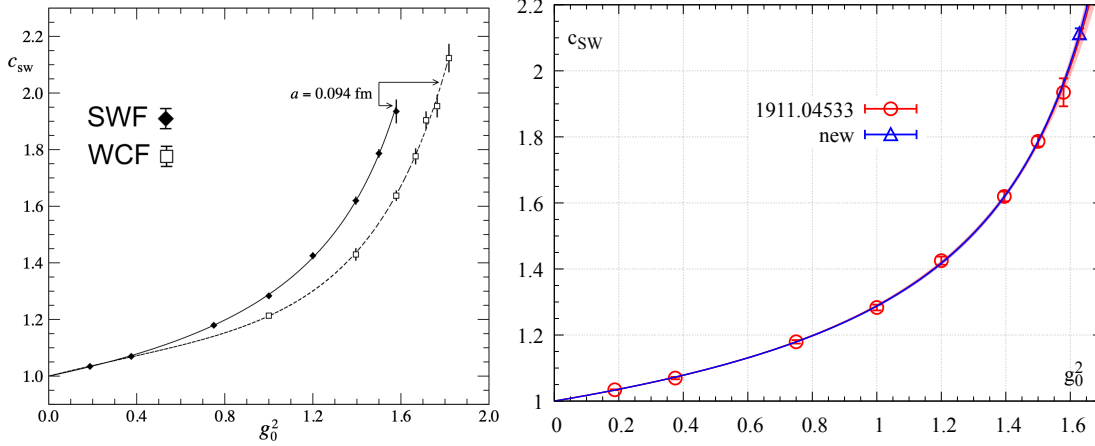
Figure 3.5: Non-perturbative tuning of c_{SW} using the Schrödinger functional for $2+1$ dynamical fermions. Left plot: comparison of the WCF and exponentiated clover fermions as presented in [11]. Right plot: more recent results for the tuning of c_{SW} (blue line and points) with an additional value of the coupling $\beta = 3.685$, as done in [24]. The red line and points are the results from [11].

3.3.2 Setting the chiral trajectory

In order to simplify the treatment of the continuum limit, the OPENLAT Initiative decided to adopt the approach previously used by the QCDSF [27] and CLS [28] collaborations for dealing with the tuning of the quark mass parameters. The key idea is to link the masses of the light quark and of the strange quark together by tuning them at the $\mathrm{SU}(3)$ flavor-symmetric point, where all quark masses are equal to each other and fix the trace of the quark matrix when changing the coupling: $\mathrm{Tr}[M] = \mathrm{const}$. This is done to reduce the mass-dependence cutoff effects.

The starting point of the procedure is the relation, coming from chiral perturbation theory, for the meson mass combination $m_{\pi K}$, which is known to be proportional to the trace of the quark mass matrix. This information can then be used as a tuning condition, that is, one changes the quark mass values until the physical value is obtained:

$$m_{\pi K}^2 = \frac{2}{3}\left(m_K^2 + m_\pi^2/2\right) \equiv m_{\pi K}^2|_{\mathrm{phys}}. \tag{3.7}$$

This mass tuning needs to be done at every value of the coupling, i.e. for every lattice spacing, with some minor perturbative effects coming from isospin breaking and electromagnetic effects, see [29]. When a different pion/kaon mass is desired, one needs to change the quark mass matrix

103

accordingly, keeping $\mathrm{Tr}[M]$ fixed. This means that as the pion mass approaches the physical point, so will the kaon mass.

3.3.3 Generation of SWF Ensembles

At this point, we have presented all the ingredients needed by the OPENLAT initiative to generate the SWF ensembles. As discussed in Section 1.4, in order to fix the scale of the simulation, one needs one physical input for every parameter of the simulation. We chose to parametrize everything in terms of the pion mass m_π, the kaon mass m_K and the gradient-flow time scale t_0 [30], following the procedures of the CLS collaboration [28, 31]. In particular, the two quantities ϕ_2 and ϕ_4 are kept fixed while approaching the continuum:

$$\phi_2 = 8t_0 m_\pi{}^2 \qquad \phi_4 = 8t_0 \left(m_K{}^2 + \frac{1}{2} m_\pi{}^2 \right). \tag{3.8}$$

This criterion, together with the non-perturbative tuning of c_{SW} and the fixing of the trace quark mass matrix, sets the conditions for our simulations.

At the time of this writing, in the table below we report the available ensembles from the OPENLAT Initiative at the flavor-symmetric point:

β	Size	a[fm]	m_π	L [fm]	$m_\pi L$	N_{cfg}
3.685	$24^3 \times 96$	0.12	411.54(46)	2.9	6.0	1200
3.8	$24^3 \times 96$	0.094	410.58(84)	2.3	4.5	1200
	$32^3 \times 96$	0.094	409.57(46)	3.0	6.0	1300
3.9	$48^3 \times 96$	0.077	410.39(22)	3.6	7.5	300
4.0	$48^3 \times 96$	0.064	408.93(25)	3.0	6.4	600

Table 3.1: Current Stage I ensembles of the OPENLAT Initiative.

Modern lattice simulations have complicated algorithms and a plethora of control observables need to be monitored while generating configurations via MCMC. Here we present a non-exhaustive list of the checks that our collaboration performs while generating ensembles, with selected results for the $\beta = 3.8$, $V = 32^3 \times 96$ ensemble shown as examples:

- Lack of thermalization effects in the plaquette and in the energy density. If there is a drift in the average value of one of these quantities, then the algorithm is not sampling the PDF properly. In particular, a jump in either observable would imply the transition from a local minimum of the action to another, which is undesirable.
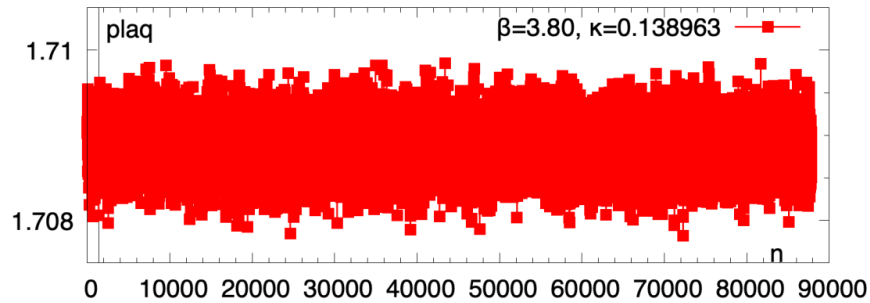
Figure 3.6: Monte Carlo history of the plaquette during the a094m400 ensemble Markov chain. As expected, since the system is thermalized properly, the value is constant along the chain within the bounds of small stochastic fluctuations.

- Check the fluctuations, distribution and autocorrelation of the topological charge Q computed via the gradient flow. This is crucial in order to ensure correct sampling from all topological sectors, see Appendix 1B. The autocorrelation of this observable will be analyzed more in detail in Section 3.4.1
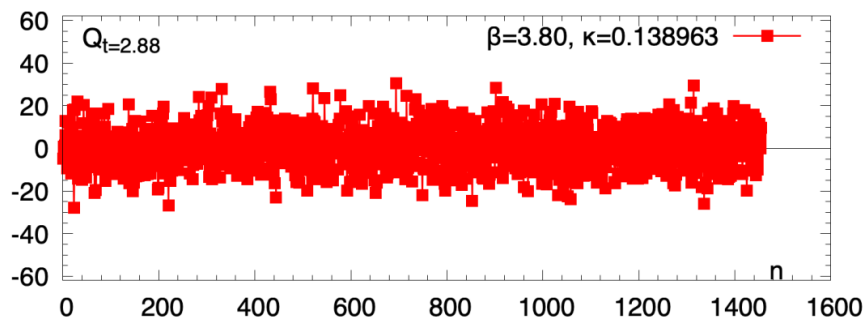
Figure 3.7: Monte Carlo history of the topological charge computed with the gradient flow during the a094m400 ensemble Markov chain.
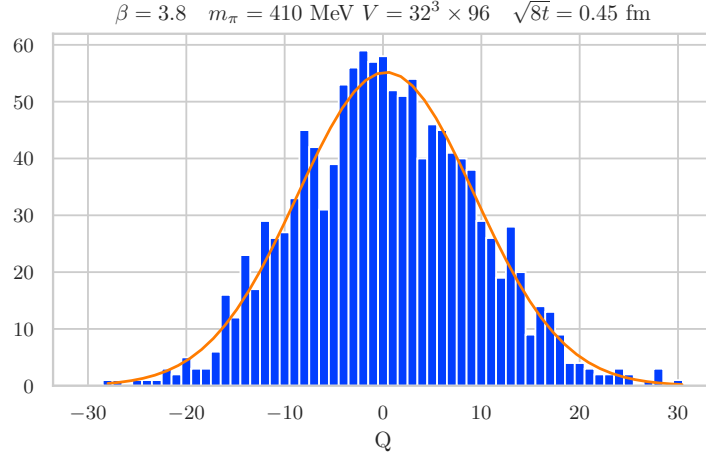
Figure 3.8: Distribution of the topological charge computed with the gradient flow during for the ensemble a094m400. The orange line is a Gaussian fit to the histogram. The data is well described by a Gaussian, meaning that the algorithm is sampling all topological sectors effectively.

- We ensure that the distribution of the error in the Hamiltonian at the end of a single MD update, ΔH, matches the acceptance rate of the SMD algorithm. A more detail discussion of this criterion can be found in [32], but, in general, if large fluctuations and non Gaussian behavior were to be found in the distribution of ΔH that would imply that the MD step size $\delta\tau$ is too large leading to suboptimal sampling.
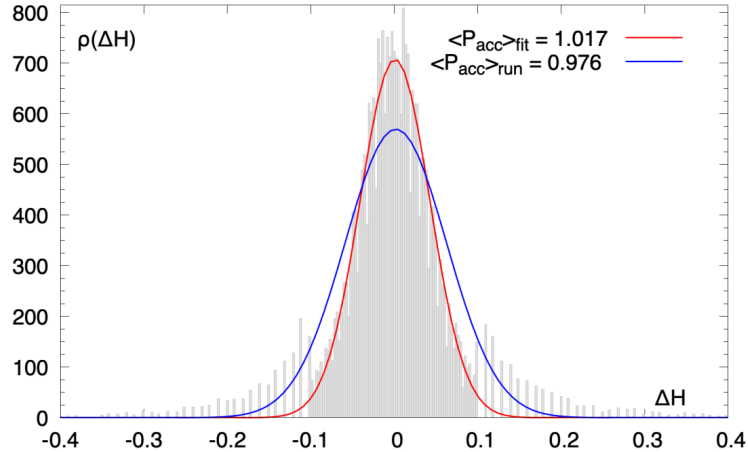
Figure 3.9: Distribution of the error in the Hamiltonian ΔH of the MD updates during the a094m400 ensemble Markov chain. The solid lines represent the expected distribution from the acceptance rate (blue) and the fitted distribution (red). The relative discrepancy is acceptable as long as the fitted Gaussian is narrower and no apparent deviations from a normal distribution are present.

- For the chiral trajectory, we make sure that the tuning quantity $\phi_4 = 8t_0(m_K^2 + m_\pi^2/2)$ is within 0.5% of the target value of 1.115 with an error of max. 1σ. This quantity is a hadronic observable, we will discuss it more in details in Section 3.4.3.

3.4 First SWF Ensembles and Light Hadron Spectroscopy

Now that we have presented how the SWF ensembles are generated, we turn our attention to their physical properties. As mentioned in the presentation of the OPENLAT Initiative, one of the aims of the collaboration is to provide potential users with all the relevant metadata. In particular, we want to also provide reference values for the autocorrelation of the configurations and light hadron masses. These quantities are not meaningful when computed on a single configuration; they are only useful as ensemble averages. Hence, it would be interesting to provide these results in the most unbiased and reproducible form as possible.

In this chapter we, will first present some results for pure gauge observables and the integrated autocorrelation times for the topological charge of the ensembles under consideration, then in Section 3.4.2 we will present a method for Bayesian estimation of the hadron spectrum and, finally, in Section 3.4.3 we will show the preliminary results.

This analysis is restricted to a subset of all the configurations shown in Table 3.1 as shown in Table 3.2; this is due to the ongoing generation of these ensembles. For simplicity, we assign a readable name to each ensemble, which we will use to refer to them in the rest of this section.

Ensemble	t_0/a^2	a [fm]	N_{cfg}	N_{srcs}
a12m400	1.48548(64)	0.12	600	100
a094m400S	2.4482(50)	0.094	500	100
a094m400	2.43979(89)	0.094	1200	100
a077m400	3.6198(30)	0.077	900	100
a064m400	5.2588(46)	0.064	1000	100

Table 3.2: SWF Ensembles used for the analysis. We note that the ensemble a094m400S is labeled differently as it represents a smaller volume compared to the others. N_{cfg} represents the number of gauge field configurations analyzed, but not all of them are to be considered independent. N_{srcs} is the number of stochastic sources used to compute the averaged hadron correlators.

3.4.1 Gradient Flow Scale Determination and Autocorrelation Analysis

The first analysis that we performed on the ensembles was to determine the value of the gradient flow scale t_0 to fix the lattice spacing in physical units. We used the value of $\sqrt{t_0} = 0.1464(18)$ fm from [28] to convert to fermis. The determination of t_0 has been carried out following the procedure outlined in Appendix 1C and the results are summarized in Figure 3.10 for the five ensembles at hand.

Scale Setting with the Gradient Flow

a12m400	$t_0/a^2 = 1.48548(64)$	
a094m400	$t_0/a^2 = 2.43979(89)$	
a094m400S	$t_0/a^2 = 2.4482(50)$	
a077m400	$t_0/a^2 = 3.6198(30)$	
a064m400	$t_0/a^2 = 5.2588(46)$	
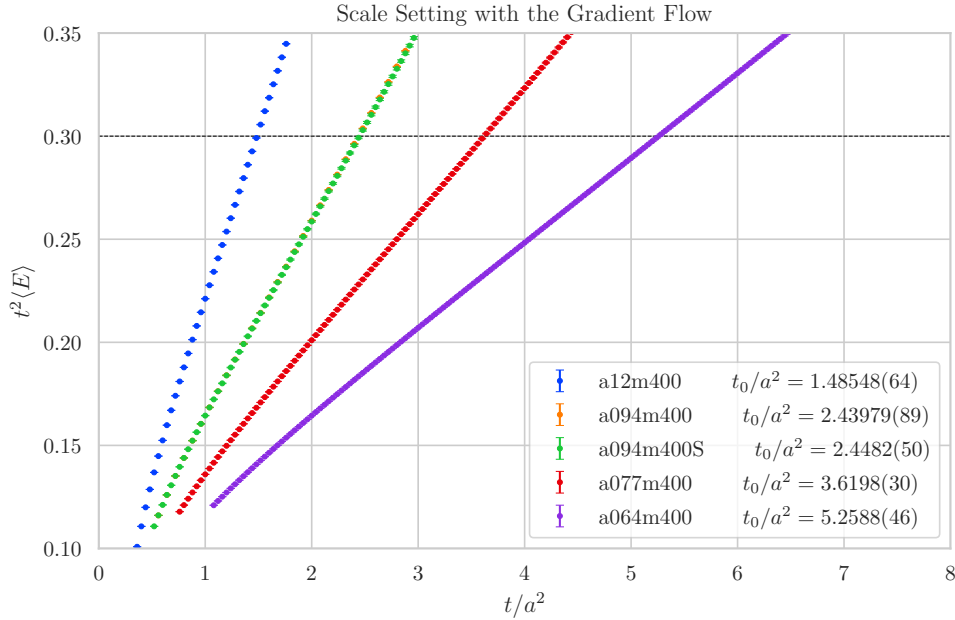
Figure 3.10: Scale setting example on the Stabilized Wilson Fermions ensembles, more details found in Chapter 3. In order to determine the lattice spacing a in physical units one determines the point t_{f}/a^2 at where $t_{\mathrm{f}}^2\langle E\rangle = 0.3$.

Another quantity of interest that can function as a check for the quality of the ensembles but that can only be determined at the ensemble level is the integrated autocorrelation time τ_{int} of some observables. In particular, it is known that the topological charge (see Appendix 1B for a detailed discussion) has the longest correlation length along the Markov chain [33]. Its value is crucial to understand if all topological sectors are being sampled correctly and also to determine if the configurations that are being produced are statistically independent. To do this, we employed the windowing procedure outlined in Appendix 1A.1 on data for the topological

charge Q at non-zero flow-time, with $\sqrt{8t_\mathrm{f}} \approx 0.3$ fm. The integrated autocorrelation time has also been computed for the hadronic two-point correlators for the pion, the proton and the Ω baryon, as these are the quantities that we focused on afterward. For the correlators, we chose a value for the source-sink separation $t = 20$ to compute the values for τ_π, τ_P,τ_Ω, this is an arbitrary choice, but there is no reason to think the autocorrelation, especially at large t should depend on it. The results are reported in Table 3.3.

Ensemble	τ_Q	τ_π	τ_p	τ_Ω
a12m400	0.53(4)	1.56(26)	0.53(4)	0.50(4)
a094m400S	0.66(13)	0.51(6)	0.54(6)	0.51(6)
a094m400	0.64(6)	0.55(6)	0.51(4)	0.44(6)
a077m400	0.54(10)	0.63(8)	0.72(10)	0.75(13)
a064m4001	2.25(83)	0.81(15)	1.02(21)	1.03(20)
a064m4002	1.52(42)	0.50(6)	0.57(11)	0.66(11)

Table 3.3: Integrated autocorrelation times of the topological charge Q and of the pion, proton and Ω two-point correlators on the SWF flavor-symmetric ensembles.

One important remark is that the ensemble a064m400 has two entries in the table: they correspond to two different MD updates spacings between saves of the gauge field configurations during the execution of the SMD algorithm. The save interval was first set to a small value, but a preliminary analysis of τ_Q suggested that the save interval was too small, as there was a large autocorrelation between the configurations, $\tau_Q = 2.25$. This is to stress the importance of this analysis.

From the results for τ_π, τ_p and τ_Ω we can observe that for hadronic observables we observe that a064m4001 and a12m400 have values for the integrated autocorrelation time of at least one hadron that are considerably larger than the optimal value of 0.5, as explained in Appendix 1A.1. The solution we adopted has been to use statistical blocking for these ensembles, see Appendix 1A.1.1, with block size 2, before analyzing the data to decorrelate the samples.

3.4.2   A Bayesian Analysis Framework for Hadron Masses

In Chapter 2 we have extensively presented results of hadron correlators and effective masses. However, to the modern lattice practitioner, the analysis used in the previous chapter might have appeared too simplistic. In particular, we relied on constant fits for the effective masses. While this is still a widely used approach, an increasing number of precision studies now use more sophisticated techniques, even for basic hadron spectroscopy.

One of the main issues with the constant fit approach is the large bias that is introduced in selecting the fitting range from the effective mass. While this might not seem like a large problem in the case of mesons, where the signal does not degrade towards large source-sink separation times, in the case of baryons, where the signal-to-noise problem arises, this bias can sometimes be non-negligible. However, the systematic error introduced by choice of the fitting range is not necessarily always investigated, nor is there a clear framework for how to estimate it.

One apparently obvious improvement over constant fits of the effective mass is to improve the approximating functional form. In particular, since we know that correlators are a sum of decaying exponentials as shown in Section 1.4, one could try directly fitting the correlator this way. Such an approach, usually called in jargon "multi-state fitting", is valid and gives good results; however, the fitting algorithms tend to be unstable for such functionals due to their non-linearity. Furthermore, the higher excited states have a much sharper decay due to their large exponent, which in turn leaves almost no signal on the correlator. These two problems lead to standard fitting procedures being somewhat unreliable and biased in the case of multi-state fits.

Bayesian Fitting with Constraints   A solution to the problem of the stability of multi-state fits for hadron correlators can be the introduction of educated initial guesses for the fit parameters. These initial values, or *priors*, introduce bias in the end results of the fit, which needs to be accounted for. Luckily, Bayesian statistics can help to deal with this issue in a systematic and mathematically rigorous way.

The approach to Bayesian curve fitting in the presence of prior constraints applied to the

specific case of lattice correlator data was first introduced by Lepage et al. in [34]. The starting point is to consider the exact form of the correlator from theory:

$$C_{th}(t, A_n, E_n) = \sum_{n=1}^{\infty} A_n e^{-E_n t}, \qquad (3.9)$$

where A_n are the amplitudes of the wavefunction corresponding to the energy levels E_n. A traditional fit would proceed by truncating the sum up to a fixed number of stated N_s and considering all the A_n and E_n as parameters. Then one would minimize the χ^2 function defined as:

$$\chi^2(A_n, E_n) = \sum_{t,t'} \frac{\Delta C(t) \Delta C(t')}{\sigma_{t,t'}^2}, \qquad (3.10)$$

where $\Delta C(t) = \overline{C(t)} - C_{th}(t, A_n, E_n)$, with $\overline{C(t)}$ begin the ensemble average data for the correlator at time t. The term $\sigma_{t,t'}^2 = \overline{C(t)C(t')} - \overline{C(t)}\,\overline{C(t')}$ is the correlation matrix instead. The first consideration to make is that contributions of high-energy states are suppressed for large t, which means that there exists a time $t_{\min}$ above which only a finite number of states are contributing, hence truncating the infinite sum in the theoretical formula. Choosing a too small value for $t_{\min}$ would not be beneficial for fitting purposes and would bias the values of A_n, E_n away from their true values, while a too large value would lead to an effective loss of information leading to increased statistical errors. Without a rigorous way of determining the systematic uncertainties, this choice of $t_{\min}$ is biased and necessarily ad hoc, the same way the choice of a fitting window for the constant case is. A good procedure would then be to test all possible values of $t_{\min}$ over a range of maximum numbers of states in the fit N_s.

To constrain the fits and improve stability a choice of priors $\tilde{A}_n, \tilde{E}_n$ for the amplitudes and energies of the states is made. Then the χ^2 is redefined as:

$$\chi^2 \rightarrow \chi^2_{aug} = \chi^2 + \chi^2_{prior} \quad \text{with} \ \chi^2_{prior} = \sum_{N_s} \frac{A_n - \tilde{A}_n}{\sigma_{\tilde{A}_n}^2} + \sum_{N_s} \frac{E_n - \tilde{E}_n}{\sigma_{\tilde{E}_n}^2} \qquad (3.11)$$

We should stress that the priors $\tilde{A}_n, \tilde{E}_n$ and their uncertainties $\sigma_{\tilde{A}_n}^2, \sigma_{\tilde{E}_n}^2$ are inputs to the fit. This augmented χ-square function χ^2_{aug} will then favor fits with the final parameters close to the prior. It is then crucial to select the Gaussian uncertainty of the priors to a reasonable value that is not

too constraining. With these definitions, the constrained fitting procedure is then to fit the data for all possible values of $t_{\min}$ and N_s is increased until the parameter of interest converges.

It is interesting to notice that the final value of χ^2_{aug} is not of particular interest in the case of constrained fits. What really matters is the stability and the convergence of the fits as $t_{\min}$ changes and N_s is increased.

Bayesian Model Averaging   To further determine an unbiased value for the mass of the hadrons, we use a Bayesian model averaging procedure based on the Akaike Information Criterion (AIC) that has been recently proposed in [35] for the case of lattice correlator data with data windowing selection. This procedure further eliminates the need to "manually" check for convergence in the constrained fitting procedure as the number of states. In turn, it defines a weight for every fit result, parametrized by the pair $t\mathrm{min}, N_s$ that is used to determine the weighted average of the fits.

Given a fit result with parameters $M = A_1, \ldots A_n, E_1, \ldots E_n$ the AIC is:

$$\mathrm{AIC}_M = -2 \log(\mathrm{pr}(M)) + \chi^2_{\mathrm{aug}} + 2k + 2n \tag{3.12}$$

where $\mathrm{pr}(M)$ is the model likelihood function of the model, k is the number of fit parameters, including priors, and n is the number of excluded points from the fit, i.e. it depends on changes of $t_{\min}$. The minimum value of the AIC among all models is used to determine the relative likelihood of a given model M_i and some data D as:

$$\mathrm{pr}(M_i|D) = \frac{\exp(\mathrm{AIC}_{\min} - \mathrm{AIC}_i)}{\sum_j \exp(\mathrm{AIC}_{\min} - \mathrm{AIC}_j)} \tag{3.13}$$

where the denominator is just to fix the normalization to allow a probabilistic interpretation of the quantity. The relative likelihood is then used as weight for computing the weighted sum of a fit parameter $\rho \in M$. The unbiased estimator for the parameter is then its weighted average and its uncertainty is given by:

$$\sigma_\rho = \sum_i \sigma_{\rho,i} \mathrm{pr}(M_i|D) + \sum_i \langle \rho \rangle_i^2 \mathrm{pr}(M_i|D) - \left(\sum_i \langle \rho \rangle_i \mathrm{pr}(M_i|D) \right)^2 \tag{3.14}$$

There are a few considerations to make on this procedure. The first and most important is that it requires no input whatsoever: the whole fitting procedure only now requires making an educated guess for the priors and attaching to their values a reasonably large uncertainty such that they do not become dominant in the fit. No further input is required. Another interesting feature that can be immediately inferred from Equation (3.12) is that this method favors models that a simpler, i.e. fit functions with fewer number of states. This sort of built-in Occam's razor is very useful to a posteriori gain intuition on how many states can be determined from the given set of data. At the same time, Equation (3.12) tells us that this procedure favors models with larger fitting ranges, hence more data, which is certainly a feature that we implicitly desire.

3.4.3   Light Hadron Spectrum and Scaling from SWF Ensembles

In order to compute the light hadron spectrum on the SWF ensembles we have implemented the exponential clover action in the `Chroma` [36] software package, for both the QDP++ (traditional CPU) and QDP-JIT (just-in-time compiled architecture-independent) backends. The propagators have been computed using the BiCGSTAB solver from the highly optimized QUDA GPU package. In addition we utilized the `lalibe` [37] software, which builds on top of `Chroma` to calculate fermion contractions.

For this study, we decided to use Gaussian invariant smearing [38], see Section 1.3.4, and in order to choose the correct parameters, a quick study of the smearing results was conducted on 200 configurations with $a = 0.094$ fm. The results can be seen in Figure 3.11. When dealing with smearing, one needs to choose the parameters in order to suppress the higher excited states and reach the plateau earlier. However, if the smearing is excessive, then the signal starts to degrade and the statistical errors increase. In our case we opted for $N_{\mathrm{smear}} = 32$ and $\sigma = 3.86$, where σ, following the notation used in Section 1.3.4, is $\sqrt{4\alpha N_{\mathrm{smear}}}$ and represents a dimensionless smearing radius.

A common technique in lattice QCD calculations is the so-called "variational method" [39, 40]. In this context, it means that one computes a given correlation function with different combinations
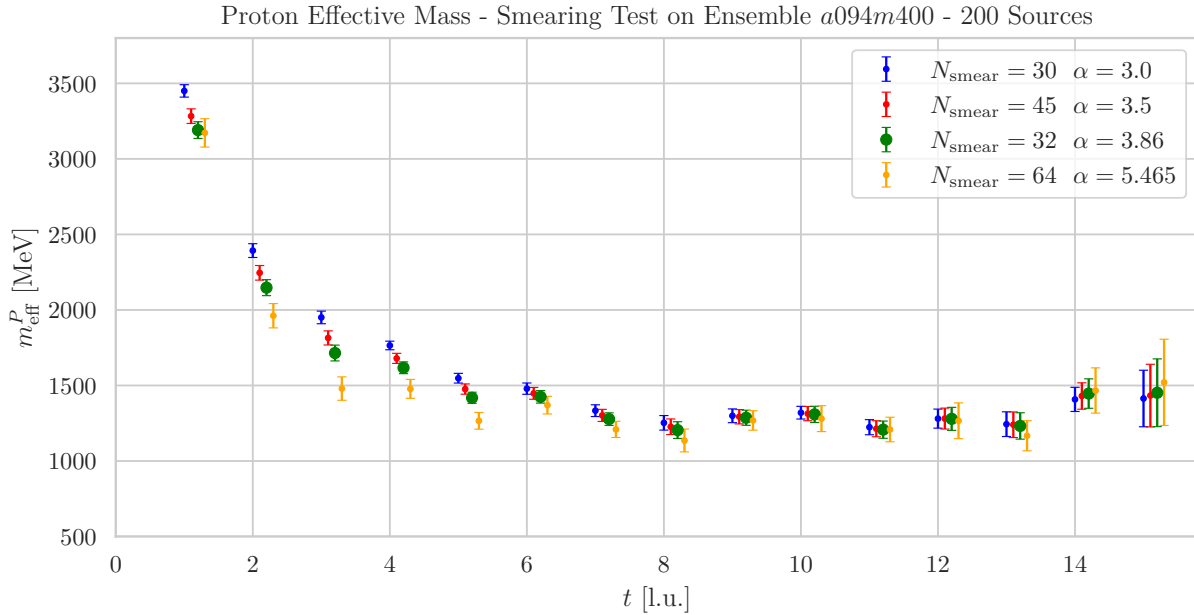
Figure 3.11: Proton effective mass computed with different Gaussian smearing parameters. The data is from the ensemble a094m400, 200 configurations, 10 sources per configuration with smeared-smeared propagators.

of interpolating operators at the source and the sink. Then a matrix is constructed with all the results and a Generalized Eigenvalue Problem (GEVP) is solved to greatly improve the signal of the lowest lying states. This procedure, however, requires computing several propagators for every source, as different interpolating operators lead to different sources and hence linear systems. Following the analysis conducted for example in [41] we used a numerically cheaper, but not so powerful, method: computing only the point-smeared (PS) and smeared-smeared (SS) correlators for every hadron. This can be done with only one propagator solve, as the two correlators share the source.

The fitting of the amplitudes and energy levels of the two correlators is performed simultaneously, utilizing the Bayesian fitting procedure highlighted above. The correlated fits have been performed with the use of the gvar and lsqfit python packages, which have been developed starting from the work done in [34]. The ansatz for the multi-state correlator, either PS or SS,

with N_s states reads:

$$C(t, Z_{P,n}, Z_{S,n}, E_n) = \sum_{n=0}^{n=N_s} Z_{S,n}^2 e^{-E_n t} + \sum_{n=0}^{n=N_s} Z_{S,n} Z_{P,n} e^{-E_n t}, \qquad (3.15)$$

where the factors $Z_{P,n}$ and $Z_{S,n}$ represent the amplitudes of the source and the sink for a given energy level E_n.

As highlighted previously, our fitting procedure only requires a choice of the priors. For the hadron spectroscopy case, we had to choose values for all the parameters E_n, $Z_{P,n}$ and $Z_{S,n}$ together with their uncertainties to use for χ_{prior}, see Equation (3.11). The priors for the amplitudes $\tilde{Z}_{P,n}$ have been chosen to be all equal to the ground state one, which is fixed from the data, as we will see later. For the $\tilde{Z}_{S,n}$ priors, the values from the second exited stated up to N_s have been taken as half the ground state one. This is motivated by the expectation that smearing should reduce the overlap with higher excited states. The uncertainties of the amplitude priors have been set by the data for the ground state and set to twice the ground state uncertainty for the excited state. So, to summarize:

$$\tilde{Z}_{P,0} \to \text{fixed from data} \qquad\qquad \tilde{Z}_{P,i} = \tilde{Z}_{P,0} \text{ for } i = 1 \dots N_s \qquad (3.16)$$

$$\tilde{Z}_{S,0} \to \text{fixed from data} \qquad \tilde{Z}_{S,1} = \tilde{Z}_{S,0}; \tilde{Z}_{S,i} = \frac{\tilde{Z}_{S,0}}{2} \text{ for } i = 2 \dots N_s$$

$$\sigma_{\tilde{Z}_{P,0}} \to \text{fixed from data} \qquad\qquad \sigma_{\tilde{Z}_{P,i}} = 2\sigma_{\tilde{Z}_{P,0}} \text{ for } i = 1 \dots N_s$$

$$\sigma_{\tilde{Z}_{S,0}} \to \text{fixed from data} \qquad \sigma_{\tilde{Z}_{S,1}} = 2\sigma_{\tilde{Z}_{S,0}}; \sigma_{\tilde{Z}_{S,i}} = \sigma_{\tilde{Z}_{S,0}} \text{ for } i = 2 \dots N_s.$$

To fix the ground state values, we plot the ground state amplitude computed with the effective mass, for both the SS or PS smearing:

$$\tilde{Z}_{S/P,0} = e^{-m_{\text{eff}}(t)t} C_{S/P}(t) \quad \text{where } m_{\text{eff}}(t) = \ln\left(\frac{C_{S/P}(t+1)}{C_{S/P}(t)}\right), \qquad (3.17)$$

where $C_{S/P}(t)$ is the short-hand notation for the two-point correlator computed either with SS or PS respectively. In Figure 3.12 we show an example of the prior fixing procedure for $\tilde{Z}_{P,0}$ and $\tilde{Z}_{S,0}$ from the data, noting that the gray bands in the plots represent the $\pm\sigma_{\tilde{Z}_{S/P,0}}$ band, which are taken to be approximately 10 times larger than the expected uncertainties of the real amplitudes.

This is to stress that the bias introduced by this prior choice is really minimal, as it would be extremely hard to argue that the true value of the amplitude does not lie in the shaded region. From the values of $\tilde{Z}_{S/P,0}$ and their corresponding $\sigma_{\tilde{Z}_{S/P,0}}$, the values for all the other priors are set as we explained earlier.
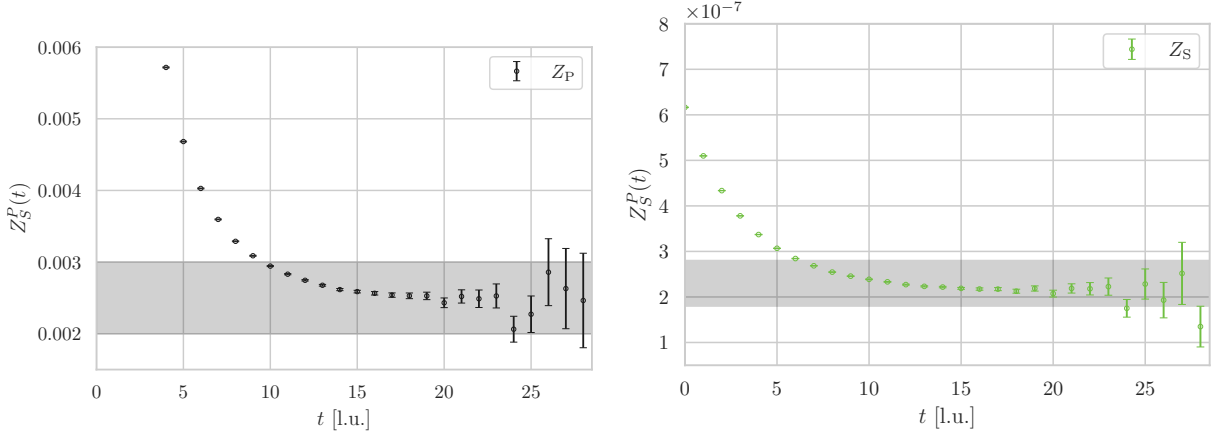
Figure 3.12: Amplitudes for the point-smeared (left) and smeared-smeared (right) correlators and the respective prior choices. The data is from the $a094m400$ ensemble, with 100 stochastic sources per configuration.

The choice of the energy priors is made in a similar manner, with the difference being that a logarithmic prior is used on the difference between the energy levels to force the order of the energy states and improve the stability of the fit. In practice, we fix the prior $\tilde{E}_0$ and its associated $\sigma_{\tilde{E}_0}$ by looking at the effective mass plot, again following the rule of thumb of 10 times the expected final uncertainty for $\sigma_{\tilde{E}_0}$. All the other energy priors are set to be ordered in steps of $2m_\pi$ above the ground state, with equal $\sigma_{\tilde{E}_n}$. In other terms:

$$\tilde{E}_0 \to \text{fixed from data} \qquad \ln(\tilde{E}_i - \tilde{E}_{i-1}) = \ln(2m_\pi) \ \text{ for } i = 1 \ldots N_s \qquad (3.18)$$

$$\sigma_{\tilde{E}_0} \to \text{fixed from data} \qquad \sigma_{\tilde{E}_i} = \sigma_{\tilde{E}_0} \text{for } i = 2 \ldots N_s.$$

The value of m_π to be used in fixing the priors does not need to be very precise, because of the very large size of $\sigma_{\tilde{E}_n}$, so it can be determined with a rough constant fit to the effective mass, without fear of impacting the final results.

116

In Figure 3.13 we show the procedure for fixing the ground state energy priors and widths from the pion and proton effective mass. Again, the shaded region is the width of the prior.
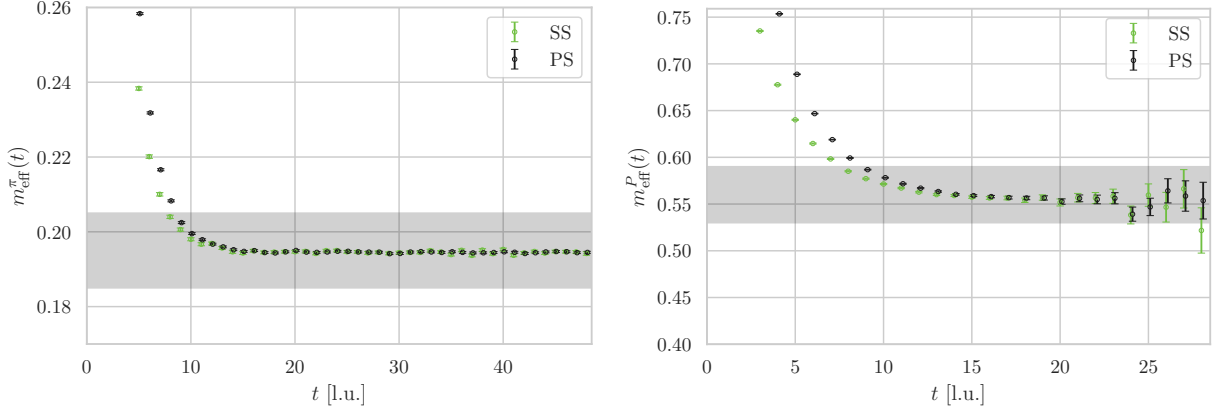
Figure 3.13: Effective mass for the pion (left) and proton (right) correlators and the respective prior choice for the ground state energy E_0. The data is from the $a094m400$ ensemble, with 100 stochastic sources per configuration.

With this information, we have all the required inputs for our fits. As outlined in the previous section, we perform a scan over a range of $t_{\min}$, while the maximum value is fixed to the maximum value for the mesons and to a value where it does not affect the fit results due to the low signal-to-noise. Then we also perform fits in the range of $N_S = 1, \ldots, 6$. All the results are then combined using model averaging. In Figures 3.14 to 3.16 we show our results for the a094m400 ensemble for the pion, the proton and the Ω baryon, as these are the only possible hadrons at the flavor-symmetric point.

The first feature to notice is that the fits all converge as N_S increases to the same value. This is the expected behavior, as first discussed in [34]. The low N_S states only agree with the other results when $t_{\min}$ is large enough so that effectively no information on the higher states is present. This can also be inferred by looking at the final fitted parameters for the excited states for large $t_{\min}$, which end up being almost unvaried from their prior value, meaning that the data and the fitting procedure pose no constraint on those parameters. It should be noted that the height of the top panel in Figures 3.14 to 3.16 is the width of the prior of the ground state energy. The gray band in all plots shows the model average result and error. We can see that it is a good

117
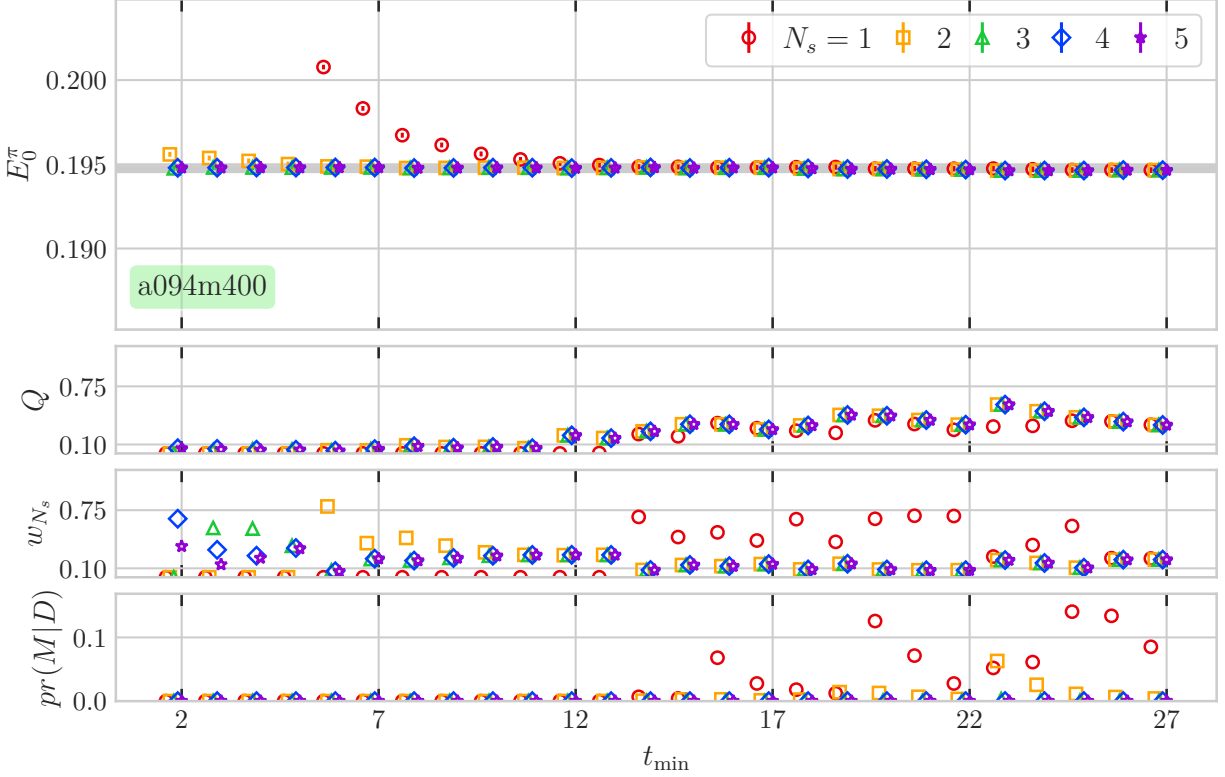
Figure 3.14: Fit results for the pion mass for the a094m400 ensemble.

representation of the data and is indeed compatible with all the converged fits and its uncertainty is not significantly smaller; on the contrary, it is of the same order. This is to maybe go against the myth that Bayesian fitting artificially decreases the final uncertainty.

The second panel of each figure shows the Q factor, or p-value, of all the fits, which is the probability that the χ^2 from the fit could have been larger, assuming the best-fit model is correct and that the data are all Gaussian and consistent with each other. In general, good fits have $Q > 0.1$. The third panel represents the normalized Gaussian Bayes Factor (GBF) w_{N_s} of the fits at fixed $t_{\min}$ as N_S is changed. The GBF, or $\mathrm{pr}(D|M)$, is the probability of obtaining the fit data by randomly sampling the parameter model used in the fit. When comparing fit models at fixed $t_{\min}$, the ones with higher w_{N_s} will be more likely to represent the data.

From this data, we can infer that some fits have very poor quality, as expected, especially for low number of states and low $t_{\min}$, this is apparent when looking at the Q factor. The information
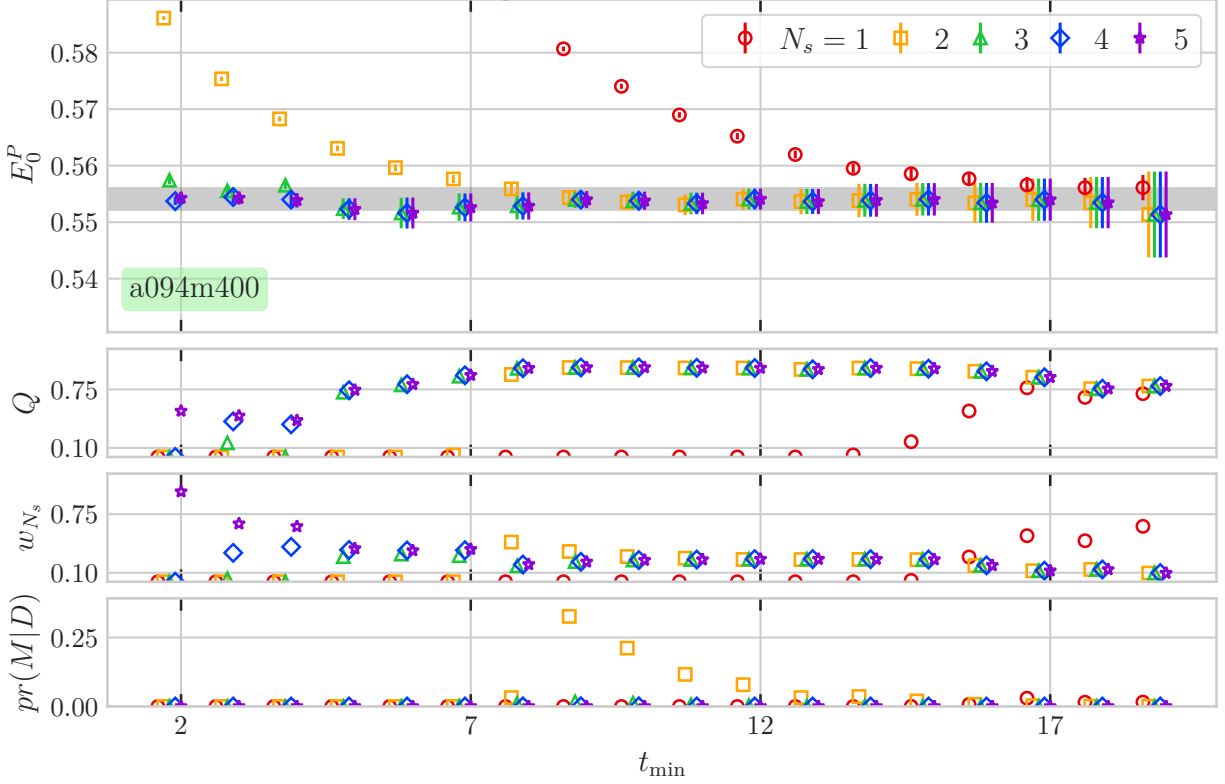
Figure 3.15: Fit results for the proton mass for the a094m400 ensemble.

of w_{N_S}, on the contrary, can help determine whether a certain number of states is too large. This happens when all fits with more than n states, for a given value of $t_{\min}$, have similar w_{N_S}, which implies that adding additional states to the fit function does not affect the fit.

However, what in the end impacts our final estimator is the $\mathrm{pr}(M|D)$, shown in the bottom panel in the figures, which, as we explained in the previous section, is used a weight when averaging the models. It is interesting to notice that the models that have a significant weight have $N_S = 1, 2$. This is expected, as from Equation (3.12), we see that fits with large N_S are exponentially suppressed in weight.

With this procedure, repeated for all ensembles, we performed a first study of the scaling of hadron masses towards the continuum for SWF ensemble at the flavor symmetric point; all the results for the individual fits can be found in Appendix 3A. First, in Figure 3.17 we computed ϕ_4 according to Equation (3.8), to check the mass tunings of the ensembles. According to the quality
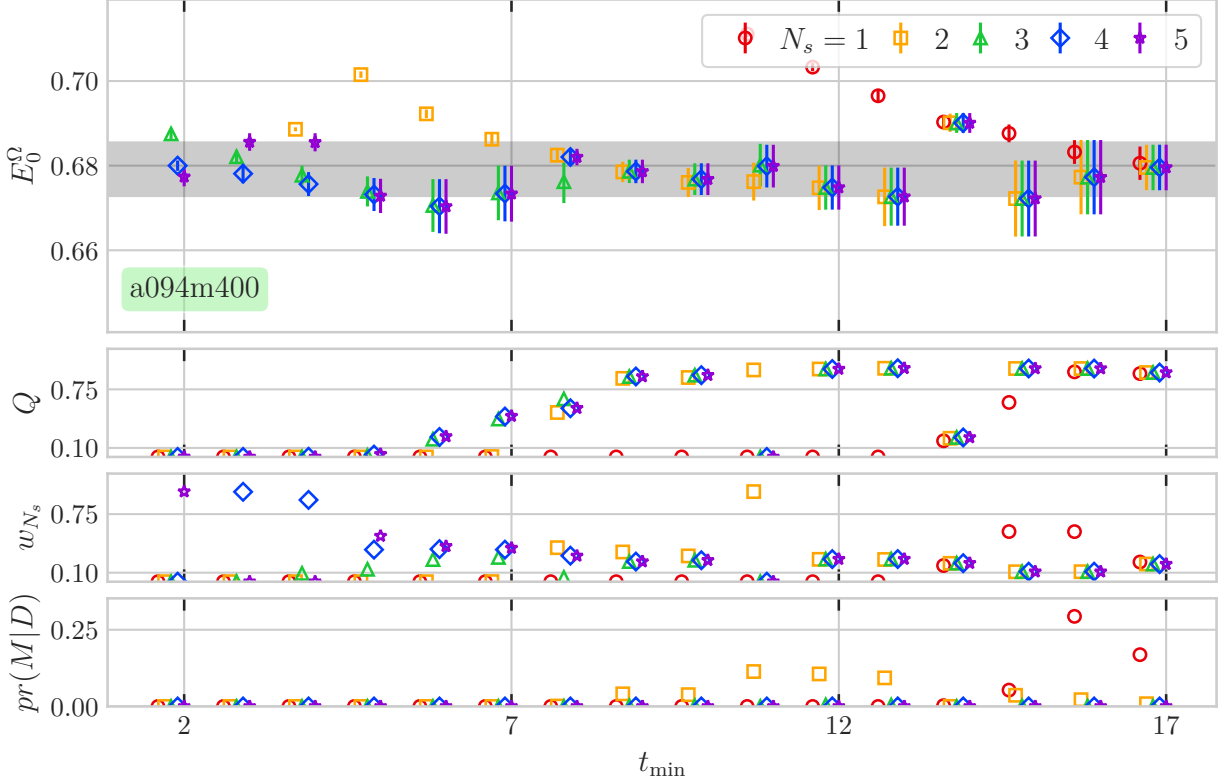
Figure 3.16: Fit results for the Ω baryon mass for the `a094m400` ensemble.

criterion that was imposed, the value of ϕ_4 should be within the gray band at the 1σ level, which is indeed the case, so the tuning has been performed correctly.

In Figure 3.18, we show the results, from which we can indeed notice that SWF show almost no sign of cutoff effects for hadronic observables, they are, in fact, at the level of 1%. The continuum limit extrapolations we for the masses of the proton and Ω are $m_p = 1175.0 \pm 6.4$ MeV and $m_\Omega = 1454.7 \pm 16.8$, the which uncertainties of $\approx 0.5\%$ and $\approx 1.1\%$ respectively are on par with other determinations [41].
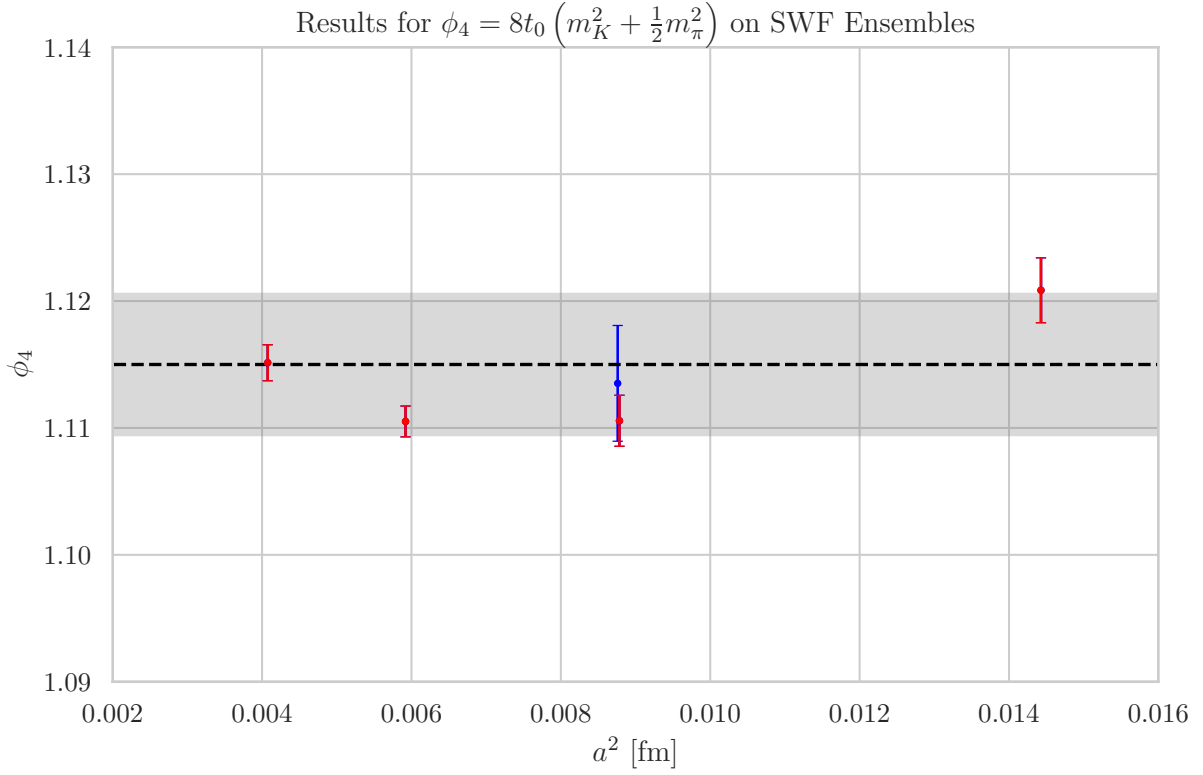
Figure 3.17: Values of ϕ_4 for SWF at the $\mathrm{SU}(3)$ flavor-symmetric point.
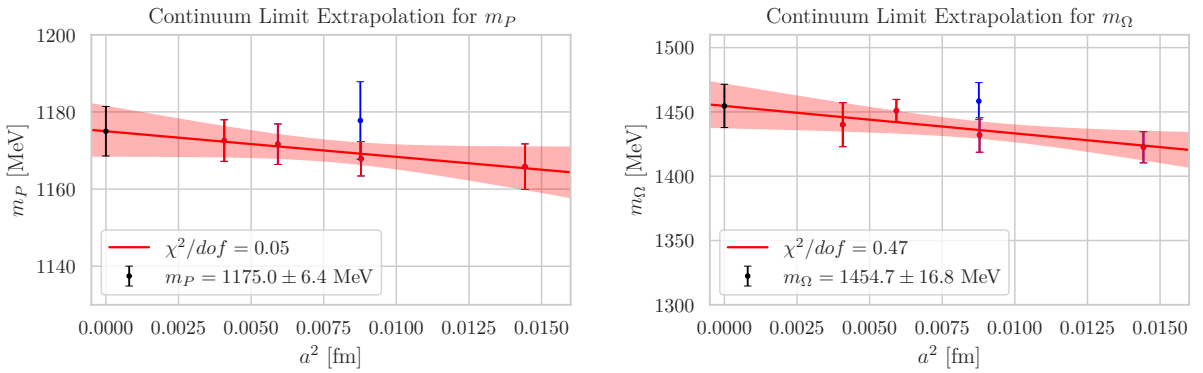
Figure 3.18: Continuum limit extrapolation for the proton (left) and Ω (right) masses for SWF at the $\mathrm{SU}(3)$ flavor-symmetric point.

3.5 Outlook for Studies with SWF

The results for the scaling properties of the hadron spectrum are very encouraging and show the validity and quality of the new SWF ensembles that have been generated by the OPENLAT initiative. At the most recent International Symposium on Lattice Field Theories, there have been several works presented that used SWF, with topics in spectroscopy [4, 24], finite temperature and master-field simulations [6]. Another current study, which we should mention, is the ongoing calculation of CP violating operators for searches of neutron electric dipole moment [42], which should see its first results in the upcoming year. Some of these studies are still in the preliminary stage of what are calculations that have been previously done on other ensembles using SWF configurations instead. All these results give growing evidence of reduced cutoff effects when compared to other discretizations.

At the same time, more ensembles with different physical parameters, most importantly different pion masses, are continuously being tuned and generated with an ongoing effort by the OPENLAT Initiative on several HPC facilities worldwide, and when they are ready and publicly made available, we expect to see wide adoption of the SWF ensembles for studies of Lattice QCD.

APPENDICES

APPENDIX 3A

BAYESIAN FIT RESULTS FOR SWF HADRON MASSES

In the table below we report the fit results for the SWF ensembles, from which Figures 3.17 and 3.18 are obtained. All results are obtained using the Bayesian analysis framework presented in Section 3.4.2.

Ensemble	t_0/a^2	am_π	am_p	am_Ω
a12m400	1.48548(64)	0.25075(28)	0.7096(35)	0.8659(73)
a094m400S	2.4482(50)	0.19483(40)	0.5586(47)	0.6917(68)
a094m400	2.43979(89)	0.19470(17)	0.5541(19)	0.6791(63)
a077m400	3.6198(30)	0.159903(83)	0.4568(20)	0.5658(34)
a064m400	5.2588(46)	0.132932(84)	0.3793(17)	0.4659(55)

Table 3A.1: Final results for masses from SWF ensembles.

We also include, in Figures 3A.1 to 3A.4, the stability plots and weights for the ensembles not shown in Section 3.4. It is important to notice the stability of the fitting procedure results, how they are always compatible with the vast majority of the fits and do not underestimate the uncertainties.

a12m400 Ensemble
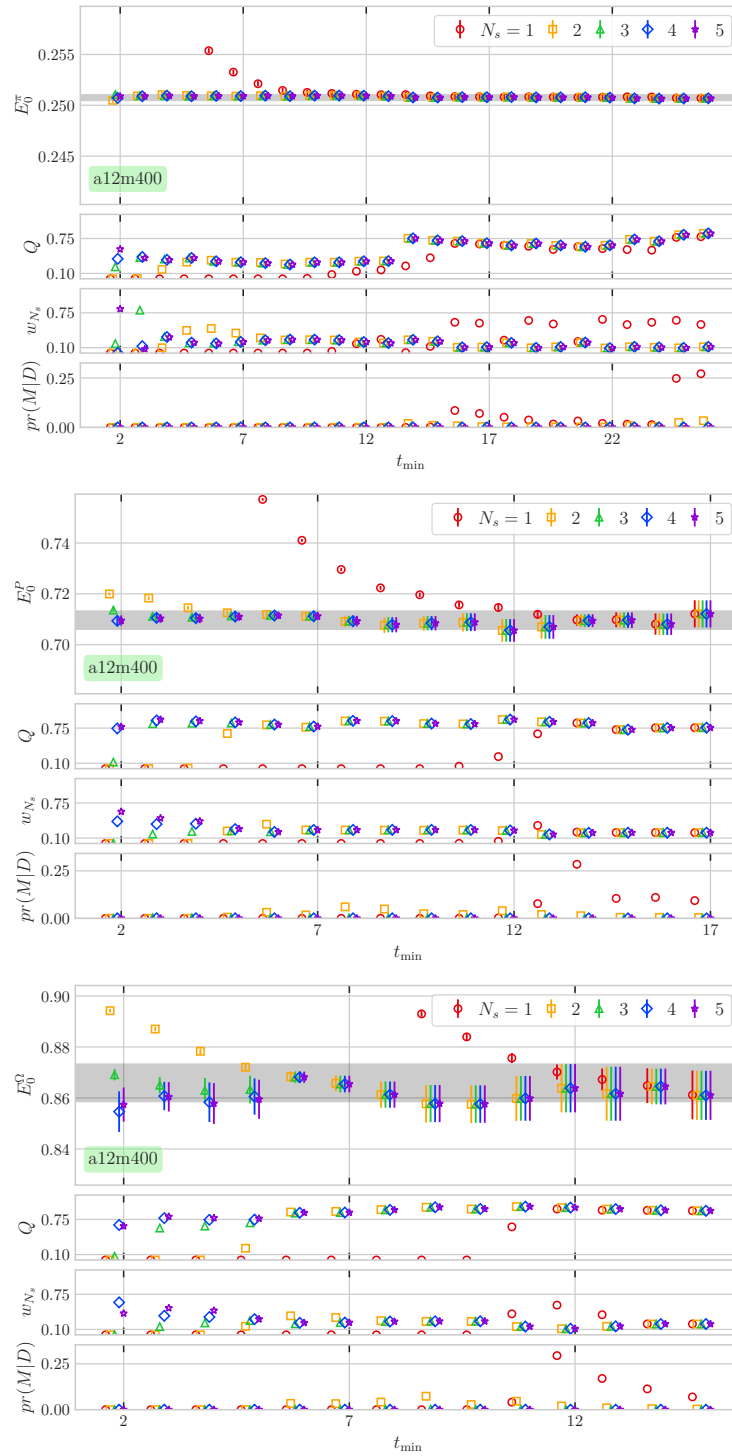
Figure 3A.1: Fit results for the for the a12m400 ensemble.

a094m400S Ensemble
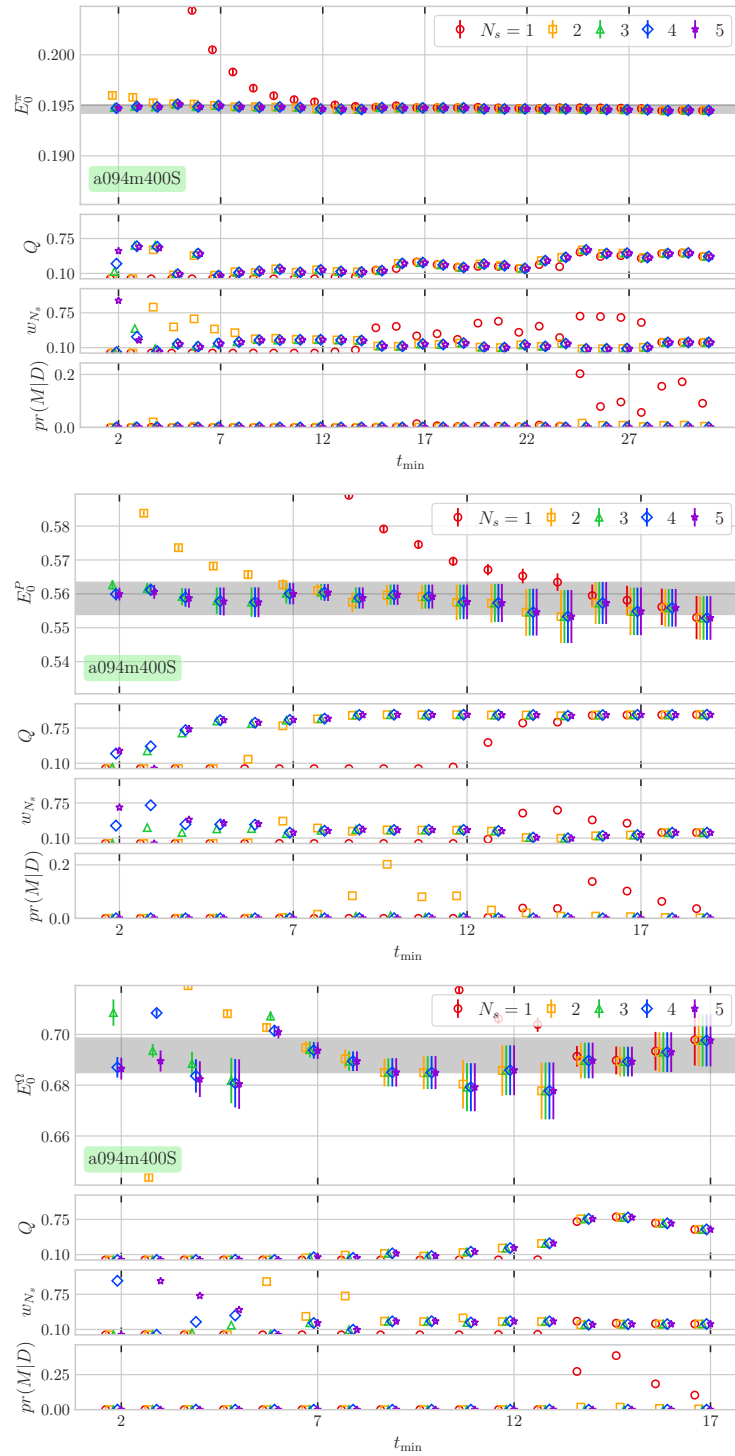
Figure 3A.2: Fit results for the for the a094m400S ensemble.

a077m400 Ensemble
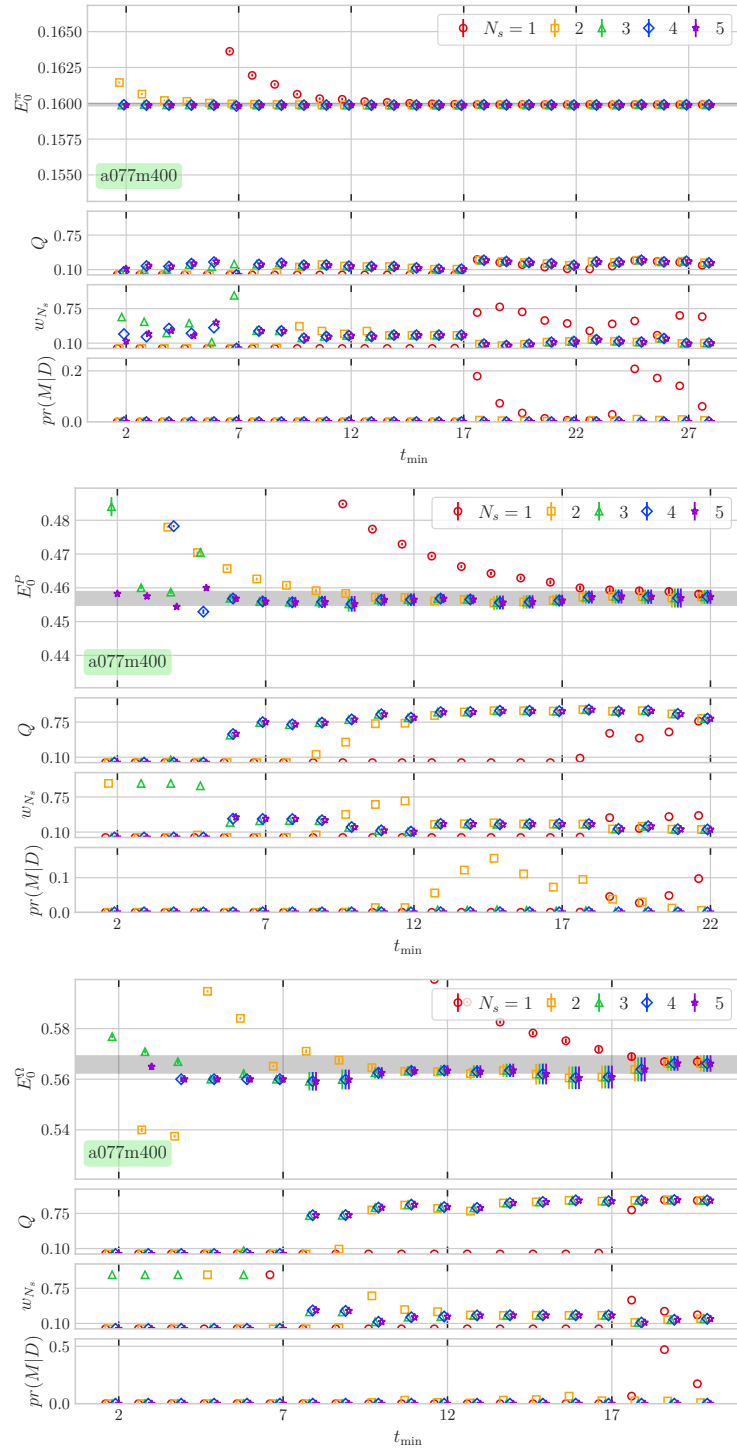
Figure 3A.3: Fit results for the for the a077m400 ensemble.

a064m400 Ensemble
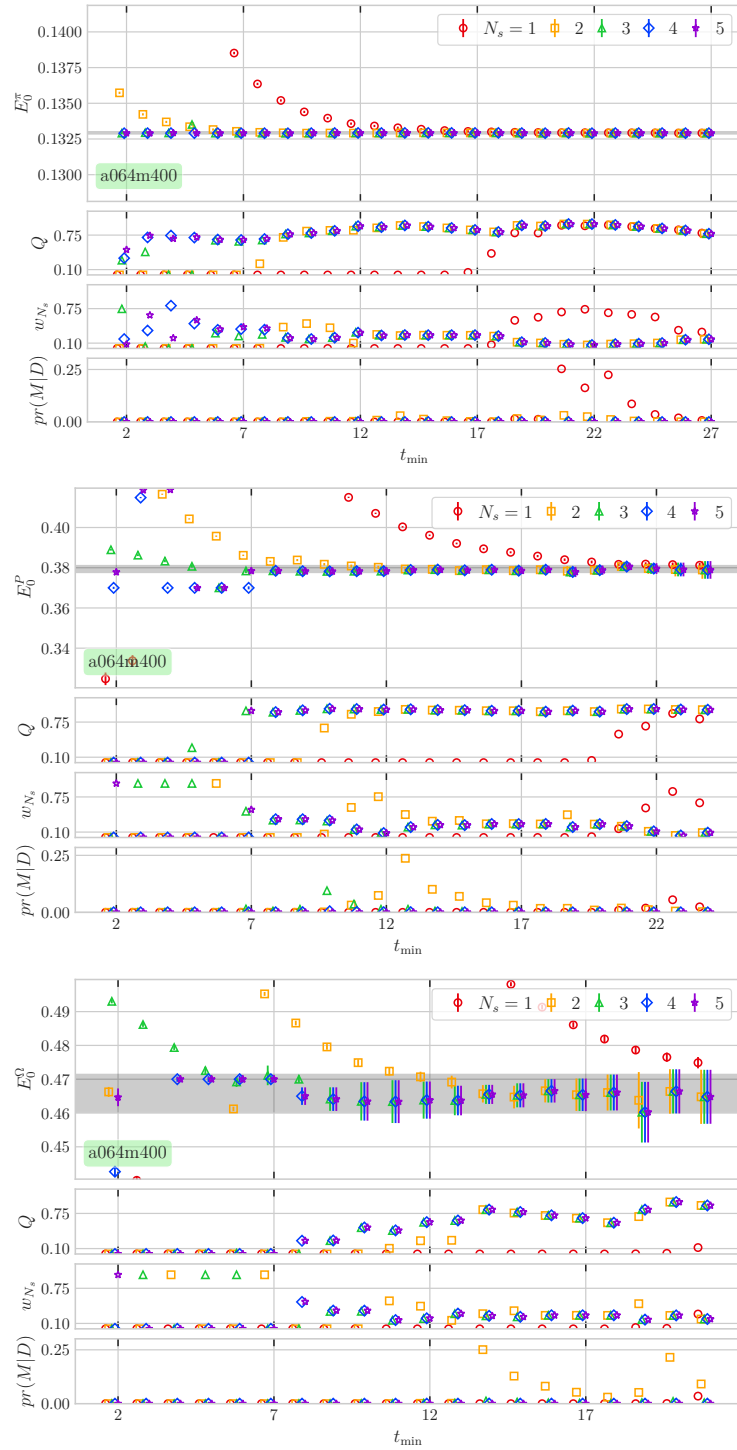
Figure 3A.4: Fit results for the for the a064m400 ensemble.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] B. Sheikholeslami and R. Wohlert, "Improved continuum limit lattice action for QCD with wilson fermions," *Nuclear Physics B*, vol. 259, no. 4, pp. 572–596, Sep. 30, 1985, ISSN: 0550-3213. DOI: 10.1016/0550-3213(85)90002-1.

[2] Y. Aoki *et al.*, "FLAG review 2021," *The European Physical Journal C*, vol. 82, no. 10, p. 869, Oct. 4, 2022, ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-022-10536-1.

[3] M. Lüscher, "Stochastic locality and master-field simulations of very large lattices," *EPJ Web of Conferences*, vol. 175, M. Della Morte, P. Fritzsch, E. Gámiz Sánchez, and C. Pena Ruano, Eds., p. 01 002, 2018, ISSN: 2100-014X. DOI: 10.1051/epjconf/201817501002.

[4] A. Francis, J. R. Green, P. M. Junnarkar, C. Miao, T. D. Rae, and H. Wittig, "Lattice QCD study of the h dibaryon using hexaquark and two-baryon interpolators," *Physical Review D*, vol. 99, no. 7, p. 074 505, Apr. 12, 2019, ISSN: 2470-0010, 2470-0029. DOI: 10.1103/PhysRevD.99.074505.

[5] D. Mohler and S. Schaefer, "Remarks on strange-quark simulations with wilson fermions," *Physical Review D*, vol. 102, no. 7, p. 074 506, Oct. 27, 2020, ISSN: 2470-0010, 2470-0029. DOI: 10.1103/PhysRevD.102.074506.

[6] P. Fritzsch, J. Bulava, M. Cè, A. Francis, M. Lüscher, and A. Rago, "Master-field simulations of QCD," in *Proceedings of The 38th International Symposium on Lattice Field Theory — PoS(LATTICE2021)*, May 16, 2022, p. 465. DOI: 10.22323/1.396.0465.

[7] M. Cè *et al.*, "Approaching the master-field: Hadronic observables in large volumes," in *Proceedings of The 38th International Symposium on Lattice Field Theory — PoS(LATTICE2021)*, May 16, 2022, p. 383. DOI: 10.22323/1.396.0383.

[8] F. Cuteri *et al.* "OPEN LATtice initiative," OPEN LATtice initiative. (2021), [Online]. Available: https://openlat1.gitlab.io/.

[9] C. Davies, A. Irving, R. Kenway, and C. Maynard, "International lattice data grid," *Nuclear Physics B - Proceedings Supplements*, vol. 119, pp. 225–226, May 2003, ISSN: 09205632. DOI: 10.1016/S0920-5632(03)01509-3.

[10] M. G. Beckett *et al.*, "Building the international lattice data grid," *Computer Physics Communications*, vol. 182, no. 6, pp. 1208–1214, Jun. 2011, ISSN: 00104655. DOI: 10.1016/j.cpc.2011.01.027.

[11] A. Francis, P. Fritzsch, M. Lüscher, and A. Rago, "Master-field simulations of o(a)-improved lattice QCD: Algorithms, stability and exactness," *Computer Physics Communications*,

vol. 255, p. 107 355, Oct. 2020, ISSN: 00104655. DOI: 10.1016/j.cpc.2020.107355.

[12] M. Lüscher, "Code available at https://luscher.web.cern.ch/luscher/openQCD/,"

[13] M. Hasenbusch, "Speeding up finite step-size updating of full QCD on the lattice," *Physical Review D*, vol. 59, no. 5, p. 054 505, Feb. 3, 1999. DOI: 10.1103/PhysRevD.59.054505.

[14] M. Hasenbusch, "Speeding up the hybrid monte carlo algorithm for dynamical fermions," *Physics Letters B*, vol. 519, no. 1, pp. 177–182, Oct. 25, 2001, ISSN: 0370-2693. DOI: 10.1016/S0370-2693(01)01102-9.

[15] C. Urbach, K. Jansen, A. Shindler, and U. Wenger, "HMC algorithm with multiple time scale integration and mass preconditioning," *Computer Physics Communications*, vol. 174, no. 2, pp. 87–98, Jan. 15, 2006, ISSN: 0010-4655. DOI: 10.1016/j.cpc.2005.08.006.

[16] M. Lüscher and S. Schaefer, "Lattice QCD with open boundary conditions and twisted-mass reweighting," *Computer Physics Communications*, vol. 184, no. 3, pp. 519–528, Mar. 1, 2013, ISSN: 0010-4655. DOI: 10.1016/j.cpc.2012.10.003.

[17] M. Lüscher, "Lattice QCD and the schwarz alternating procedure," *Journal of High Energy Physics*, vol. 2003, no. 5, pp. 052–052, May 20, 2003, ISSN: 1029-8479. DOI: 10.1088/1126-6708/2003/05/052.

[18] M. Lüscher, "Schwarz-preconditioned HMC algorithm for two-flavor lattice QCD," *Computer Physics Communications*, vol. 165, no. 3, pp. 199–220, Feb. 1, 2005, ISSN: 0010-4655. DOI: 10.1016/j.cpc.2004.10.004.

[19] A. Horowitz, "Stochastic quantization in phase space," *Physics Letters B*, vol. 156, no. 1, pp. 89–92, Jun. 1985, ISSN: 03702693. DOI: 10.1016/0370-2693(85)91360-7.

[20] A. M. Horowitz, "The second order langevin equation and numerical simulations," *Nucl. Phys. B*, vol. 280, p. 510, 1987. DOI: 10.1016/0550-3213(87)90159-3.

[21] A. M. Horowitz, "A generalized guided monte carlo algorithm," *Physics Letters B*, vol. 268, no. 2, pp. 247–252, Oct. 1991, ISSN: 03702693. DOI: 10.1016/0370-2693(91)90812-5.

[22] K. Jansen and C. Liu, "Kramers equation algorithm for simulations of QCD with two flavors of wilson fermions and gauge group SU (2)," *Nuclear Physics B*, vol. 453, no. 1, pp. 375–392, Oct. 1995, ISSN: 05503213. DOI: 10.1016/0550-3213(95)00427-T.

[23] M. Lüscher, "Ergodicity of the SMD algorithm in lattice QCD, unpublished notes (2017), http://luscher.web.cern.ch/luscher/notes/smd-ergodicity.pdf,"

[24] A. S. Francis *et al.*, "Properties, ensembles and hadron spectra with stabilised wilson fermions," in *Proceedings of The 38th International Symposium on Lattice Field Theory —*

PoS(LATTICE2021), May 16, 2022, p. 118. DOI: 10.22323/1.396.0118.

[25] M. Lüscher, S. Sint, R. Sommer, P. Weisz, and U. Wolff, "Non-perturbative o(a) improvement of lattice QCD," *Nuclear Physics B*, vol. 491, no. 1, pp. 323–343, Apr. 28, 1997, ISSN: 0550-3213. DOI: 10.1016/S0550-3213(97)00080-1.

[26] J. Bulava and S. Schaefer, "Improvement of n f = 3 lattice QCD with wilson fermions and tree-level improved gauge action," *Nuclear Physics B*, vol. 874, no. 1, pp. 188–197, Sep. 2013, ISSN: 05503213. DOI: 10.1016/j.nuclphysb.2013.05.019.

[27] W. Bietenholz *et al.*, "Tuning the strange quark mass in lattice simulations," *Physics Letters B*, vol. 690, no. 4, pp. 436–441, Jun. 2010, ISSN: 03702693. DOI: 10.1016/j.physletb.2010.05.067.

[28] M. Bruno, T. Korzec, and S. Schaefer, "Setting the scale for the CLS 2 + 1 flavor ensembles," *Physical Review D*, vol. 95, no. 7, p. 074 504, Apr. 12, 2017, ISSN: 2470-0010, 2470-0029. DOI: 10.1103/PhysRevD.95.074504.

[29] S. Aoki *et al.*, "Review of lattice results concerning low-energy particle physics: Flavour lattice averaging group (FLAG)," *The European Physical Journal C*, vol. 77, no. 2, p. 112, Feb. 2017, ISSN: 1434-6044, 1434-6052. DOI: 10.1140/epjc/s10052-016-4509-7.

[30] M. Lüscher, "Properties and uses of the wilson flow in lattice QCD," *Journal of High Energy Physics*, vol. 2010, no. 8, p. 71, Aug. 2010, ISSN: 1029-8479. DOI: 10.1007/JHEP08(2010)071.

[31] M. Bruno *et al.*, "Simulation of QCD with n f = 2 + 1 flavors of non-perturbatively improved wilson fermions," *Journal of High Energy Physics*, vol. 2015, no. 2, p. 43, Feb. 2015, ISSN: 1029-8479. DOI: 10.1007/JHEP02(2015)043.

[32] T. Takaishi and P. de Forcrand, "Testing and tuning symplectic integrators for the hybrid monte carlo algorithm in lattice QCD," *Physical Review E*, vol. 73, no. 3, p. 036 706, Mar. 17, 2006, ISSN: 1539-3755, 1550-2376. DOI: 10.1103/PhysRevE.73.036706.

[33] M. Lüscher and S. Schaefer, "Lattice QCD without topology barriers," *Journal of High Energy Physics*, vol. 2011, no. 7, p. 36, Jul. 7, 2011, ISSN: 1029-8479. DOI: 10.1007/JHEP07(2011)036.

[34] G. P. Lepage *et al.*, "Constrained curve fitting," *Nuclear Physics B - Proceedings Supplements*, vol. 106-107, pp. 12–20, 2002, ISSN: 0920-5632. DOI: https://doi.org/10.1016/S0920-5632(01)01638-3.

[35] W. I. Jay and E. T. Neil, "Bayesian model averaging for analysis of lattice field theory results," *Physical Review D*, vol. 103, no. 11, p. 114 502, Jun. 9, 2021, ISSN: 2470-0010, 2470-0029. DOI: 10.1103/PhysRevD.103.114502.

[36]   R. G. Edwards and B. Joó, "The chroma software system for lattice QCD," *Nuclear Physics B - Proceedings Supplements*, vol. 140, pp. 832–834, Mar. 2005, ISSN: 09205632. DOI: 10.1016/j.nuclphysbps.2004.11.254.

[37]   A. Gambhir *et al.*, "Code available at https://github.com/callat-qcd/lalibe,"

[38]   S. Güsken, "A study of smearing techniques for hadron correlation functions," *Nuclear Physics B - Proceedings Supplements*, vol. 17, pp. 361–364, Sep. 1, 1990, ISSN: 0920-5632. DOI: 10.1016/0920-5632(90)90273-W.

[39]   B. J. Owen *et al.*, "Variational approach to the calculation of g a," *Physics Letters B*, vol. 723, no. 1, pp. 217–223, Jun. 2013, ISSN: 03702693. DOI: 10.1016/j.physletb.2013.04.063.

[40]   J. Dragos *et al.*, "Nucleon matrix elements using the variational method in lattice QCD," *Physical Review D*, vol. 94, no. 7, p. 074 505, Oct. 25, 2016, ISSN: 2470-0010, 2470-0029. DOI: 10.1103/PhysRevD.94.074505.

[41]   N. Miller *et al.*, "Scale setting the möbius domain wall fermion on gradient-flowed HISQ action using the omega baryon mass and the gradient-flow scales t 0 and w 0," *Physical Review D*, vol. 103, no. 5, p. 054 511, Mar. 24, 2021, ISSN: 2470-0010, 2470-0029. DOI: 10.1103/PhysRevD.103.054511.

[42]   J. Dragos, T. Luu, A. Shindler, J. de Vries, and A. Yousif, "Confirming the existence of the strong CP problem in lattice QCD with the gradient flow," *Physical Review C*, vol. 103, no. 1, p. 015 202, Jan. 19, 2021. DOI: 10.1103/PhysRevC.103.015202.

CHAPTER 4

QUANTUM STATE PREPARATION ALGORITHMS FOR THE SCHWINGER MODEL

Despite the ever-increasing computational capacities of modern HPC facilities, one issue with LFT calculations seems impossible to solve: the sign problem. This arises when sampling a PDF in the MCMC that is not strictly positive. Clearly, this implies that the PDF is not actually a probability distribution, but an approximation. Since the PDF in LQCD comes from the exponential of the Euclidean action and the fermion determinants, one must ensure their positivity, but this is not possible in the presence of chemical potentials [1], topological terms [2, 3] or real-time evolution [4].

A possible solution would be to switch to Hamiltonian formalism, where the sign problem is absent, but that would lead to large vector spaces that grow exponentially with the number of lattice sites, which are untreatable even on modern supercomputers. One way to deal with this formalism would be to utilize some computing machine that can easily deal with these bases. Luckily, such machines do exist: Quantum Computers.

The field of Quantum Computing (QC) made giant leaps forward in the last decade, mostly thanks to the experimental improvements that led to the creation of the first hardware. The idea of quantum computers was first proposed in the 70s, with the development of the field of quantum information [5–7]. Later it was suggested as a mean to simulate quantum problems directly [8–10]. The main advantage they have as opposed to classical computers, in the context of Physics applications, is that they have a natural way to implement the antisymmetry requirements of fermionic systems, which will become important as we will discuss in Section 4.2.1. In more general terms, quantum computers use *qubits* as their basic information unit as opposed to bits. Operations on qubits are performed using *quantum gates* as opposed to binary logic gates.

Currently, we are in the so-called Noisy Intermediate Scale Quantum (NISQ) era of hardware, meaning that current state-of-the-art machines have between tens to a few hundreds qubits, which are not enough for designing fault-tolerant algorithms, hence the noisy, and are also not

enough for "quantum supremacy" (that is solving a problem faster than any classical computer could). Current hardware designs are varied [11], some of them are based on superconducting circuits [12], trapped ions [13], optical lattices [14] or solid-state systems [15].

The model for computations that we will employ in this work is the quantum gate array, in which an algorithm is represented by a chain of quantum gate operations applied to the qubits of the system. In this model, a physics simulation can be performed by first *preparing a state* of interest, then letting it evolve in real-time with the Hamiltonian and finally measuring an observable. Each of the three phases of the computation presents its challenges and is actively being studied. In this work, we will focus on three algorithms for state preparation: the Adiabatic State Preparation (ASP), the Quantum Approximate Optimization Algorithm (QAOA) and the Rodeo Algorithm.

Dealing with QCD on a quantum computer is currently beyond our capabilities, though there is active research in possible ways to represent the LFTs on qubits [16–20] or by employing tensor networks instead [21]. Furthermore, with the current hardware limitations, there is no chance of reaching the scale of the simulations that are currently being done in LQCD, especially when considering limitations due to noise. There is nevertheless a strong push to investigate the feasibility of quantum computations of lattice field theories and a search for new algorithms and approaches. In this work, we will then shift our focus to the Schwinger model, which is a toy model for QCD as it shares some interesting properties like confinement.

This chapter is structured as follows: first, in Section 4.1, we will give a brief introduction to quantum computing, then, in Section 4.2, the Schwinger model will be introduced and a possible way to discretize its Hamiltonian on a quantum computer will be presented. Finally, in Sections 4.3 to 4.5, we will discuss three algorithms for state preparation and compare their results.

4.1   A Quick Tour of Quantum Computing

In computer science, the basic block of information is the bit, a binary state that can either be 0 or 1. A combination of n bits, a bit-string, can represent numbers from 0 to 2^n. In quantum

computing, one deals with qubits instead. The fundamental states are usually represented as:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{4.1}$$

Qubits, however, as opposed to bits, can be in any superposition of these two states, in particular, they can be in any state:

$$|q\rangle = a\,|0\rangle + b\,|1\rangle \qquad \text{with } a, b \in \mathbb{C}, \;\; |a|^2 + |b|^2 = 1. \tag{4.2}$$

Qubit states are normalized to one and their amplitudes can be complex. A convenient representation of one qubit can be given by the Bloch sphere, given by the parametrization based on two angles θ and ϕ:

$$|q\rangle = \cos\frac{\theta}{2}\,|0\rangle + \sin\frac{\theta}{2}e^{i\phi}\,|1\rangle. \tag{4.3}$$

Quantum states can then be represented as points on the surface of the sphere, as shown pictorially
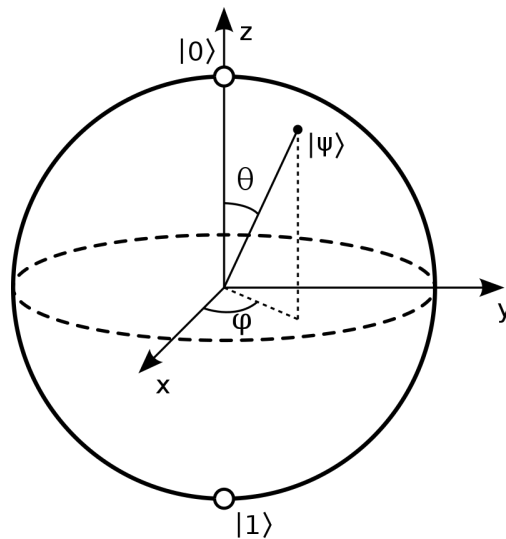
Figure 4.1: Graphic representation of the Bloch sphere. A quantum state corresponds to points on the surface of the sphere, parametrized by two angles θ and ϕ.

in Figure 4.1.

136

States composed of multiple qubits are obtained by taking the tensor product of single qubits:

$$|q_1 q_2\rangle = |q_1\rangle \otimes |q_2\rangle = \begin{pmatrix} a_1 \\ b_1 \end{pmatrix} \otimes \begin{pmatrix} a_2 \\ b_2 \end{pmatrix} = \begin{pmatrix} a_1 \begin{pmatrix} a_2 \\ b_2 \end{pmatrix} \\ b_1 \begin{pmatrix} a_2 \\ b_2 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} a_1 a_2 \\ a_1 b_2 \\ b_1 a_2 \\ b_1 b_2 \end{pmatrix}. \tag{4.4}$$

With this basic understanding of how qubits work, we can start looking at which operations one can apply to them.

4.1.1 Quantum Gate Operations

Even though most computer programmers don't think about the bit-level operations happening in their CPUs, they are fundamentally designing algorithms that are chains of logic gates, such as NOT, AND, OR and XOR and their combinations. These gates are then grouped together into opcodes of the CPU, operations that can be done in a single clock cycle like addition, multiplication or more complex ones, which take bits contained in the registry and run through the relevant logic gate path and then write the result to another registry location. Quantum computers today still don't have this last level of abstraction and algorithms are still being written in terms of single gate operations. However, the fundamental gates are not the familiar binary ones. We will now summarize the main features of such gates.

All qubit gate operations are required to be unitary transformations on the qubits they act upon. This is imposed by the constraint of preserving the normalization of the wave function. There is, in principle, no limit on the number of qubits that a single gate operation can act upon, but due to hardware constraints, NISQ machines only implement 1 and 2 qubit operations. In general, they are represented with diagrams where horizontal lines stand for single qubits and boxes and vertical lines stand for gate operations. The flow is read from left to right. For a generic 1-qubit gate, this looks like:

$$|q\rangle \qquad\qquad U\,|q\rangle$$
$$\text{---}\boxed{U}\text{---}$$
$$\text{initial state} \qquad\qquad \text{final state}$$
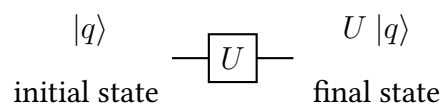
All 1-qubit can be represented by a generic $\mathrm{SU}(2)$ transformation, which requires three parameters, usually identified with three angles:

$$U(\theta, \phi, \lambda) = \begin{pmatrix} \cos \frac{\theta}{2} & -e^{i\lambda} \sin \frac{\theta}{2} \\ e^{i\phi} \sin \frac{\theta}{2} & e^{i(\phi+\lambda)} \cos \frac{\theta}{2} \end{pmatrix}. \tag{4.5}$$

This general version is, however, not very useful, so some special cases have their unique symbols and diagrams.

Pauli Gates   The first class of gates that we introduce are the single qubit gates that represent the Pauli matrices. These gates inherit the properties of Pauli matrices: for example, they anti commute and are involutory (they are their own inverse), so their square is the identity matrix. For completeness, we list them below:

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \rightarrow \quad \boxed{I}$$

$$X = \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \rightarrow \quad \boxed{X}$$

$$Y = \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad \rightarrow \quad \boxed{Y}$$

$$Z = \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad \rightarrow \quad \boxed{Z}$$

It can be interesting to notice that the X gate maps $|0\rangle \rightarrow |1\rangle$ and $|1\rangle \rightarrow |0\rangle$ so it is the quantum analogous of the NOT gate.

Rotation Gates   Another class of gates that will be very important in this work is the Rotation Gates. These represent rotations around one of the three axes X, Y, Z. These form a complete basis for $\mathrm{SU}(2)$, alternative to the Pauli basis, meaning that any gate can, in principle, be decomposed into rotations. These are defined as:

$$R_X(\theta) = e^{-i\frac{\theta}{2}X} = \begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix} \quad \rightarrow \quad \boxed{R_X(\theta)}$$

$$R_Y(\theta) = e^{-i\frac{\theta}{2}Y} = \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix} \quad \rightarrow \quad \boxed{R_Y(\theta)}$$

$$R_Z(\theta) = e^{-i\frac{\theta}{2}Z} = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix} \quad \rightarrow \quad \boxed{R_Z(\theta)}$$

Phase Shift Gates   One useful quantum gate is the Phase Shift gate. This is parametrized by one angle ϕ and is defined as:

$$P(\phi) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix} \quad \rightarrow \quad \boxed{P(\phi)}$$

This gate corresponds to a rotation along the z-axis on the Bloch sphere. It maps the basis states as: $|0\rangle \rightarrow |0\rangle$ and the $|1\rangle \rightarrow e^{i\phi}|1\rangle$. It should be noted that the phase shift gate is not Hermitian, but the relation $P^\dagger(\phi) = P(-\phi)$ holds instead.

Hadamard Gate   A crucial element in quantum computing is the possibility of creating states that are the superposition of the basis states. This is in practice done by the use of the Hadamard Gate:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad \rightarrow \quad \boxed{H}$$

In particular, it maps the basis states to states that have an equal probability of being in either basis state:

$$H|0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} |0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \tag{4.6}$$

$$H|1\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} |1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$

These states, when measured, have 50% probability of being either 0 or 1.

Controlled Two-Qubit Gates   The class of controlled 2-qubit gates is that of operations acting on two qubits where one of the two is kept constant and is used to determine the weights of the transformation on the other qubit. For a general 1-qubit operator U, its controlled version is given by:

$$CU = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{00} & u_{01} \\ 0 & 0 & u_{10} & u_{11} \end{pmatrix} \rightarrow$$

A particular case is that of the controlled X gates or, as it is most often called, the CNOT gate:

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \rightarrow$$

The CNOT gate flips the state of the target qubit based on the control qubit: if the control is $|0\rangle$, the target qubit is left unchanged; if the control is $|1\rangle$, then the target is flipped. A key use of the CNOT gate is to entangle or disentangle two qubits, with the typical example being the preparation of a Bell State with just a Hadamard and a CNOT:

$$|\psi\rangle = \tfrac{1}{\sqrt{2}}(\,|00\rangle + |11\rangle)$$

which is a maximally entangled state.

Another gate that we will employ is the Controlled Phase gate, which is simply given by:

$$CP(\phi) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\phi} \end{pmatrix} \rightarrow$$

Ising Gates   The last class of gates that we present are the so-called Ising Gates. These have special relevance for physics problems as they are often used in real-time evolution algorithms. They are two-qubit gates that represent the exponential of pairs of X, Y, Z gats acting on both qubits. Their definitions are as follows:

$$R_{XX}(\theta) = e^{-i\frac{\theta}{2}X_1 \otimes X_2} = \begin{pmatrix} \cos\frac{\theta}{2} & 0 & 0 & -i\sin\frac{\theta}{2} \\ 0 & \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} & 0 \\ 0 & -i\sin\frac{\theta}{2} & 0 & \\ -i\sin\frac{\theta}{2} & 0 & 0 & \cos\frac{\theta}{2} \end{pmatrix} \rightarrow \boxed{R_{XX}(\theta)}$$

$$R_{YY}(\theta) = e^{-i\frac{\theta}{2}Y_1 \otimes Y_2} = \begin{pmatrix} \cos\frac{\theta}{2} & 0 & 0 & i\sin\frac{\theta}{2} \\ 0 & \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} & 0 \\ 0 & -i\sin\frac{\theta}{2} & 0 & \\ i\sin\frac{\theta}{2} & 0 & 0 & \cos\frac{\theta}{2} \end{pmatrix} \rightarrow \boxed{R_{YY}(\theta)}$$

$$R_{ZZ}(\theta) = e^{-i\frac{\theta}{2}Z_1 \otimes Z_2} = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 & 0 & 0 \\ 0 & e^{i\frac{\theta}{2}} & 0 & 0 \\ 0 & 0 & e^{i\frac{\theta}{2}} & 0 \\ 0 & 0 & 0 & e^{-i\frac{\theta}{2}} \end{pmatrix} \rightarrow \boxed{R_{ZZ}(\theta)}$$

These gates are difficult to implement in hardware, so, in practice, a particular decomposition is often used:
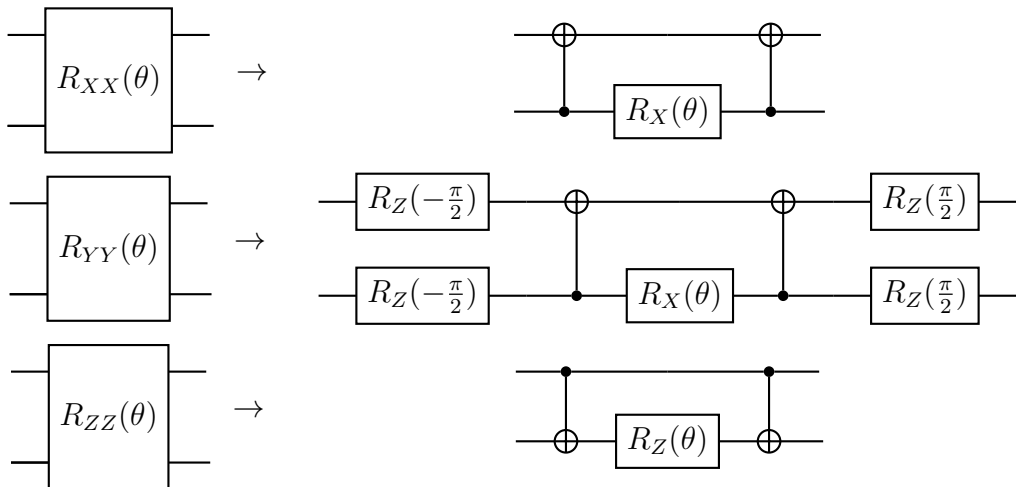
4.1.2 Measurement of Observables

In quantum computing algorithms the wave function is propagated through all the gates, effectively applying all the corresponding operators. Once the algorithm is finished, the final wave function is in a superposition of states, whose Hilbert space is defined by the number of qubits used in the algorithm. When a measurement is performed, each qubit can only return the value of 0 or 1, depending on the state that the wave function collapses on. The reading of an observable is then a binary string of the size of the number of qubits that are measured simultaneously. The probability of a given read is given by the amplitudes of the different states in the wave function. If two or more qubits are entangled, then the measurement of one will affect the state of the other. A measurement on a given qubit is usually denoted in algorithm diagrams with the symbol:

$$|q\rangle = a\,|0\rangle + b\,|1\rangle \qquad\qquad |0\rangle \text{ or } |1\rangle$$

<div align="center">quantum state   classical state</div>

As an example, if the wave function of a 2-qubit system after a series of quantum gates is in the state:

$$|\psi\rangle = \sqrt{\frac{1}{3}}\,|00\rangle - \sqrt{\frac{1}{2}}\,|01\rangle - i\sqrt{\frac{1}{6}}\,|10\rangle, \tag{4.7}$$

then a measurement on the two qubits would return 00 with probability $p(00) = 1/3$, 01 with $p(01) = 1/2$ and 10 with $p(10) = 1/6$. The output 11 would have zero probability from the above quantum state.

One important thing to consider is that measurements in quantum computing are destructive, meaning that the quantum state is lost, due to wave function collapse, after measuring. This has the practical implication that in order to perform any measurement, one needs to repeat the whole algorithm several times (or perform many "shots" in the jargon) to estimate the relative probabilities of the final states. This presents an interesting dichotomy: all of the quantum evolution, through unitary gates as the one we have presented in this section, is fully deterministic, while the measurement, a non-unitary operation, is a stochastic process.

4.1.3   Practical Considerations

The current state of quantum simulations for the field of theoretical physics is, as stated previously, in the phase of developing new tentative algorithms to solve problems. In order to test them, one has two choices: using a quantum simulator or utilizing physical hardware.

Quantum simulators are programs that translate an algorithm described in terms of qubit lines and gates to a linear algebra problem, where the size of vectors in the basis is determined by 2^N, where N is the number of qubits. The simulators offer a lot of flexibility, as one can inspect the wave function state in its entirety along the algorithm, something that we cannot do on physical hardware. Furthermore, one is free to design custom gates or operations freely. The drawback is obviously the high computational cost of the simulation and the memory limitations that effectively limit the number of qubits that one can simulate, but this should be obvious to the reader as the exponential growth in basis states is the reason to use quantum computers instead of classical ones in the first place.

The most widely used quantum simulator is probably Qiskit [22], developed and maintained by IBM. It provides a set of tools to effectively write algorithms and analyze their results. As with most simulators, it can both allow access to the wave function state or simulate measurement processes by randomly sampling the state vector given its amplitudes. Another feature that is worth mentioning is the unified framework that Qiskit offers for executing algorithms on the quantum hardware by IBM that is made publicly available [23].

While quantum simulators provide a powerful tool for testing algorithms, it is still tempting for many to try to execute them directly on physical hardware. This is, given the current state of quantum devices, purely an exercise. In practice, very little, if anything, needs to be changed in the algorithm to allow it to be executed on quantum hardware, while most systems that are being simulated are small enough to be fully analyzed either classically or with a simulator. This, however, is about to change due to the rapid increase in qubit count of machines.

Noise in Quantum Computing   The main difference between quantum simulators and real hardware is the effect of noise. Physical qubits are susceptible to external perturbations coming from the external environment. There are multiple factors that add noise to the results of an algorithm: the first one is the coherence time of the qubits themselves, i.e. the time a qubit can retain its information without random state changes due to the environment.

The second factor is the "efficiency" of the quantum gates, that is, the level of precision of the unitary transformations applied to the qubits. In particular, two-qubit gates are hard to construct, which results in them being the noisiest. A metric that is currently used to describe the length of a quantum algorithm is, in fact, the count of its CNOT gates, as they impact the total precision of the algorithm orders of magnitude more than single-qubit gates. The last source of noise in quantum devices is the measurement process, which has some level of experimental uncertainty that can lead to false reads of the state of qubits.

For context, the publicly available IBM machines of the Falcon family are quantum systems of 5 or 7 superconducting transmon qubits. Each qubit has a relaxation time of $\approx 100 \ \mu$s, while the CNOT gate error of the order of 1% and the readout error is on the level of 3%. The data is taken directly from [23].

Noise is such a significant issue in current quantum hardware that there is a strong push to find possible ways to perform "error mitigation" algorithmically, for example, through redundant encoding or CNOT extrapolations [24, 25], while the Holy Grail of the field would be self-correcting logical qubits [26, 27], which can be achieved by combining hundreds of noisy qubits together into one logical unit.

In quantum simulators like Qiskit, there are options to numerically simulate the sources of noise of the hardware, which allows for better studies of error mitigation mechanisms.

4.1.4   Structure of Quantum Simulations

In an ideal setup, once quantum computers become large enough and noise is under control, a quantum Hamiltonian simulation would be, in general, structured into three parts: state prepara-

tion, real-time evolution and measurement. One would first choose a system of interest, for which a given observable needs to be measured, but whose spectrum of states is not known beforehand. This is the case of large problems, where exact diagonalization is not feasible classically or for states coming from interactions.

The first step is then that of *state preparation*, that is, starting from qubits that are all aligned to the $|0\rangle$ state (this is the most usual convention on the hardware) and perform the right transformations to put them in the desired initial state of the system under consideration. This operation is not trivial, especially when no analytical or numerical data is available on the given state. In the case where the amplitudes of the wave function (projected onto the binary basis) are known the controlled-rotation [28], Grover-oracle [29] or the quantum-random-access-memory [30, 31] methods can be applied among others. When the amplitudes are not known, as in the case of many physics applications, and just the Hamiltonian is given, one has to rely on some dynamical algorithms for state preparation. It is important to note that a good algorithm for state preparation should just be precise, in the sense of efficiently constructing the desired state, but also short, as not to introduce noise in the rest of the simulation.

Once a state is prepared, quantum simulations can let it evolve in real-time or by applying an interaction Hamiltonian different from the one used to prepare the initial state. These calculations are incredibly challenging on classical computers as the sign-problem prevents their effective numerical simulation.

The final but crucial component of a quantum simulation is obviously the measurement of observables. This operation is also not trivial to perform in QC as measurements can only be done on single qubits, which collapse the information of the whole wave function. Some of the most widely used algorithms are the so-called quantum-classical algorithms and the Variational Quantum Eigensolver in particular [32, 33], which rely on the variational principle to construct parametrized quantum states that are optimized with classical algorithms to obtain measurements. An important topic of research in measurements is that of finding optimal groupings of qubits to measure, which depend on the level of entanglement of the system and on the observable operator

itself [34].

The focus of this work and in Sections 4.3 to 4.5 we will present three such methods applied to the case of the Schwinger Model: the Adiabatic State Preparation (ASP) algorithm, the Quantum Approximate Optimization Algorithm (QAOA) and the Rodeo Algorithm (RA). We also will only compare these methods in terms of their algorithm lengths and state preparation efficiency, but not on secondary observables computed with quantum algorithms because of the overhead they introduce.

4.2 The Schwinger Model with a θ-term

The Schwinger model is a theory of Quantum Electrodynamics (QED) in $1 + 1$ dimensions; its symmetry group is $\mathrm{U}(1)$. First analyzed by Julian Schwinger in a series of papers in the early 50s [35, 36], it was soon utilized as a toy model for understanding properties of more complicated theories, such as the relation between gauge invariance and the mass of the gauge bosons [37, 38]. Later it was utilized by Coleman to better understand confinement [39, 40], as in the case of 1 spatial dimension, the electric potential has a linear dependence on the distance between charged particles —as opposed to inverse in the case of 3 spatial dimensions. This leads to charged particles being confined to bound states, as is the case for quarks in QCD.

The massless case of the Schwinger model can be solved analytically in the case of periodic boundary conditions, which makes it a suitable tool for theoretical investigations [41, 42]. In the case of massive fermions, however, this is not the case, but some results can be obtained through mass perturbation theory [43–45]. One last peculiar feature of the Schwinger model is the topological effects given by the coupling of the non-integrable phase to the chiral condensate, which leads to the well-known θ-vacuum [41, 42]. Some work has also been done in studying the massive Schwinger model with the addition of a θ-term in the Lagrangian to understand the vacuum structure of the theory.

In the presence of a topological θ-term its Lagrangian is typically written as:

$$\mathcal{L} = -\frac{1}{4}F_{\mu\nu}F^{\mu\nu} + \frac{g\theta}{4\pi}\epsilon_{\mu\nu}F^{\mu\nu} + i\bar{\psi}\gamma^{\mu}(\partial_{\mu} + igA_{\mu})\psi - m\bar{\psi}\psi, \tag{4.8}$$

where $F_{\mu\nu} = \partial_\mu A_\nu - \partial_\nu A_\mu$ is the field tensor, the A_μ are U(1) gauge fields, and $\epsilon_{\mu\nu}$ is the totally antisymmetric tensor. In $1 + 1$ dimensions, the gamma matrices are $\gamma^0 = \sigma^z$, $\gamma^1 = i\sigma^y$, and $\gamma^5 = \gamma^0\gamma^1$. The theory has three parameters—the gauge coupling g, the fermion mass m, and the θ angle.

Following prior works [46], we perform a chiral transformation of the fields $\psi \to e^{i\frac{\theta}{2}\gamma_5}\psi$, $\bar{\psi} \to \bar{\psi}e^{i\frac{\theta}{2}\gamma_5}$ and the path integral measure [47, 48] to arrive at an equivalent Lagrangian:

$$\mathcal{L} = -\frac{1}{4}F_{\mu\nu}F^{\mu\nu} + i\bar{\psi}\gamma^\mu(\partial_\mu + igA_\mu)\psi - m\bar{\psi}e^{i\theta\gamma_5}\psi. \tag{4.9}$$

Note that we have traded the explicit θ-term and its direct relation to the gauge field at the cost of introducing a θ dependence in the fermion mass term. We choose the temporal gauge $A_0 = 0$, and then a standard Legendre transform yields the Hamiltonian:

$$H = \int dx \left[-i\bar{\psi}\gamma^1(\partial_1 + igA_1)\psi + m\bar{\psi}e^{i\theta\gamma_5}\psi + \frac{1}{2}E^2\right], \tag{4.10}$$

where in one spatial dimension, the electric field $E = F^{10} = -\dot{A}^1$ has only one component, and there is no magnetic field. More details about the Legendre transform can be found in Appendix 4A. To satisfy gauge invariance, additional local constraints that govern the interaction between matter and gauge fields must be imposed. These constraints, which are provided by the Gauss law, in the temporal gauge become: $\partial_1 E(x) = g\bar{\psi}(x)\gamma^0\psi(x)$. This equation will become important when constructing a spin Hamiltonian for use on QC.

4.2.1 Discretization on a Lattice

The continuous Hamiltonian of Equation (4.10) needs to be discretized in order to perform numerical simulations. To do this, we employ Kogut-Susskind staggered fermions [49, 50], following previous works found in [46, 51–53]. The fermion discretization that we use splits the upper $\psi_u(x)$ and lower $\psi_d(x)$ components of the Dirac fermions and places them on distinct sites of a lattice that has $2x = n$ sites, where x is the number of points in a spatial direction before the splitting, see Figure 4.2.
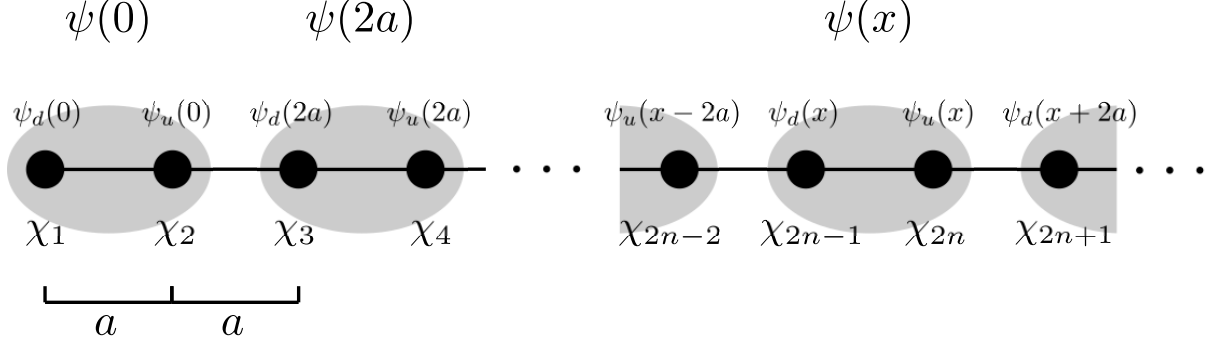
Figure 4.2: Schematic representation of the staggered discretization of the fermions at two neighboring lattice sites. A sub-lattice with half spacing introduced and the upper and lower components of the field are split.

The field $\psi(x)$ is then mapped onto a field χ_n according to:

$$\frac{\chi_n}{\sqrt{a}} \leftrightarrow \begin{cases} \psi_u(x) \text{ for } n \text{ even} \\ \\ \psi_d(x) \text{ for } n \text{ odd} \end{cases} \tag{4.11}$$

We also scale the gauge field by the coupling and relabel the gauge field $A_1(x) \to \frac{\phi_n}{ag}$ and its derivative $E(x) = \dot{A}_1(x) \to -gL_n$ to be consistent with the notation in other works. The resulting Hamiltonian is then:

$$H = -i \sum_{n=1}^{N-1} \left(\frac{1}{2a} - (-1)^n \frac{m}{2} \theta \right) \left[\chi_n^\dagger e^{i\phi_n} \chi_{n+1} - \text{h.c.} \right]$$

$$+ m \cos\theta \sum_{n=1}^{N} (-1)^n \chi_n^\dagger \chi_n + \frac{g^2 a}{2} \sum_{n=1}^{N-1} L_n^2. \tag{4.12}$$

The field ϕ_n, the electric potential, lives on the lattice site n, while the field L_n on the link between sites, just like the U field when we discretized QCD. On the lattice, Gauss's law is not trivially derived from discretizing the continuum version $\partial_1 E(x) = g\bar{\psi}(x)\gamma^0\psi(x)$, but one can define it in the strong-coupling limit by looking at the ground state of the theory, which corresponds to a "filled Dirac sea" condition. A more detailed discussion is found in [54]. The relevant result for Gauss's law is:

$$L_n - L_{n-1} = \chi_n^\dagger \chi_n - \frac{1 - (-1)^n}{2}. \tag{4.13}$$

For a given matter configuration, the gauge fields are now completely determined if one fixes open boundary conditions, i.e. fixing the derivative of the field to 0 at the boundaries. The L_n

field can, therefore, be reabsorbed in the definition of the fermion fields, as we will see later. It is important to note that this is only the case for 1 spatial dimension and this cannot be generalized to higher ones. In practice, this will enable us to only deal with fermionic degrees of freedom when constructing quantum algorithms, but if the aim is to simulate QCD on a quantum device eventually, then the problem of finding an appropriate formulation for the gauge degrees of freedom needs to be addressed. Crucially, since the gauge field is a bosonic field, representing it onto a fermionic spin-chain is a non-trivial task; this is in sharp contrast with what happens in classical simulations, where bosons are simpler than fermions to deal with.

In deriving this Hamiltonian, some care has been put into expanding terms to higher orders in a and shifting some terms by one site, utilizing the properties of staggered fermions to only have terms that either contain the field χ at one same site or at neighboring sites, more details are found in Appendix 4A. These "hopping terms", that are symmetric with respect to the two fermion species, allow us to treat the upper and lower component of the original ψ field identically. This is done because we want to map the Schwinger model onto an Ising-like spin Hamiltonian, as we will see in the next section.

4.2.2   The Spin Hamiltonian of the Schwinger Model

All Hamiltonians that need to be simulated on quantum computers eventually have to be mapped onto a spin chain, because of the fundamental structure of quantum devices. Luckily, there exists a very simple procedure to map a Hamiltonian formulated in second quantization, that is, with the creation and annihilation operators, to a spin system preserving all the necessary anti-symmetry properties: the Jordan-Wigner transformation [55] This takes the operators of a fermion field, χ_n in our case and maps them onto a chain of operators on a chain of spins, one per lattice site:

$$\chi_n = \left(\prod_{l<n} -iZ_l \right) \frac{X_n - iY_n}{2} \qquad \chi_n^\dagger = \left(\prod_{l<n} -iZ_l \right) \frac{X_n + iY_n}{2},\qquad (4.14)$$

where Pauli matrices act on the spin variables located at each lattice point n, and we used the shorthand notation $X_i = \sigma_i^x$, $Y_i = \sigma_i^y$, $Z_i = \sigma_i^z$.

If open boundary conditions are introduced, i.e. the conjugate momentum L at the boundary is fixed, one can solve Gauss's law from Equation (4.13) and obtain:

$$L_n = L_0 + \frac{1}{2} \sum_{l=1}^{n} \left(Z_l + (-1)^l \right), \tag{4.15}$$

where the value of L_0 specifies the boundary conditions. The constant L_0 is redundant, as it can be reabsorbed into the θ angle shifting it by $2\pi L_0$ [40] without changing the theory; thus, we can safely set $L_0 = 0$. The last component of the gauge field, the potential ϕ phases, can be absorbed into the fermion fields instead by a gauge transformation $\chi_n \to \prod_{l<n}[e^{-i\phi_n}]\chi_n$ without loss of generality. We then have only fermionic degrees of freedom in our Hamiltonian.

The final Hamiltonian to be implemented in the quantum algorithm can then be derived by applying the Jordan-Wigner transformation to the χ_n and $\chi_n^\dagger$ fields, see Appendix 4A for the full details, and omitting constant terms it can be decomposed as the sum of three terms $H = H_{ZZ} + H_\pm + H_Z$, where:

$$H_{ZZ} = \frac{J}{2} \sum_{n=2}^{N-1} \sum_{1 \le k < l \le n} Z_k Z_l$$

$$H_\pm = \frac{1}{2} \sum_{n=1}^{N-1} \left(w - (-1)^n \frac{m}{2} \sin\theta \right) [X_n X_{n+1} + Y_n Y_{n+1}] \tag{4.16}$$

$$H_Z = \frac{m\cos\theta}{2} \sum_{n=1}^{N} (-1)^n Z_n - \frac{J}{2} \sum_{n=1}^{N-1} (n \bmod 2) \sum_{l=1}^{n} Z_l.$$

For consistency of notation with [53], we introduced the scaled couplings $w = 1/(2a)$ and $J = g^2 a/2$. The Hamiltonian only has four free physical parameters: m, θ, w and J. By inspecting the Hamiltonian it is easy to see that the H_Z term is diagonal, while $H_\pm$ is only linking nearest-neighboring qubits. On the contrary, H_{ZZ} is an all-to-all term, i.e. it is non-local. The origin of this is the elimination of the gauge field by using Gauss's law and open boundary conditions, which removes the gauge degrees of freedom at the cost of locality. Hence, the scaling of any quantum algorithm containing this Hamiltonian is not linear with the size of the system, but quadratic.

For further reference, it is also interesting to count the number of CNOT gates that an algorithm that implements Equation (4.16) is composed of. In particular, we are interested in looking at time

evolution operators with this Hamiltonian. In order to do this one must consider the operator $U(t, t + \delta t) = \exp(-iH\delta t)$, with δt being some time difference. Since not all terms in the Hamiltonian commute with each other, it cannot be simply exponentiated. In fact, one has to use the formula by Sophus Lie for the exponential of a sum of matrices:

$$\exp[it(A + B)] = \lim_{n \to \infty} \left[\exp(itA/n) \exp(itB/n)\right]^n, \tag{4.17}$$

this is also more commonly known as the Trotter formula. In practice, one cannot calculate all terms of the equation, so one has to truncate the product under the assumption that t, or δt in our case, is small. This leads to the so-called Trotter-Suzuki formulas for the expansion of the exponentials [56]. For the simulation of the Schwinger Hamiltonian we split it into three terms: H_X, H_Y and H_Z, where each contains only the corresponding operators; this is just a regrouping of the terms of Equation (4.16). With these three terms, which all commute with each other, one can use different orders of the Trotter-Suzuki formula to approximate the time-evolution operator. In particular, we used:

$$U(t, t + \delta t) = e^{-iH(t)\delta t} \approx e^{-iH_X(t)\delta t} e^{-iH_Y(t)\delta t} e^{-iH_Z(t)\delta t} + \mathcal{O}(\delta t^2), \tag{4.18}$$

$$U(t, t + \delta t) = e^{-iH(t)\delta t} \approx e^{-iH_X(t)\frac{\delta t}{2}} e^{-iH_Y(t)\frac{\delta t}{2}} e^{-iH_Z(t)\delta t} e^{-iH_Y(t)\frac{\delta t}{2}} e^{-iH_X(t)\frac{\delta t}{2}} + \mathcal{O}(\delta t^3),$$

for the first- and second-order formulas. As in regular numerical integration, the total error, when simulating from $t = 0$ to T is of one power of δt more, so $\mathcal{O}(\delta t)$ and $\mathcal{O}(\delta t^2)$ respectively. With these approximations, we can now see immediately that implementing the time evolution operator is trivial with the Ising gates defined in Section 4.1.1 and we can thus conclude that for the first-order Trotter formula, the number of CNOT gates needed is $4(N - 1) + (N - 1)(N - 2)$, where the first term comes from the H_X and H_Y terms in the Hamiltonian, while the latter part comes from H_Z. For the second-order Trotter-Suzuki approximation, since we have placed the H_Z term in the middle, the count of CNOT gates is simply: $8(N - 1) + (N - 1)(N - 2)$. In the simple cases of $N = 4, 8$, which we will consider in the following sections, the number of gates are then:

N	1st Order	2nd Order
4	18	30
8	70	98

4.3 Adiabatic State Preparation

The Adiabatic State Preparation (ASP) is a common reference when dealing with state preparation algorithms, because of its relative simplicity and robustness. The algorithm allows initializing a set of qubits to the ground state of a chosen "target" Hamiltonian H, without the need of any additional qubit. What is needed instead is to find a "simple" initial Hamiltonian H_0 with a ground state that is trivially solvable and easy to construct on a quantum computer. H_0 should be related to the full target one in such a way that a set of coefficients can be used to interpolate between them.

The Adiabatic Theorem is well-known fact in quantum mechanics, with its roots in the early times of the theory [57]. It states that for a system governed by a time-dependent Hamiltonian $H(t)$ a state prepared in one of its eigenstates at $t = 0$, so that $|\psi(0)\rangle = |\epsilon_i(0)\rangle$ will approximately remain in the corresponding instantaneous eigenstate $|\psi(t)\rangle = |\epsilon_i(t)\rangle$ of the Hamiltonian provided that $H(t)$ varies "sufficiently slowly". In the case of the evolution to a finite time T, the adiabatic theorem becomes approximate and there an *adiabatic condition* has been formulated [58, 59] that needs to be satisfied for the final state to be in an eigenstate of the Hamiltonian. This condition, after introducing the scaled time $s = t/T$, reads:

$$\frac{1}{T} \max_{s \in [0,1]} \frac{|\langle \epsilon_i(s)| \, \partial_s H(s) \, |\epsilon_j(s)\rangle|}{|\epsilon_i(s) - \epsilon_j(s)|^2} \ll 1 \tag{4.19}$$

The adiabatic condition then is satisfied if there is a sufficiently large energy gap between the state of interest and the neighboring ones and if the total time evolution is large. The derivative term is assumed to be small for a well-behaved Hamiltonian [59], so we will not consider it in this analysis. In the case of the ground state (GS), if we call $\Delta(s) = \epsilon_1(s) - \epsilon_0(s)$ the difference between the first excited state energy and the GS energy, the adiabatic condition is, for a well-behaved Hamiltonian,

roughly:

$$\frac{1}{T\Delta^2(s)} \ll 1 \tag{4.20}$$

ASP works by applying a set of adiabatic time evolution operators, each with a different Hamiltonian, which comes from the discretization of an interpolation between H_0 and H, so that the instantaneous operator is $H(s) = a(s)H_0 + b(s)H$ where the time-dependent coefficients $a(s)$ and $b(s)$ are monotonically decreasing and increasing respectively. This allows the state to evolve towards the desired target ground state by always remaining close to the ground state of the interpolation of the Hamiltonians at all time steps. More rigorous versions of this condition have been formulated, with more stringent bounds depending on the properties of the Hamiltonian [60, 61].

In the case of the Schwinger model, we choose the initial Hamiltonian:

$$H_0 = H_{ZZ} + H_Z|_{m\to m_0, \theta\to 0}, \tag{4.21}$$

whose ground state is a state of alternating up and down spins. The parameters of the Hamiltonian, $w = \frac{1}{2a}$, θ and m, are rewritten to be adiabatic time-dependent, while the coupling parameter $J = g^2 a/2$ is kept constant:

$$w \to \frac{t_n}{T}w \quad \theta \to \frac{t_n}{T}\theta \quad m \to \left(1 - \frac{t_n}{T}\right)m_0 + \frac{t_n}{T}m, \tag{4.22}$$

for a set of $M+1$ values of $t_n = t_{n-1} + \delta t_n$ with, naturally, $t_0 = 0$. The set of time intervals $\delta t_1, \delta t_2 \ldots \delta t_N$ has to satisfy: $T = \sum_{n=1}^{M} \delta t_n$. The simplest discretization is a linear one, corresponding to taking all δt_n to be the same and hence $t_n = n\frac{T}{M} = n\delta t$, but other non-uniform values for δt_n can be chosen as we will discuss later.

To perform the state preparation one first defines an adiabatic time evolution operator $U(t, t + \delta t) = e^{-iH(t)\delta t}$. We label the linear discretization of the adiabatic evolution $L1$ or $L2$ depending on the order of the Trotter product formula, see Section 4.2.2, we use to evaluate the evolution $U(t) = e^{-iH_A(t)\delta t}$, which intuitively correspond to the order of the integration scheme from t to $t + \delta t$. The other discretizations used in this work are $\sin^2$ or $\cos^2$ spacing for the time:

$$\delta t_n = 2\frac{T}{M}\sin^2\left(\pi\frac{n}{M}\right) \quad n = 0, \ldots, M, \tag{4.23}$$

$$\delta t_n = 2\frac{T}{M}\cos^2\left(\pi\frac{n}{2M}\right) \quad n = 0, \ldots, M, \tag{4.24}$$

which we call respectively $S1$, $S2$ for the sine and $C1$, $C2$ for the cosine, where again the number corresponds to whether for the adiabatic time evolution operator we use a first order or a second order Trotter product formula.

4.3.1 Numerical Results

For clarity of exposition, since there are several free parameters in the model, we restricted our analysis to the following combinations: $(m, m_0, w, J, \theta) = (0, 0.5, 0.5, 0.5, 0)$ and $(1, 0.5, 0.5, 0.5, \pi/4)$. This is the massless case, but starting from a non-zero mass in the simple Hamiltonian, unit lattice spacing and coupling and vanishing θ and the other is a massive case with a non-zero topological angle. We also considered two sizes of the system: $N = 4$ and $N = 8$.

The first numerical test that we considered to understand if the ASP could be used at all, was to compute the spectrum of the Hamiltonian as a function of the scaled time $s = t/T$. In Figure 4.3 we plot the spectrum and in Table 4.1 we report the minimum values for $\Delta(s)$. The spectrum is calculated by numerically diagonalizing the interpolating Hamiltonian for every value of s. Curiously, for the analysis of the algorithm, the massless case with $\theta = 0$ is more interesting because of the smaller gap along the adiabatic evolution.
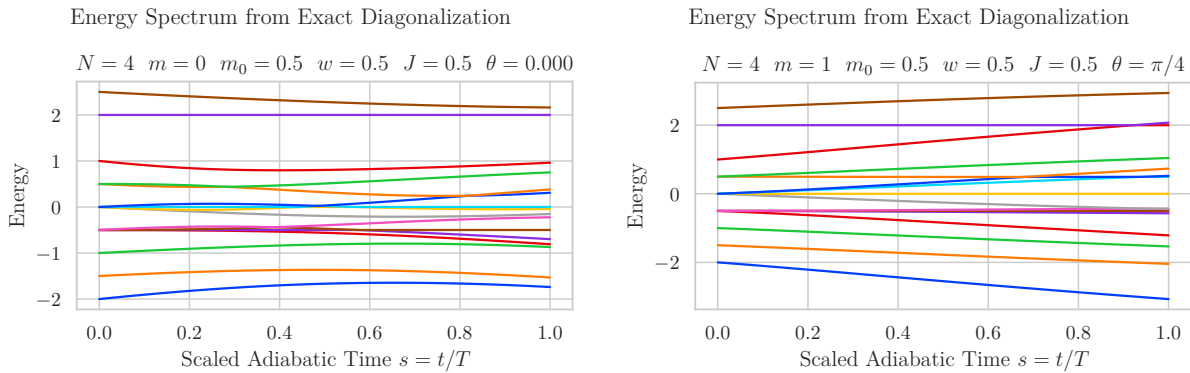
Figure 4.3: Spectrum of the Schwinger model for $N = 4$ with $(\theta, m) = (0, 0)$ (left) and $(\theta, m) = (\pi/4, 1)$ (right).

N	$(\theta, m) = (0, 0)$	$(\theta, m) = (\pi/4, 1)$
4	0.208	0.5
8	0.185	0.5

Table 4.1: Values for the minimum energy difference between the first excited state and the ground state $\min_{s \in [0,1]} \Delta(s) = \min_{s \in [0,1]} (\epsilon_1(s) - \epsilon_0(s))$.

On a classical computer, we can simulate the quantum algorithm for the ASP and perform a study of how the choice for the discretization and the total time of evolution T affect the quality of the ground state that is prepared. We focus in particular on two quantities as metrics for our algorithms: the energy expectation value, and its relative error to the exact result, and on the overlap $S = |\langle \psi | \psi_E \rangle|^2$ of the final state with the exact ground state $|\psi_E\rangle$ computed from diagonalization.

There are then four parameters that can affect the quality of the end state from ASP for a given set of physical parameters: the total time of simulation T, which controls the quality of the adiabatic approximation; the number of discretization point M, which affects the size of the error made in the approximation of the time-evolution operator; the type of discretization used, we considered linear, sine or cosine; and the order of the Trotter formula.

For quantum simulations, the choice of the discretization does not impact the final length, in terms of gates, of the algorithm, so it can be set freely. On the contrary, a larger value of T would necessitate a large number of M as well to preserve the error in the Trotter formula small, as it depends on δt, making the algorithm longer. The choice of the Trotter order also affects the length of the algorithm as it requires twice as many gates to perform a single time evolution step.

In Figure 4.4, we report the results of the relative error of the GS energy for varying values of T for the linear discretization for different values of M and δt and Trotter order.

One important thing to notice is that the error, for large very M/T, i.e. for small δt has a plateau after 10 irrespective of the order in the Trotter formula for discretization. The error in that regime is, in fact, given by the adiabatic approximation and not by the discretization. We can also understand it by the fact that the plateaus have lower values for increasing T. From the
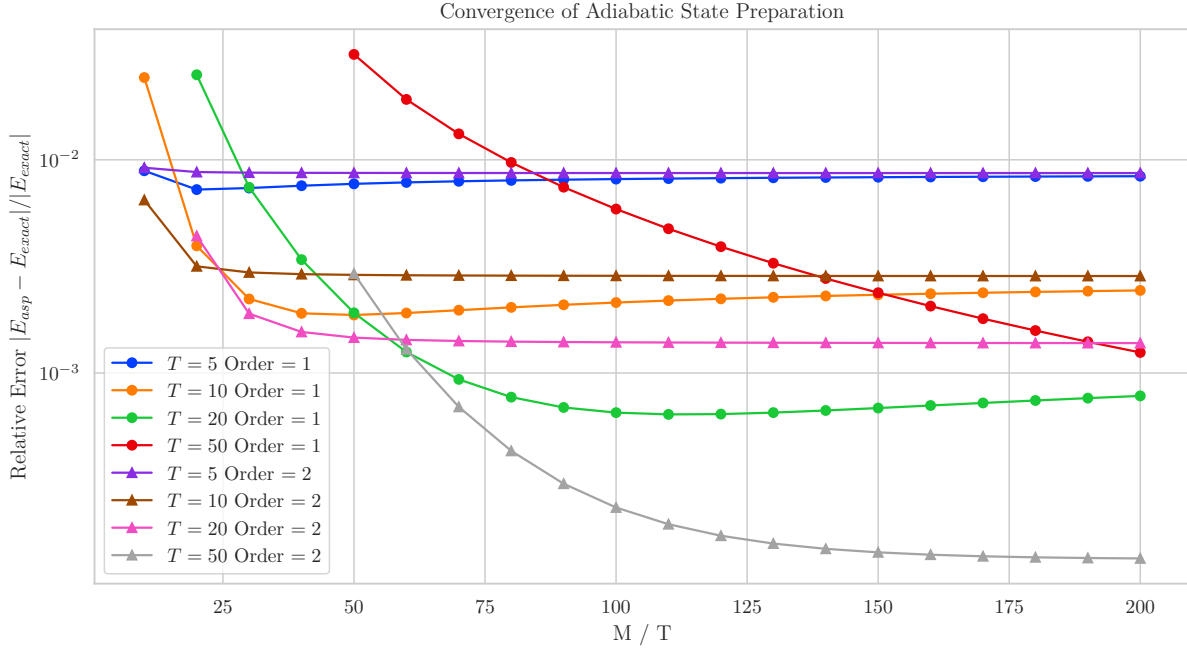
Figure 4.4: Relative error of a quantum simulation of ASP for a $N = 4$ system with $J = 0.5$, $m_0 = 0.5$, $m = 0$, $w = 0.5$, $\theta = 0$. The different points correspond to different parameters of the algorithm, in particular the order of the Trotter formula, the total time of simulation T and the number of steps M.

figure, one can also infer that with most combinations of parameters, the error on the GS energy is always very small, showing the effectiveness of ASP as a state preparation algorithm for this particular system.

From the plot of Figure 4.4 a curious phenomenon appears when looking at the behavior of the relative error as the order of the Trotter formula is changed. Clearly, for larger δt higher order give better results, but then as δt decreases, it appears that the first order approximation is better. However, from the top figure, we can see that for very small δt, the two discretizations eventually converge. It appears then that the "better" converge to the solution of the first order Trotter results for intermediate values of δt is somewhat incidental.

As an extreme case, we show in Figure 4.5 the results for the energy and the overlap with the ground state for the case of just $M = 10$ and $T = 5$. With these parameters, the adiabatic approximation clearly cannot hold, yet the ASP is still able to prepare a GS that has more than

95% overlap with the true ground state and for certain discretizations, the difference is smaller that 1%. It should be noted even in this scenario the resulting algorithm is very long in terms of what can be simulated on current quantum hardware. By the data of Section 4.2.2, we can infer that even in the $N = 4$ first-order Trotter case, the ASP algorithm is 180 CNOT gates long with this Hamiltonian.
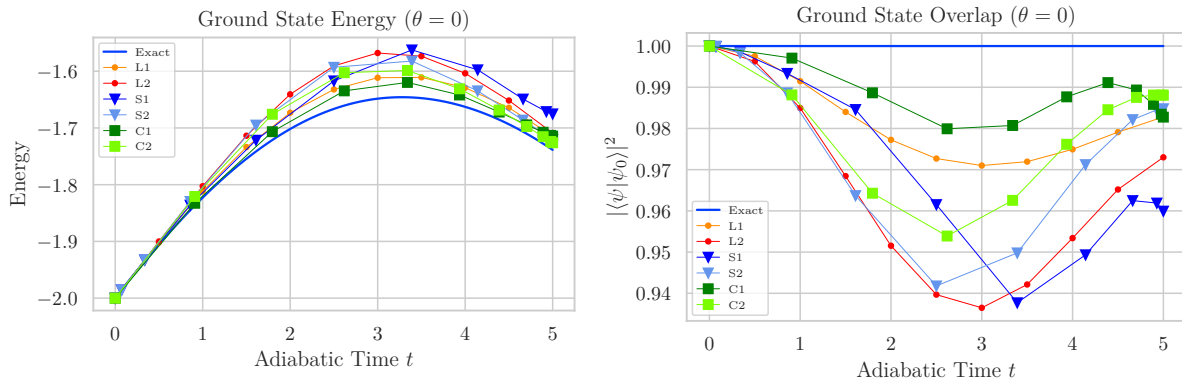
Figure 4.5: Simulation of ASP of a $N = 4$ system with $J = 0.5$, $m_0 = 0.5$, $m = 0$, $w = 0.5$, $\theta = 0$. The different series represent different adiabatic time discretizations and with different Trotterization scheme orders. The simulation is done using the Qiskit package from IBM. The data is for the energy of the system (left) and for the final state overlap with the true ground state (right). The exact line represents the results obtained from the exact diagonalization of the adiabatic Hamiltonian at every time point.

One interesting consideration that one can draw is that in the cases of cosine spacing, the data seems to be in better agreement, hinting that it is beneficial to perform more precise steps towards the end of the evolution, where the Hamiltonian is closer to the target one. Also, it is important to notice that the overlap is always within 10% from the exact ground state, meaning that the evolution is, in fact, keeping the state close to the intermediate ground state throughout the process.

4.3.2 Noise Model for Adiabatic State Preparation

One of the largest limitations of ASP is the long algorithm resulting from the adiabatic time evolution discretization. On NISQ machines, such algorithms easily become prohibitively expensive

because they contain several CNOT gates which have relatively high error rates. Using the tools provided by Qiskit, we performed a set of simulations in which the error rate of the CNOT gates was changed using the simple setup that we described in the previous section, i.e. $M = 10$ and $T = 5$, with the best discretization we found: $C1$. The intent is to determine what CNOT gate error rate would allow the ASP algorithm to give a good estimate of the ground state of the $N = 4$ Schwinger model. In fact, there is a big discrepancy between the noiseless and the results for error rates of 10^{-3} already after one single step of the adiabatic time evolution. Qiskit allows the execution of a quantum algorithm with a custom "noise model", which means that at every step of the algorithm a parametrized random error is introduced to simulate the effect of the hardware noise. In this particular case, we changed the value of the 2-qubit error parameter. Remembering that the current NISQ hardware has error rates that are in the range $10^{-3} - 10^{-2}$, it is easy to check from Figure 4.6 that the noise prevents the algorithm from producing a good quality state on a real machine in the already fragile setup we are considering.

A common technique used in quantum simulations is zero-noise extrapolation [24, 25], which consists in artificially increasing the length of the algorithm by introducing pairs of CNOT gates. These pairs leave the result unaffected in principle, the square of the CNOT operator is the identity, but only increase the noise coming from the two-qubit gate error rate. By repeating the procedure with an increasing number of CNOT pairs, one can then perform an extrapolation to remove all the contributions from the two-qubit gates. However, from Figure 4.7, one can observe that an extrapolation performed this way would lead to wrong results. In the figure, we observe that the two-qubit gate error rate affects the results for the energy at low precision, but for very large values, and crucially in the region of current hardware levels, the energy tends to 0, i.e. the state is pure noise.

From this analysis, we can conclude that ASP is not yet a practical algorithm for state preparation for the Schwinger model on real quantum hardware due to the constraints set by the noise of the machines. Even in the case of very short time evolutions, in which the error coming from the adiabatic approximation is small but not negligible, like the one we presented for $T = 5$, current
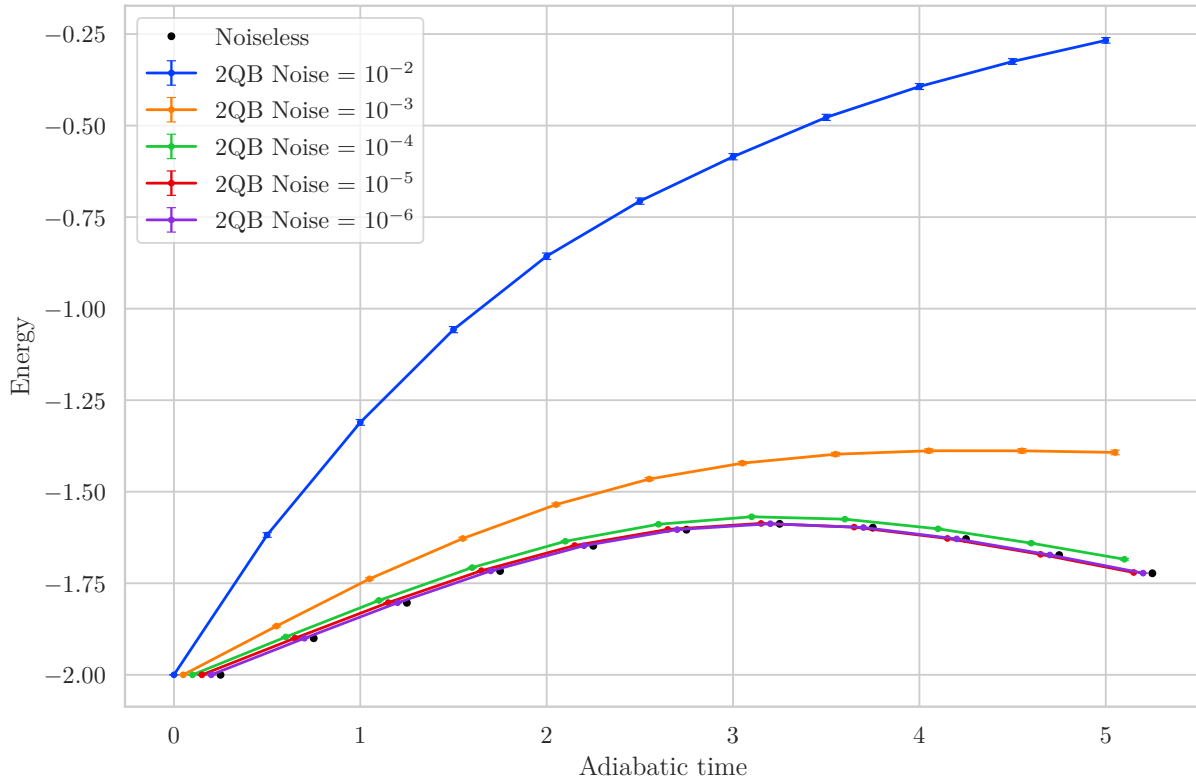
Figure 4.6: Ground state energy from ASP with varying CNOT error rate in the same setup as Figure 4.5 for linear first order Totter. Error bars represent the standard error coming from performing 10000 shots of the algorithm. The points have been shifted from the true t coordinate for clarity.

noise levels render ASP completely unfeasible. From our results, it appears that at least one order of magnitude decrease in two-qubit error rates, down to below 10^{-4}, is needed in order for ASP to become effective.

4.4 Quantum Approximate Optimization Algorithm

The Quantum Approximate Optimization Algorithm (QAOA) [62] is a well-known algorithm for optimization problems. However, using the variational principle, one can turn the QAOA into a hybrid classical-quantum algorithm for state preparation. It has the advantage of producing relatively short algorithms with a small number of parameters.

Just as ASP, it relies on the existence of a trivially solvable Hamiltonian H_0 with eigenstate
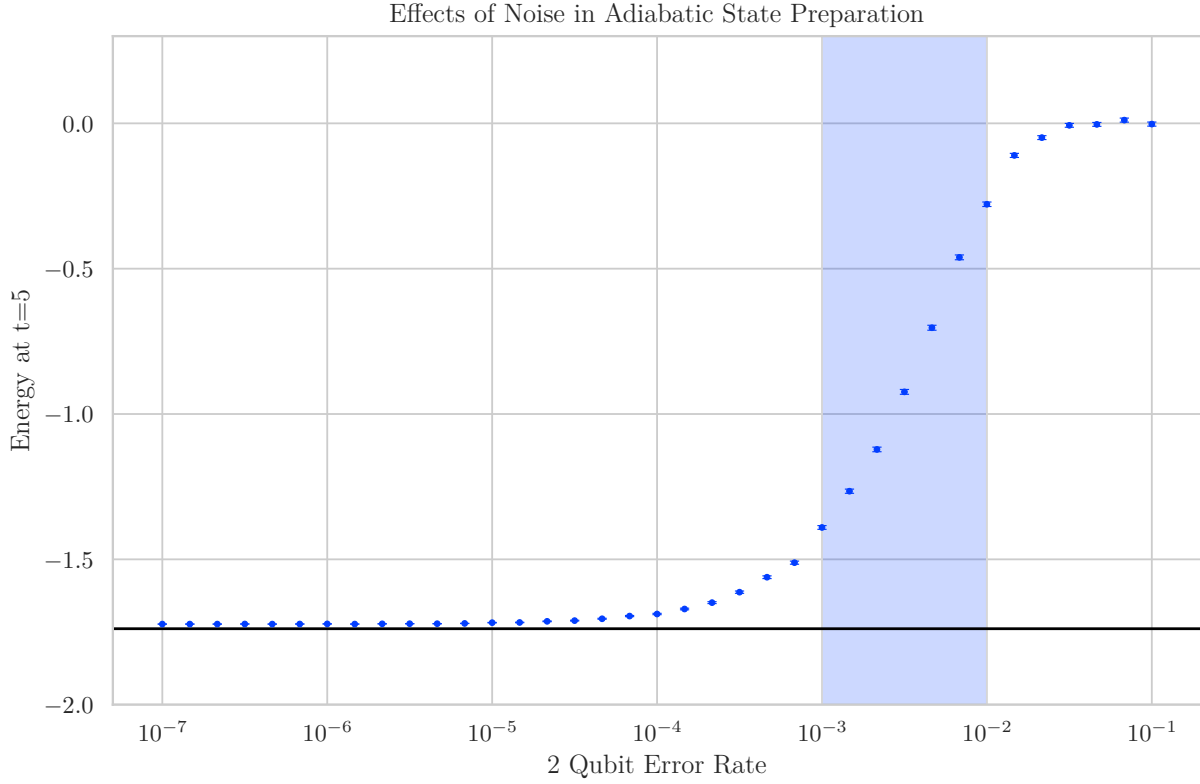
Figure 4.7: Ground state energy from ASP with varying CNOT error rate in the same setup as Figure 4.5 at the final time point $t = 5$. The shaded area represents the current typical error-rates.

$|\psi_0\rangle$ from which, with a set of unitary transformations, and approximation of the ground state of H, the target Hamiltonian. One then makes the QAOA ansatz for the eigenstate of the target Hamiltonian:

$$|\psi_M(\vec{\gamma}, \vec{\beta})\rangle = \left(\prod_{k=0}^{M-1} e^{-i\beta_{M-k}H_0} e^{-i\gamma_{M-k}H} \right) |\psi_0\rangle \tag{4.25}$$

which depends on the $2M$ real coefficients $\vec{\beta}, \vec{\gamma}$. The variational principle applied to this ansatz reads:

$$\langle \psi_M(\vec{\gamma}, \vec{\beta})| \, H \, |\psi_M(\vec{\gamma}, \vec{\beta})\rangle = E_0^V \geq E_0, \tag{4.26}$$

where E_0 is the true ground state of the system, which means that we can use a minimization algorithm to find the optimal values $\vec{\gamma}^*$ and $\vec{\beta}^*$ such that $|\psi_M(\vec{\gamma}^*, \vec{\beta}^*)\rangle$ minimizes the energy E_0^V. In our case, we used simulated annealing [63] as a minimizer because it can deal with the many parameters and multiple local minima of the problem, thanks to the stochastic nature of the

algorithm. The minimization we performed was a fully classical computation, but in principle, one can use a hybrid algorithm, which consists in calculating the expectation values on a quantum device with the parameters at every step in the minimization algorithm determined on a classical computer based on the results of the quantum simulation. However, given the high noise levels of current QC this is not possible in practice. So the minimization was setup to use a trotterized version of the unitary transformations to more closely represent the quantum hardware.

One of the convenient advantages of the QAOA method for state preparation is that one is free to choose the number of steps M in the ansatz without any other limitation other than the successful convergence of the minimizer. This clearly leads to short quantum algorithms. There is one caveat, though, that is if the magnitude of one of the coefficients $\vec{\gamma}$ or $\vec{\beta}$ is large, and since the time evolution operator needs to be Trotterized, there could be an error introduced as opposed to the optimal case. For this reason, while minimizing, one can add the further constraint of requiring all coefficients to be below a certain maximum value. If this cannot be satisfied by the minimizer, then a larger number of steps M is required.

4.4.1 QAOA Results

The quality of the state prepared by QAOA depends on the optimal coefficients that are found. In the case of the Schwinger model, with the same parameters we used in Section 4.3, so $(m, m_0, w, J, \theta) = (0, 0.5, 0.5, 0.5, 0)$ and $(1, 0.5, 0.5, 0.5, \pi/4)$, the results for the energy expectation value of the ground state and the ground state overlap are reported in Table 4.2 for two values of the number of QAOA steps. All results are for second order Trotter formula.

From the table, we can observe that QAOA is extremely efficient at preparing the ground state and at estimating its energy. It also does it with extremely few CNOT gates when compared to ASP. Consider that for the $M = 10$ case of ASP, the number of CNOT gates per qubit, with the second order approximation, is 75 for $N = 4$ and 245 for $N = 8$, with also less accurate results. The downside of QAOA is the minimization process. This can be done accurately on a classical machine running a simulator when the system sizes are small enough to allow exact

161

Method	N	(θ, m)	M	CNOT/qubit	Rel. Err. E_0	GS Overlap
QAOA	4	$(0, 0)$	2	12	0.0029	0.9975
QAOA	4	$(0, 0)$	3	18	0.0041	0.9968
QAOA	4	$(\pi/4, 1)$	2	12	0.00045	0.9996
QAOA	4	$(\pi/4, 1)$	3	18	0.0031	0.9971
QAOA	8	$(0, 0)$	2	28	0.0089	0.9701
QAOA	8	$(0, 0)$	3	42	0.0068	0.9846
QAOA	8	$(\pi/4, 1)$	2	28	0.00047	0.9988
QAOA	8	$(\pi/4, 1)$	3	42	0.00040	0.9989

Table 4.2: Results for the QAOA method for varying system sizes and parameters with first order Trotter formula.

exponentiation, and even then, some combination of parameters such as $(\theta, m) = (0, 0)$ in the $N = 8$ case are not highly optimized. There is, however, no guarantee that this process could be done effectively in the hybrid computation scenario with NISQ machines. Furthermore, in our results, because of the stochasticity of simulated annealing different reruns of the minimization algorithm led to slightly different results, so the values reported in table are the averages of 5 runs of the optimizer. The difference between the runs, however, does not change the qualitative conclusions that we can draw when comparing QAOA with ASP, as they are orders of magnitude better for comparable algorithm lengths. To summarize, the results clearly show that once the optimal coefficients $\vec{\gamma}^*, \vec{\beta}^*$ are known, QAOA can prepare the GS more effectively than ASP.

4.4.2 Blocked QAOA

One option to decrease the number of CNOT gates per qubit even further would be to use custom optimized 2-qubit gates [64], which can be constructed for any unitary operation on two qubits. However, since the Hamiltonian in Equation (4.16) is non-local due to the H_{ZZ} term, this cannot be done trivially. The QAOA algorithm, nevertheless, since it relies on a rather general ansatz, can still be applied to this case. The solution we propose to deal with the non-local term is to define a modified Hamiltonian H_B, which we will call "blocked", where only the diagonal and nearest-neighbor terms of the full Hamiltonian are kept, so $H_B = H_\pm + H_Z + H'_{ZZ}$, and that can in principle be represented as a single 2-qubit gate. H'_{ZZ} is only the local part of H_{ZZ}. We then

modify the QAOA ansatz to:

$$|\psi_M(\vec{\gamma}, \vec{\beta})\rangle = e^{-i\beta_M H_0} e^{-i\gamma_M H} \left(\prod_{k=1}^{M-1} e^{-i\beta_{M-k} H_0} e^{-i\gamma_{M-k} H_B} \right) |\psi_0\rangle \tag{4.27}$$

where the first $M-1$ unitary application contain the H_B while only one application of the full Hamiltonian is applied on the last step. The rationale is then to encode as much as possible of the nearest-neighbor interactions through the blocked Hamiltonian, while the last step "corrects" for non-local effects. This can only be applicable in the parameter region where H_{ZZ}, which contains the non-local terms, is not the dominant term of the Hamiltonian, so we are consequently constrained to the case where J is not large. The results for this blocked approach, for the $N=4$ case, can be found in Table 4.3; the results for $N=8$ are qualitatively similar, so are not reported.

Method	N	(θ, m)	# of Steps	# of CNOT/ qubit	Rel. Err. E_0	GS Overlap
QAOA	4	$(0,0)$	2	11	0.0073	0.9906
QAOA	4	$(0,0)$	3	16	0.0043	0.9960
QAOA	4	$(\pi/4, 1)$	2	11	0.00022	0.99973
QAOA	4	$(\pi/4, 1)$	3	16	0.00026	0.99969

Table 4.3: Blocked QAOA results for $N=4$.

The gain in terms of CNOT gates is more apparent for larger N, obviously. As a side consequence of the blocking procedure is the possibility of scaling the system size while keeping the optimal parameters $\vec{\gamma}$ or $\vec{\beta}$ fixed as N changes. This is not perfect, as the non-local part of the Hamiltonian is obviously changing, but it leads to surprisingly good ansätze for larger systems without expensive minimization runs. In Table 4.4, we show results for different system sizes with the same QAOA parameters determined with simulated annealing on the $N=4$ system for the $(\theta, m) = (0,0)$ case and three steps, two of which are blocked and the last one is done with the full Hamiltonian.

As the data shows, blocked QAOA can be used as an extremely cheap starting point for further optimizations or other state preparation algorithms in the case of large N.

N	# CNOT/qubit	Rel. Err. E_0	GS Overlap
4*	16	0.0043	0.9960
6	22	0.0083	0.9820
8	27	0.0168	0.9387
10	32	0.0231	0.8795

Table 4.4: QAOA Blocked with $M = 3$ steps. The results for the $N = 4$ are obtained after parameter optimization; the results for $N = 6, 8, 10$ have been computed using the same optimal parameters for $N = 4$.

4.5 The Rodeo Algorithm

The recently proposed Rodeo Algorithm (RA) [65–67] is a promising method that uses a stochastic cosine filter to isolate eigenstates of a given Hamiltonian, so it can also be used as a state preparation algorithm. The method relies on a series of controlled time evolution operators acting on a series of *ancilla qubits*. The final state of the ancilla qubits, when measured, can be used to assess the overlap of the initial state with an energy eigenstate. A schematic representation is given in Figure 4.8.
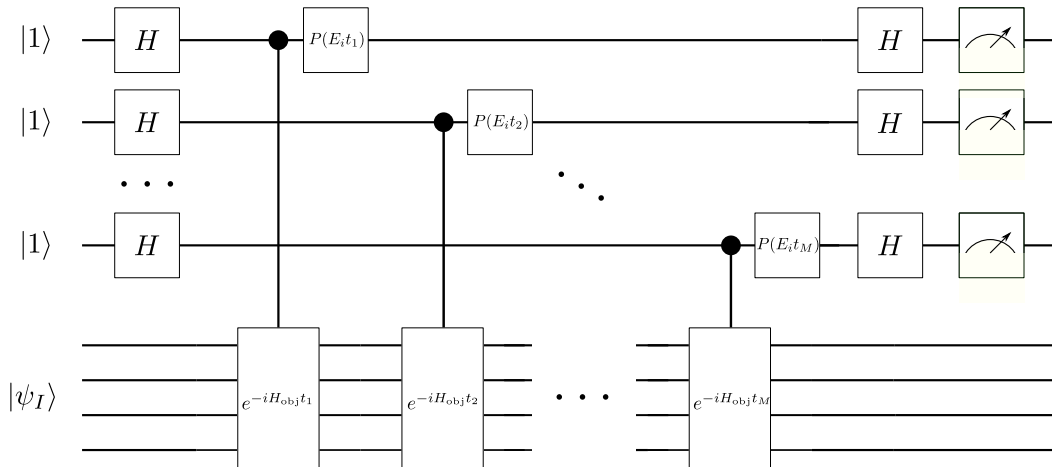

Figure 4.8: Circuit diagram for the Rodeo Algorithm.

One of the simplest applications of the RA is that of finding the overlap of a given state $|\psi_I\rangle$ with the eigenvalues of a given object Hamiltonian H_{obj}. To do this, one first selects a set of energies $E_i \in [E_{\min}, E_{\max}]$ and for each energy one run of the algorithm is performed, thus

performing as "scan" for the energy values. At the start of each run, for a fixed energy E_i, an initial state of N qubits $|\psi_I\rangle$ is prepared, this has to be a trivial operation, and another set of M ancilla qubits are all initialized to the fully mixed state with a Hadamard gate. The algorithm then consists of performing a controlled time evolution with the desired object Hamiltonian H_{obj} for a time t using one ancilla qubit and then applying a phase gate P with angle $E_i t$, a Hadamard and measuring the qubit. This chain of operators is applied to each ancilla qubit.

The action of the algorithm, for a single ancilla qubit, can be written in matrix form as:

$$
\begin{pmatrix} \frac{I}{\sqrt{2}} & \frac{I}{\sqrt{2}} \\ \frac{I}{\sqrt{2}} & \frac{-I}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & e^{iE_i t} \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & I e^{-iH_{\text{obj}} t} \end{pmatrix} \begin{pmatrix} \frac{I}{\sqrt{2}} & \frac{I}{\sqrt{2}} \\ \frac{I}{\sqrt{2}} & \frac{-I}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} 0 \\ |\psi_I\rangle \end{pmatrix}
\tag{4.28}
$$

from which the probability of an ancilla qubit to be in the state $|1\rangle$ for each eigenvalue ϵ_j of the Hamiltonian can be calculated to be:

$$
\cos^2\left[(\epsilon_j - E_i)\frac{t}{2} \right] = \left| \frac{1}{2} + \frac{1}{2} e^{-it(\epsilon_j - E_i)} \right|
\tag{4.29}
$$

The probability for a set of M ancilla qubits, on which the same operations of Equation (4.28) are applied each with a different random value for the time $t_1, t_2, \ldots, t_M$ is:

$$
P_M = \prod_{m=1}^{M} \cos^2\left[(\epsilon_j - E_i)\frac{t}{2} \right]
\tag{4.30}
$$

which is a cosine filter which suppresses values of $E_i \neq \epsilon_j$ by a factor of $\frac{1}{4^M}$. The randomness of the choice of t is introduced to cancel potential constructive interference away from ϵ_j.

In practice, one performs a series of iterations of the rodeo algorithm, i.e. choosing M random values for the time and measuring the probability of all the ancilla qubits being $|1\rangle$, by drawing S samples of M Gaussianly distributed values for t with some fixed root-mean-squared value, which effectively controls the sharpness of the filtering. This is then repeated for all values of E_i.

In Figure 4.9, we show the results for the Schwinger model with $N = 4, 8$ qubits for a set of numbers of ancilla qubits, or "cycles" of the RA algorithm. The results are for 200 values of the energy E_i with $S = 20$ random time samples per value. The exact results come from diagonalization and the initial state is the alternating spin chain that was also used in the previous

methods. In the largest case, the simulation contains $M + N = 17$ qubits, but this is not possible to run easily using Qiskit on a regular machine due to memory limitations. The choices were then to either perform the calculation on an HPC center or, the path we chose, to write a custom, less general but more optimized simulator for the rodeo algorithm. This custom simulator can be used with any Hamiltonian and can easily simulate up to 20 qubits, considering the sum of both ancilla and system ones.

The results of Figure 4.9 clearly show the effectiveness of the RA in identifying the eigenvalues of H_{obj} and the overlap of $|\psi_I\rangle$ with them. One important consideration is that as M increases, the value of the overlap factors approaches their true value strictly from above. This is due to the constant $1/2^M$ baseline coming from Equation (4.30), which we can also see decreasing constant noise level on the plot for energies far away from the eigenvalues. The baseline comes from the random chance of having all the ancilla qubits being in the $|1\rangle$ state, with no relation to the RA at all. The width of the peaks is, as stated before, controlled by the random distribution of the values of t, so there is a trade-off between the resolution and the accuracy of determination of the central value.

It should be noted that the controlled time evolution is not a trivial operation on a quantum computer because it requires substituting all CNOT gates in the regular time evolution to the so-called Toffoli gates [68]. These are the extension of the CNOT gates to the three-qubit case (or more if made more general) and it is sometimes called the CCNOT gate. On most hardware there is no method to apply three-qubit operations, so to get a realistic estimate of the length of the algorithm one needs to substitute all the Toffoli gates with their decomposition into Hadamard, phase gates and CNOT gates. In particular, one three-qubit Toffoli gate can be decomposed using 5 CNOTs [69]. To circumvent the problem of dealing with three-qubit gates in the RA it has been proposed in [66] to modify the algorithm with the use of a pairs of controlled reversal gates, one for each commuting piece of the Hamiltonian. This method can greatly reduce the length of the RA.

Another thing to consider when analyzing the length of the RA is the need to use a Trotter
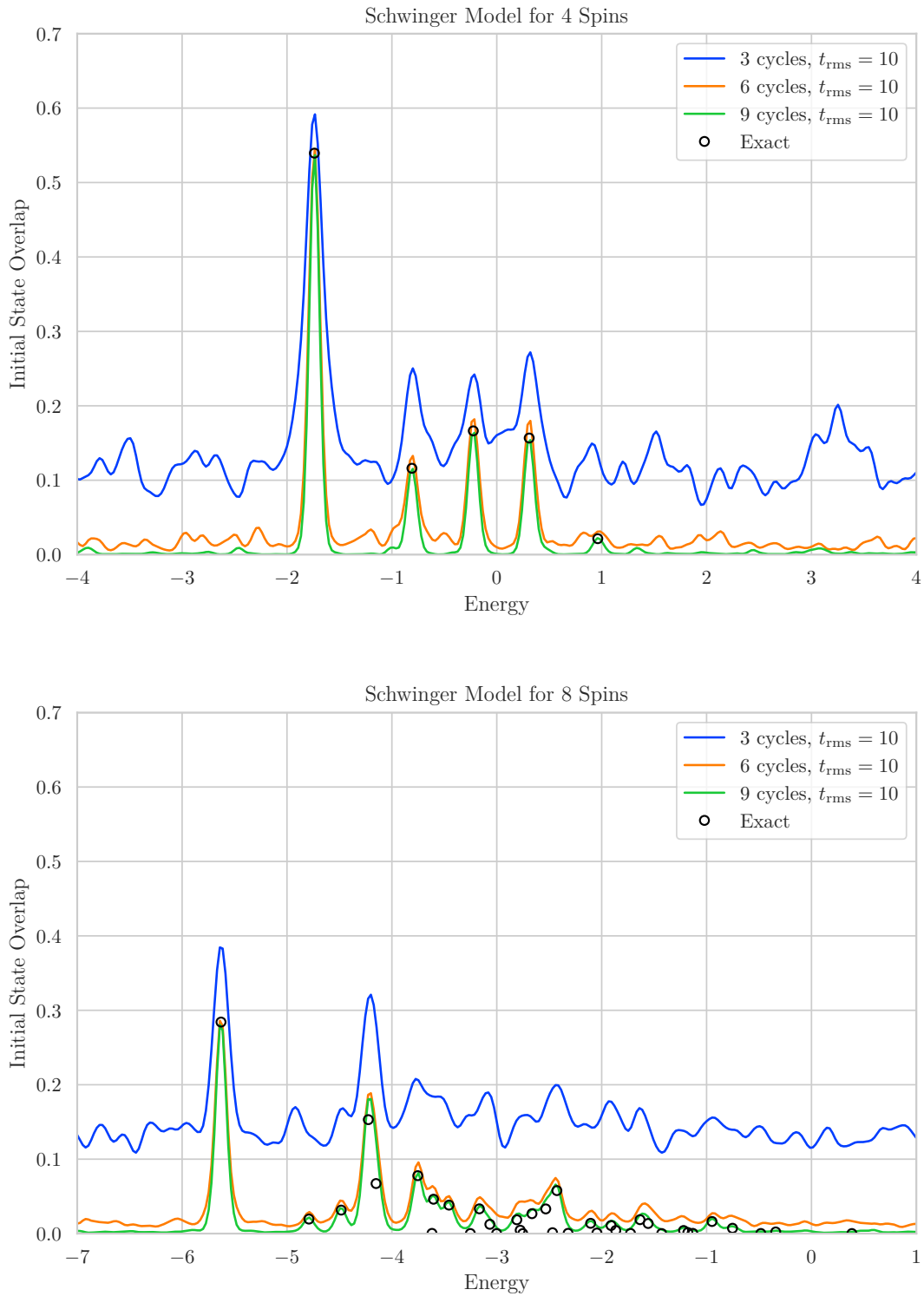
Figure 4.9: Spectral overlap factors for the $|1010\rangle$ initial state from the RA for the Schwinger model with parameters $(m, m_0, w, J, \theta) = (0, 0.5, 0.5, 0.5, 0)$. The exact results come from diagonalization of the Hamiltonian.

approximation of the time evolution operator. This is needed because of the large values needed for t_{rms}. The parameter controls the width of the peaks of the energy scanning procedure, as it enhances the filtering effect of Equation (4.30). In principle one could use very large t_{rms} to increase the resolution, but the downside is the increased length of the algorithm. The number of Trotter steps and the optimal δt can be inferred by our analysis of ASP, as the setup is similar when looking only at one ancilla qubit. In Figure 4.10 we show the effect of changing t_{rms} and how one can easily extract the energy eigenvalue by a Gaussian fit of the data around the peak. The relative error of the estimated energy and the exact value is 1.06^{-6}, this shows the high efficiency and conceptual simplicity of the RA for determining the spectrum of system.
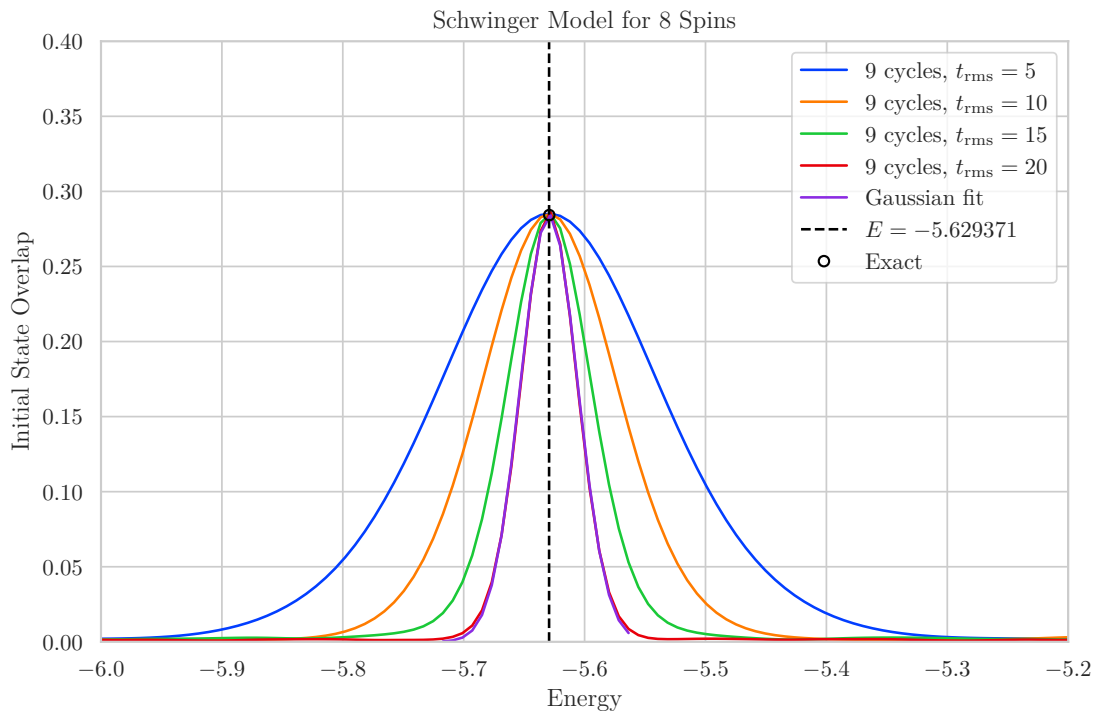
Figure 4.10: RA results for the ground state of the $N = 8$ system for different values of t_{rms}. A Gaussian fit to the data with the largest t_{rms} is shown in purple and the fit result of the central value in black.

4.5.1 Rodeo Algorithm for State Preparation

The RA can be extremely beneficial as an algorithm for state preparation because it allows preparing with ease excited states. In practice, one first needs to extract the eigenvalue from a Gaussian fit around a peak in the spectrum, then use the precisely determined value for the energy as the phase for the RA, as an iteration of the algorithm at this fixed energy value will then return the desired eigenstate if the ancilla qubits are all measured to be in the $|1\rangle$ state. Otherwise the state has to be discarded.

There are two main drawbacks of the RA as a state preparation algorithm. The first obvious one is the length of the algorithm, given by the multiple time evolutions of length $\approx t_{\mathrm{rms}}$ that need to be performed. When accounting for Toffoli gates, or the the more efficient controlled reversal gates method, the length of the algorithm far exceeds that of ASP. For this reason we do not report results for the relative error or the GS overlap obtained from the RA as it would be difficult to make a fair comparison with the methods presented in the two previous sections.

The second issue with the RA for state preparation is the efficiency: by construction, only quantum simulations where the ancilla qubits are measured to be $|1\rangle$ are in the desired state, which implies that if the initial state $|\psi_I\rangle$ has a low overlap with the desired state to be prepared then a very small number of runs in the simulations are actually useful and all the rest are to be discarded. One then must take extreme care in the choice of the initial state to maximize the output efficiency of the RA. Another possibility to enhance the efficiency is that of utilizing another algorithm for state preparation as a "preconditioner" for the RA, which would then be used to improve the results further. This would, however, limit the usage of the RA to the states accessible by the preconditioner.

In general, the RA is a promising algorithm for determining the energy spectrum of a theory and for state preparation. While the first task can be accomplished also on NISQ machines because of the relative ease of identifying the peaks of the spectrum even with low t_{rms}, the latter requires very long algorithms and is currently impractical, at least for the case of the Schwinger model that we have analyzed. Once the length of quantum algorithms will not be a limiting factor, the

RA will become a potential candidate for state preparation, especially for excited stated.

4.6 Comparison of Algorithms and Conclusions

When comparing the three algorithms we presented for state preparation of the Schwinger Hamiltonian, each presents strengths and weaknesses. ASP is the most straightforward algorithm to implement and has a long history of studies that can help understand when it can be safely applied. However, it has the major downside of leading to extremely long algorithms, which is a problem on NISQ machines. Once arbitrarily long algorithms will be enabled by hardware improvements, ASP could be used as a benchmark for other methods.

The QAOA method is very promising for its extremely short algorithms when compared to ASP or the RA. On current machines, it still is not applicable for the Schwinger model, but it in the foreseeable future it can become of use. The quality of the results is also much better than ASP, though it relies on one's ability to perform the minimization correctly. There could be systems for which the variational minimization performed with simulated annealing might fail to find a local minimum that is sufficiently close to the ground state for QAOA to be useful. This also shows another problem with the method, which is the need for prior knowledge of the true ground state to determine whether the minimizer actually converged, otherwise one cannot discern a local minimum. On systems where exact diagonalization or a Trotterized simulation is not possible to be done classically, one needs to rely on a hybrid algorithm, with the expectation values computed via quantum hardware. While this is an exciting prospect for QC in general, for the purpose of accurately preparing a state for further studies, this could be a limitation to the applicability of the method until such hybrid minimization become very accurate as well.

The Rodeo Algorithm has the clear advantage over ASP and QAOA of being able to not only prepare the ground state. The precision of the algorithm is also remarkable, as long as sufficiently large values for t_{rms} in the controlled time evolution are chosen. This shows one of the limitations of RA for NISQ machines: the very long length of the algorithms. Some methods for reducing the algorithm length are already being studied and since the RA has been proposed only very recently, in the future there could be significant improvements.

Going forward, the study of the Schwinger model can be a useful benchmark for LFT on

171

near-term quantum hardware, in a similar way as it was when the first numerical studies were performed on classical hardware. In order to be able to simulate QCD on a lattice using quantum computers, there are two main challenges to address: the representation of the $SU(3)$ symmetry group on qubits and the dimensionality of the lattice, which are two things that the toy model we presented here did not require.

APPENDICES

DISCRETIZATION OF THE SCHWINGER MODEL HAMILTONIAN ON A LATTICE

4A.1 Definitions

We assume the following conventions for the gamma matrices and the metric tensor in $1 + 1$ dimensions:

$$\gamma^0 = \sigma^3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{4A.1}$$

$$\gamma^1 = i\sigma^2 = i \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \tag{4A.2}$$

$$\gamma^5 = \gamma^0\gamma^1 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{4A.3}$$

In Minkowski space, the gamma matrices are defined by the anti-commutator

$$\{\gamma^\mu, \gamma^\nu\} = 2\eta^{\mu\nu}, \tag{4A.4}$$

so the metric tensor $\eta^{\mu\nu}$ with the choice of gamma matrices above is:

$$\eta^{\mu\nu} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{4A.5}$$

4A.2 Chiral Transformation

Start with Lagrangian for the Schwinger model with a θ-term from Equation (4.8)

$$\mathcal{L} = -\frac{1}{4}F_{\mu\nu}F^{\mu\nu} + \frac{g\theta}{4\pi}\epsilon_{\mu\nu}F^{\mu\nu} + i\overline{\psi}\gamma^\mu\left(\partial_\mu + igA_\mu\right)\psi - m\overline{\psi}\psi. \tag{4A.6}$$

We want to perform a global chiral transformation of the fermion fields in order to remove the free-standing θ-term. The transformation is: $\psi \to e^{i\frac{\theta}{2}\gamma_5}\psi$ for the field ψ and $\overline{\psi} \to \overline{\psi}e^{i\frac{\theta}{2}\gamma_5}$ for its conjugate. Note that the sign is positive in both exponentials.

The mass term then transforms as:

$$m\overline{\psi}\psi \to m\overline{\psi}e^{i\frac{\theta}{2}\gamma_5}e^{i\frac{\theta}{2}\gamma_5}\psi = m\overline{\psi}e^{i\theta\gamma_5}\psi \tag{4A.7}$$

and so we see the origin of this term in Equation (4.9). Later in Section 4.2.1 we split this term in two by rewriting the exponential as: $e^{i\frac{\theta}{2}\gamma_5} = \mathbb{1}\cos\frac{\theta}{2} + i\gamma_5 \sin\frac{\theta}{2}$. We also note that $\gamma^\mu e^{ia\gamma_5} = e^{-ia\gamma_5}\gamma^\mu$, so the derivative term of the Lagrangian transforms becomes invariant under the global chiral transformation:

$$i\overline{\psi}\gamma^\mu \left(\partial_\mu + igA_\mu\right)\psi \to i\overline{\psi}e^{i\frac{\theta}{2}\gamma_5}\gamma^\mu \left(\partial_\mu + igA_\mu\right)e^{i\frac{\theta}{2}\gamma_5}\psi \tag{4A.8}$$

$$= i\overline{\psi}e^{i\frac{\theta}{2}\gamma_5}\gamma^\mu e^{i\frac{\theta}{2}\gamma_5}\left(\partial_\mu + igA_\mu\right)\psi \tag{4A.9}$$

$$= i\overline{\psi}e^{i\frac{\theta}{2}\gamma_5}e^{-i\frac{\theta}{2}\gamma_5}\gamma^\mu \left(\partial_\mu + igA_\mu\right)\psi \tag{4A.10}$$

$$= i\overline{\psi}\gamma^\mu \left(\partial_\mu + igA_\mu\right)\psi. \tag{4A.11}$$

The transformed Lagrangian is then:

$$\mathcal{L}' = -\frac{1}{4}F_{\mu\nu}F^{\mu\nu} + \frac{g\theta}{4\pi}\epsilon_{\mu\nu}F^{\mu\nu} + i\overline{\psi}\gamma^\mu \left(\partial_\mu + igA_\mu\right)\psi - m\overline{\psi}e^{i\theta\gamma_5}\psi. \tag{4A.12}$$

This might not seem that useful, but when one considers the path integral for the generating functional:

$$Z[J] = \int \mathcal{D}\psi\mathcal{D}\overline{\psi}\mathcal{D}A_\mu e^{i\int dx\left(\mathcal{L}[\psi,\overline{\psi},A_\mu]+J(x)...\right)} \tag{4A.13}$$

under the same global chiral transformation, the measures $\mathcal{D}\psi$ and $\mathcal{D}\overline{\psi}$ also transform. It has been shown that under chiral transformation the integral measure for fermionic theories is not invariant [47]. The case for the measure in the Schwinger model has been studied in [48]. The relevant result is:

$$\mathcal{D}\psi\mathcal{D}\overline{\psi} \to \mathcal{D}\psi\mathcal{D}\overline{\psi}e^{-\frac{ig\theta}{4\pi}\int d^2x\,\epsilon_{\mu\nu}F^{\mu\nu}}. \tag{4A.14}$$

175

Thus, the generating functional becomes:

$$Z[J] = \int \mathcal{D}\psi \mathcal{D}\overline{\psi} \mathcal{D}A_\mu e^{i\int dx \left(\mathcal{L}' - \frac{g\theta}{4\pi}\epsilon_{\mu\nu}F^{\mu\nu} + J(x)...\right)}.$$

(4A.15)

We can then consider the relevant theory as the one with the Lagrangian

$$\mathcal{L}_{full} = \mathcal{L}' - \frac{g\theta}{4\pi}\epsilon_{\mu\nu}F^{\mu\nu}$$

(4A.16)

$$= -\frac{1}{4}F_{\mu\nu}F^{\mu\nu} + i\overline{\psi}\gamma^\mu\left(\partial_\mu + igA_\mu\right)\psi - m\overline{\psi}e^{i\theta\gamma_5}\psi.$$

(4A.17)

which is the one shown in Equation (4.9), where the θ dependence is shifted from an explicit θ-term to the fermion mass term.

4A.3 Legendre Transform

To obtain the Hamiltonian of the continuum Schwinger Model, we apply the Legendre transform to the Lagrangian of Equation (4.9):

$$\mathcal{H} = \frac{\partial\mathcal{L}}{\partial\dot{\psi}}\dot{\psi} + \frac{\partial\mathcal{L}}{\partial\dot{\overline{\psi}}}\dot{\overline{\psi}} + \frac{\partial\mathcal{L}}{\partial\dot{A}_\mu}\dot{A}_\mu - \mathcal{L}.$$

(4A.18)

The derivatives of the Lagrangian with respect to the dotted fermion fields are:

$$\frac{\partial\mathcal{L}}{\partial\dot{\psi}} = \frac{\partial\mathcal{L}}{\partial(\partial_0\psi)} = \frac{\partial}{\partial(\partial_0\psi)}\left[i\overline{\psi}\gamma^\mu\partial_\mu\psi\right]$$

(4A.19)

$$= \frac{\partial}{\partial(\partial_0\psi)}\left[i\overline{\psi}\gamma^0\partial_0\psi - i\overline{\psi}\gamma^1\partial_1\psi\right] = i\overline{\psi}\gamma^0$$

$$\frac{\partial\mathcal{L}}{\partial\dot{\overline{\psi}}} = \frac{\partial\mathcal{L}}{\partial(\partial_0\overline{\psi})} = 0.$$

(4A.20)

For the derivative with respect to $\dot{A}_\mu$, we first consider:

$$\frac{\partial\mathcal{L}}{\partial(\partial_\mu A_\nu)} = \frac{\partial\mathcal{L}}{\partial F_{\alpha\beta}}\frac{\partial F_{\alpha\beta}}{\partial(\partial_\mu A_\nu)}$$

(4A.21)

$$= \frac{\partial}{\partial F_{\alpha\beta}}\left[-\frac{1}{4}F_{\mu\nu}F^{\mu\nu}\right]\frac{\partial}{\partial(\partial_\mu A_\nu)}\left[\partial_\alpha A_\beta - \partial_\beta A_\alpha\right]$$

$$= \left[-\frac{1}{2}F^{\alpha\beta}\right]\left[\delta_{\mu\alpha}\delta_{\nu\beta} - \delta_{\mu\beta}\delta_{\nu\alpha}\right]$$

$$= -\frac{1}{2}\left[F^{\mu\nu} - F^{\nu\mu}\right] = -\frac{1}{2}\left[F^{\mu\nu} + F^{\mu\nu}\right] = -F^{\mu\nu}.$$

176

From which we get that the derivative with respect to $\dot{A}_\mu$ is:

$$\frac{\partial \mathcal{L}}{\partial \dot{A}_\mu} = \frac{\partial \mathcal{L}}{\partial (\partial_0 A_\mu)} = -F^{0\mu} = -\partial^0 A^\mu + \partial^\mu A^0. \tag{4A.22}$$

We simplify this further by using the *temporal gauge* $A^0 = 0$, thus remaining with so

$$\frac{\partial \mathcal{L}}{\partial \dot{A}_\mu} = -\partial^0 A^\mu = -\dot{A}^\mu. \tag{4A.23}$$

The Legendre transform thus becomes:

$$\mathcal{H} = \frac{\partial \mathcal{L}}{\partial \dot{\psi}} \dot{\psi} + \frac{\partial \mathcal{L}}{\partial \dot{\overline{\psi}}} \dot{\overline{\psi}} + \frac{\partial \mathcal{L}}{\partial \dot{A}_\mu} \dot{A}_\mu - \mathcal{L} \tag{4A.24}$$

$$= i\overline{\psi}\gamma^0\dot{\psi} - \dot{A}^\mu \dot{A}_\mu - \mathcal{L}$$

$$= i\overline{\psi}\gamma^0\partial_0\psi - \dot{A}^\mu \dot{A}_\mu + \frac{1}{4}F_{\mu\nu}F^{\mu\nu} - i\overline{\psi}\gamma^\mu (\partial_\mu + igA_\mu)\psi + m\overline{\psi}e^{i\theta\gamma_5}\psi.$$

In $1+1$ dimensions and using the temporal gauge, the terms containing only the gauge fields can be simplified to:

$$\frac{1}{4}F_{\mu\nu}F^{\mu\nu} - \dot{A}^\mu \dot{A}_\mu = \frac{1}{4}(\partial_\mu A_\nu - \partial_\nu A_\mu)(\partial^\mu A^\nu - \partial^\nu A^\mu) - \dot{A}^0 \dot{A}_0 - \dot{A}^1 \dot{A}_1 \tag{4A.25}$$

$$= \frac{1}{2}\partial_0 A_1 \partial^0 A^1 - \dot{A}^1 \dot{A}_1 = -\frac{1}{2}\dot{A}_1 \dot{A}^1$$

and we can also merge the two fermion derivative terms:

$$i\overline{\psi}\gamma^0\partial_0\psi - i\overline{\psi}\gamma^\mu\partial_\mu\psi = -i\overline{\psi}\gamma^1\partial_1\psi \tag{4A.26}$$

Ultimately, the Hamiltonian density is given by:

$$\mathcal{H} = -i\overline{\psi}\gamma^1 (\partial_1 + igA_1)\psi - \frac{1}{2}\dot{A}_1 \dot{A}^1 + m\overline{\psi}e^{i\theta\gamma_5}\psi. \tag{4A.27}$$

Then the Hamiltonian is

$$H = \int dx \left[-i\overline{\psi}\gamma^1 (\partial_1 + igA_1)\psi - \frac{1}{2}\dot{A}_1 \dot{A}^1 + m\overline{\psi}e^{i\theta\gamma_5}\psi \right], \tag{4A.28}$$

which matches the result shown in [53] and Equation (4.10) with the appropriate definition of the electric field $E(x) = F^{10} = -\dot{A}^1(x)$.

4A.4   Kogut-Susskind Hamiltonian

The discretization using staggered fermions introduced in Section 4.2.1 can seem odd at first glance as it is not the most straightforward way to apply it. However, some choices are made specifically because the final Hamiltonian that we desire is an Ising-like one, so all terms containing multiple fermion fields must either have the fields at the same site or at neighboring ones.

Remembering the definitions $A_1(x_n) \to \frac{\phi_n}{ag}$, $E(x_n) = \dot{A}_1(x_n) \to -gL$ and

$$
\frac{\chi_n}{\sqrt{a}} \leftrightarrow
\begin{cases}
\psi_u(x_n) \text{ for } n \text{ even} \\
\psi_d(x_n) \text{ for } n \text{ odd}
\end{cases}
\tag{4A.29}
$$

we have the tools for performing the following transformations for the terms in Equation (4A.28).

The Electric Field Term   The electric field term transforms simply as

$$
\dot{A}_1 \dot{A}^1 = (gL_n)(-gL_n) = -g^2 L_n^2.
\tag{4A.30}
$$

If we perform the integration, we get:

$$
\int dx \; \dot{A}_1 \dot{A}^1 = -ag^2 \sum_n L_n^2.
\tag{4A.31}
$$

The Mass-Theta Term   First, one expands the term into sine and cosine as follows:

$$
\overline{\psi} e^{i\theta\gamma_5}\psi = \mathbb{1}\cos\theta\overline{\psi}\psi + i\sin\theta\overline{\psi}\gamma_5\psi
\tag{4A.32}
$$

The mass term $\overline{\psi}\psi$ transforms in a simple way:

$$
\begin{aligned}
\psi^\dagger(x)\gamma^0\psi(x) &= \begin{pmatrix} \psi_u^\dagger(x) & \psi_d^\dagger(x) \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} \psi_u(x) \\ \psi_d(x) \end{pmatrix} \\
&= \psi_u^\dagger(x)\psi_u(x) - \psi_d^\dagger(x)\psi_d(x) \\
&= \frac{1}{a}\left(\chi_{2n}^\dagger \chi_{2n} - \chi_{2n-1}^\dagger \chi_{2n-1} \right).
\end{aligned}
\tag{4A.33}
$$

178

Then, when integrated, this gives:

$$\int dx\, \overline{\psi}\psi = a \sum_n \frac{1}{a} \left[\chi_{2n}^\dagger \chi_{2n} - \chi_{2n-1}^\dagger \chi_{2n-1} \right] = \sum_n (-1)^n \chi_n^\dagger \chi_n. \tag{4A.34}$$

Next, we consider the pseudo-mass term $\overline{\psi}\gamma_5\psi$

$$\overline{\psi}(x)\gamma_5\psi(x) = \psi^\dagger(x)\gamma^0\gamma^0\gamma^1\psi(x) \tag{4A.35}$$

$$= \begin{pmatrix} \psi_u^\dagger(x) & \psi_d^\dagger(x) \end{pmatrix} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} \psi_u(x) \\ \psi_d(x) \end{pmatrix}$$

$$= \psi_u^\dagger(x)\psi_d(x) - \psi_d^\dagger(x)\psi_u(x).$$

Notice that these terms mix the upper and lower components, however, each term is still at a single point in x. A straightforward translation to the staggered fermions would yield:

$$\overline{\psi}(x)\gamma_5\psi(x) = \frac{1}{a} \left[\chi_{2n}^\dagger \chi_{2n-1} - \chi_{2n-1}^\dagger \chi_{2n} \right]. \tag{4A.36}$$

However, this is undesirable as when summing over all sites x there would only be terms that contain χ_{2n} and χ_{2n-1}, but none that χ_{2n} and χ_{2n+}, which would force us to deal with even and odd sites differently when applying the Jordan-Wigner transformation. A more symmetric discretization can be obtained by using the approximation $\psi(x+1) = \psi(x) + \mathcal{O}(a)$, so that:

$$\psi_u^\dagger(x)\psi_d(x) - \psi_d^\dagger(x)\psi_u(x) \approx \frac{1}{2} \left[\psi_u^\dagger(x)\psi_d(x) - \psi_d^\dagger(x)\psi_u(x) \right] \tag{4A.37}$$

$$+ \frac{1}{2} \left[\psi_u^\dagger(x)\psi_d(x+1) - \psi_d^\dagger(x+1)\psi_u(x) \right].$$

where we take the average of the term between two neighboring sites. Keeping in mind that $x+1 \leftrightarrow x + 2a$ we get:

$$\overline{\psi}(x)\gamma_5\psi(x) \approx \frac{1}{2a} \left[\chi_{2n}^\dagger \chi_{2n-1} - \chi_{2n-1}^\dagger \chi_{2n} \right] \tag{4A.38}$$

$$+ \frac{1}{2a} \left[\chi_{2n}^\dagger \chi_{2n+1} - \chi_{2n+1}^\dagger \chi_{2n} \right]$$

$$= -\frac{1}{2a} \left[\chi_{2n-1}^\dagger \chi_{2n} - \chi_{2n}^\dagger \chi_{2n-1} \right]$$

$$+ \frac{1}{2a} \left[\chi_{2n}^\dagger \chi_{2n+1} - \chi_{2n+1}^\dagger \chi_{2n} \right],$$

179

which is now symmetric. When integrated over this term becomes:

$$\int dx\, \overline{\psi}(x)\gamma_5\psi(x) = \frac{1}{2}\sum_n (-1)^n \left[\chi_n^\dagger\chi_{n+1} - \chi_{n+1}^\dagger\chi_n\right]. \tag{4A.39}$$

Combining the two results we just obtained, the mass term of the Hamiltonian is then:

$$\int dx\, m\overline{\psi}e^{i\theta\gamma_5}\psi = \cos\theta\sum_n (-1)^n\chi_n^\dagger\chi_n + \frac{i}{2}\sin\theta\sum_n (-1)^n \left[\chi_n^\dagger\chi_{n+1} - \chi_{n+1}^\dagger\chi_n\right]. \tag{4A.40}$$

The Derivative Term Again, since we are looking for a symmetric treatment of the upper and lower components of the fermion spinors, we discretize the derivative term by applying a forward difference discretization for $\partial_1\psi_d(x)$ and a backward difference for $\partial_1\psi_u(x)$. This leads to:

$$\psi^\dagger(x)\gamma^0\gamma^1\partial_1\psi(x) = \begin{pmatrix} \psi_u^\dagger(x) & \psi_d^\dagger(x) \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \partial_1\psi_u(x) \\ \partial_1\psi_d(x) \end{pmatrix} \tag{4A.41}$$

$$= \psi_u^\dagger(x)\partial_1\psi_d(x) + \psi_d^\dagger(x)\partial_1\psi_u(x)$$

$$= \psi_u^\dagger(x)\frac{\psi_d(x+1) - \psi_d(x)}{2a} + \psi_d^\dagger(x)\frac{\psi_u(x) - \psi_u(x-1)}{2a}$$

$$= \frac{1}{2a^2}\chi_{2n}^\dagger \left[\chi_{2n+1} - \chi_{2n-1}\right] + \frac{1}{2a^2}\chi_{2n-1}^\dagger \left[\chi_{2n} - \chi_{2n-2}\right].$$

Thus, when integrating, we get a simple hopping term:

$$\int dx\, \overline{\psi}\gamma^1\partial_1\psi = a\sum_n \left[\frac{1}{2a^2}\chi_{2n}^\dagger \left[\chi_{2n+1} - \chi_{2n-1}\right] + \frac{1}{2a^2}\chi_{2n-1}^\dagger \left[\chi_{2n} - \chi_{2n-2}\right]\right]$$

$$= \frac{1}{2a}\sum_n \left[\chi_n^\dagger\chi_{n+1} - \chi_{n+1}^\dagger\chi_n\right].$$

The Vector Term The last term to transform is the one containing the gauge field:

$$\overline{\psi}(x)\gamma^1 A_1(x)\psi(x) = \begin{pmatrix} \psi_u^\dagger(x) & \psi_d^\dagger(x) \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} A_1(x)\psi_u(x) \\ A_1(x)\psi_d(x) \end{pmatrix} \tag{4A.42}$$

$$= \psi_u^\dagger(x)A_1(x)\psi_d(x) + \psi_d^\dagger(x)A_1(x)\psi_u(x).$$

Again, we have terms that mix the upper and lower components of $\psi(x)$, so we discretize the derivative in the same way that we did for the plain derivative term. The gauge field is not resolved

at the level of the staggered sub-lattice, so we set $A_1(x) = \phi_{2n}/(ag) = \phi_{2n-1}/(ag)$. Again using the approximation $\psi_d(x) \approx \psi_d(x+1)$ to mix the fermion field at different x we get:

$$
\begin{aligned}
\overline{\psi}(x)\gamma^1 A_1(x)\psi(x) &\approx \frac{1}{2}\left[\psi_u^\dagger(x)A_1(x)\psi_d(x) + \psi_d^\dagger(x)A_1(x)\psi_u(x)\right] \\
&\quad + \frac{1}{2}\left[\psi_u^\dagger(x)A_1(x)\psi_d(x+1) + \psi_d^\dagger(x+1)A_1(x)\psi_u(x)\right] \\
&= \frac{1}{2a^2g}\left[\chi_{2n}^\dagger\phi_{2n-1}\chi_{2n-1} + \chi_{2n-1}^\dagger\phi_{2n-1}\chi_{2n}\right] \\
&\quad + \frac{1}{2a^2g}\left[\chi_{2n}^\dagger\phi_{2n}\chi_{2n+1} + \chi_{2n+1}^\dagger\phi_{2n}\chi_{2n}\right].
\end{aligned}
\tag{4A.43}
$$

Upon integration, this term takes the simple form:

$$
\begin{aligned}
\int dx\, \overline{\psi}(x)\gamma^1 A_1(x)\psi(x) &= a\sum_n \frac{1}{2a^2g}\left[\chi_{2n}^\dagger\phi_{2n-1}\chi_{2n-1} + \chi_{2n-1}^\dagger\phi_{2n-1}\chi_{2n}\right] \\
&\quad + a\sum_n \frac{1}{2a^2g}\left[\chi_{2n}^\dagger\phi_{2n}\chi_{2n+1} + \chi_{2n+1}^\dagger\phi_{2n}\chi_{2n}\right] \\
&= \frac{1}{2ag}\sum_n \left[\chi_n^\dagger\phi_n\chi_{n+1} + \chi_{n+1}^\dagger\phi_n\chi_n\right].
\end{aligned}
\tag{4A.44}
$$

4A.4.1 The Discretized Schwinger Hamiltonian

The pieces of the continuum Hamiltonian, Equation (4A.28), translated to staggered fermions, are:

$$
\int dx\left[-\frac{1}{2}\dot{A}_1\dot{A}^1\right] = \frac{g^2a}{2}\sum_n L_n^2
\tag{4A.45}
$$

$$
\int dx\left[m\overline{\psi}e^{i\theta\gamma_5}\psi\right] = m\cos\theta\sum_n (-1)^n\chi_n^\dagger\chi_n + i\frac{m}{2}\sin\theta\sum_n(-1)^n\left[\chi_n^\dagger\chi_{n+1} - \chi_{n+1}^\dagger\chi_n\right]
$$

$$
\int dx\left[-i\overline{\psi}\gamma^1\partial_1\psi\right] = -\frac{i}{2a}\sum_n\left[\chi_n^\dagger\chi_{n+1} - \chi_{n+1}^\dagger\chi_n\right]
$$

$$
\int dx\left[g\overline{\psi}\gamma^1 A_1\psi\right] = \frac{1}{2a}\sum_n\left[\chi_n^\dagger\phi_n\chi_{n+1} + \chi_{n+1}^\dagger\phi_n\chi_n\right].
$$

With the assumption that the field ϕ_n is small, we can use the approximation $e^{\pm i\phi_n} \approx 1 \pm i\phi_n$ and introduce a sub-leading term to get to the form of the Hamiltonian shown in [53], which is the

one we showed in Equation (4.12):

$$H = -i \sum_n \left(\frac{1}{2a} - (-1)^n \frac{m}{2} \sin \theta \right) \left[\chi_n^\dagger e^{i\phi_n} \chi_{n+1} - \chi_{n+1}^\dagger e^{-i\phi_n} \chi_n \right] \qquad \text{(4A.46)}$$

$$+ m \cos \theta \sum_n (-1)^n \chi_n^\dagger \chi_n + \frac{g^2 a}{2} \sum_n L_n^2.$$

4A.5 Jordan-Wigner Transformation

The last step in constructing the Schwinger Hamiltonian for quantum simulations is to apply the Jordan-Wigner (JW) transformation to the discretized Hamiltonian. This is a straightforward process, but here we will show the most important details. The JW transformations read:

$$\chi_n = \left(\prod_{l<n} -iZ_l \right) \frac{X_n - iY_n}{2} \qquad \chi_n^\dagger = \left(\prod_{l<n} iZ_l \right) \frac{X_n + iY_n}{2}, \qquad \text{(4A.47)}$$

First, we look at Gauss's law, which can be written as:

$$L_n - L_{n-1} = \chi_n^\dagger \chi_n - \frac{1 - (-1)^n}{2} = \left(\prod_{l<n} iZ_l \right) \frac{X_n + iY_n}{2} \left(\prod_{l<n} -iZ_l \right) \frac{X_n - iY_n}{2} - \frac{1 - (-1)^n}{2}.$$

$$\text{(4A.48)}$$

For every given n, if we fix the value at the boundary to a constant L_0, we have by induction:

$$L_n = L_0 + \frac{1}{2} \sum_{l=1}^n \left(Z_l + (-1)^l \right). \qquad \text{(4A.49)}$$

The constant L_0 can be reabsorbed in the definition of θ, so we will not consider it. This allows us to transform the last term in Equation (4.12) as:

$$\sum_{n=1}^{N-1} L_n^2 = \frac{1}{4} \sum_{n=1}^{N-1} \left[\sum_{l=1}^n \left(Z_l + (-1)^l \right) \sum_{k=1}^n \left(Z_k + (-1)^k \right) \right] = \qquad \text{(4A.50)}$$

$$= \frac{1}{4} \sum_{n=1}^{N-1} \left[\sum_{l=1}^n \sum_{k=1}^n Z_l Z_k + 2 \sum_{l=1}^n \sum_{k=1}^n (-1)^k Z_l + \text{const} \right] =$$

$$= \frac{1}{2} \sum_{n=1}^{N-1} \left[\sum_{1 \le k < l \le n} Z_k Z_l - (n \mod 2) \sum_{l=1}^n Z_l \right],$$

where we neglected the constant terms. We can easily identify these two terms with H_{ZZ} and the second term of H_Z in Equation (4.16). Next, we focus our attention on the fermionic terms,

where we first perform the rescaling $\chi_n \to \prod_{l<n}[e^{-i\phi_n}]\chi_n$ to remove the explicit dependence on the gauge field ϕ_n. One can check that this leaves the Hamiltonian unchanged, as for the diagonal term all contributions cancel immediately, while for the hopping terms the exponentials cancel up to $n-1$, with the last being canceled by the factor $e^{i\phi_n}$ already present in the Hamiltonian. The diagonal term then, under the JW transformation, becomes:

$$\chi_n^\dagger \chi_n = \frac{1}{4} \left(\prod_{l<n} i Z_l \right) (X_n + i Y_n) \left(\prod_{l<n} -i Z_l \right) (X_n - i Y_n) = \tag{4A.51}$$

$$= \frac{1}{4} \left(\prod_{l<n} Z_l^2 \right) (X_n + i Y_n)(X_n - i Y_n) = \frac{i}{4}[X_n, Y_n] = -\frac{Z_n}{2}.$$

This can easily be applied to the cosine mass term and get the first term of H_Z. The last piece remaining is the hopping term. This is a well-known result and a classic example of the JW transformation:

$$\chi_{n+1}^\dagger \chi_n = \frac{1}{4} \left(\prod_{l<n+1} i Z_l \right) (X_{n+1} + i Y_{n+1}) \left(\prod_{l<n} -i Z_l \right) (X_n - i Y_n) = \tag{4A.52}$$

$$= \frac{i}{4}(X_{n+1} + i Y_{n+1}) Z_n (X_n - i Y_n) = -\frac{i}{4}(X_n - i Y_n)(X_{n+1} + i Y_{n+1})$$

$$\chi_n^\dagger \chi_{n+1} = \frac{1}{4} \left(\prod_{l<n} i Z_l \right) (X_n + i Y_n) \left(\prod_{l<n+1} -i Z_l \right) (X_{n+1} - i Y_{n+1}) =$$

$$= -\frac{i}{4}(X_n + i Y_n) Z_n (X_{n+1} - i Y_{n+1}) = \frac{i}{4}(X_n + i Y_n)(X_{n+1} - i Y_{n+1}),$$

so their difference is:

$$\chi_n^\dagger \chi_{n+1} - \chi_{n+1}^\dagger \chi_n = \frac{i}{2}(X_n X_{n+1} + Y_n Y_{n+1}). \tag{4A.53}$$

Putting all these results together and plugging them into Equation (4.16) one gets, after splitting the terms by their Pauli matrix content, i.e. H_Z for the diagonal terms, $H_\pm$ for the hopping term and H_{ZZ} for the Z interacting term one gets:

$$H_{ZZ} = \frac{J}{2} \sum_{n=2}^{N-1} \sum_{1 \le k < l \le n} Z_k Z_l$$

$$H_\pm = \frac{1}{2} \sum_{n=1}^{N-1} \left(w - (-1)^n \frac{m}{2} \sin\theta \right) [X_n X_{n+1} + Y_n Y_{n+1}] \tag{4A.54}$$

$$H_Z = \frac{m \cos\theta}{2} \sum_{n=1}^{N} (-1)^n Z_n - \frac{J}{2} \sum_{n=1}^{N-1} (n \bmod 2) \sum_{l=1}^{n} Z_l.$$

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] P. de Forcrand, "Simulating QCD at finite density," in *Proceedings of The XXVII International Symposium on Lattice Field Theory — PoS(LAT2009)*, Jun. 23, 2010, p. 010. DOI: 10.22323/1.091.0010.

[2] T. Izubuchi, S. Aoki, K. Hashimoto, Y. Nakamura, T. Sekido, and G. Schierholz, "Dynamical QCD simulation with theta terms," in *Proceedings of The XXV International Symposium on Lattice Field Theory — PoS(LATTICE 2007)*, Mar. 21, 2008, p. 106. DOI: 10.22323/1.042.0106.

[3] J. Dragos, T. Luu, A. Shindler, J. de Vries, and A. Yousif, "Confirming the existence of the strong CP problem in lattice QCD with the gradient flow," *Physical Review C*, vol. 103, no. 1, p. 015 202, Jan. 19, 2021. DOI: 10.1103/PhysRevC.103.015202.

[4] S. Takeda, "Tensor network approach to real-time path integral," 2019, Publisher: arXiv Version Number: 1. DOI: 10.48550/ARXIV.1908.00126.

[5] J. L. Park, "The concept of transition in quantum mechanics," *Foundations of Physics*, vol. 1, no. 1, pp. 23–33, Mar. 1, 1970, ISSN: 1572-9516. DOI: 10.1007/BF00708652.

[6] C. H. Bennett, "Logical reversibility of computation," *IBM Journal of Research and Development*, vol. 17, no. 6, pp. 525–532, Nov. 1973, ISSN: 0018-8646, 0018-8646. DOI: 10.1147/rd.176.0525.

[7] A. S. Holevo, "Bounds for the Quantity of Information Transmitted by a Quantum Communication Channel," p. 10,

[8] P. Benioff, "The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines," *Journal of Statistical Physics*, vol. 22, no. 5, pp. 563–591, May 1, 1980, ISSN: 1572-9613. DOI: 10.1007/BF01011339.

[9] P. A. Benioff, "Quantum mechanical hamiltonian models of discrete processes that erase their own histories: Application to turing machines," *International Journal of Theoretical Physics*, vol. 21, no. 3, pp. 177–201, Apr. 1, 1982, ISSN: 1572-9575. DOI: 10.1007/BF01857725.

[10] P. Benioff, "Quantum mechanical hamiltonian models of turing machines," *Journal of Statistical Physics*, vol. 29, no. 3, pp. 515–546, Nov. 1, 1982, ISSN: 1572-9613. DOI: 10.1007/BF01342185.

[11] T. D. Ladd, F. Jelezko, R. Laflamme, Y. Nakamura, C. Monroe, and J. L. O'Brien, "Quantum computers," *Nature*, vol. 464, no. 7285, pp. 45–53, Mar. 2010, ISSN: 0028-0836, 1476-4687. DOI: 10.1038/nature08812.

[12] J. Clarke and F. K. Wilhelm, "Superconducting quantum bits," *Nature*, vol. 453, no. 7198, pp. 1031–1042, Jun. 2008, ISSN: 0028-0836, 1476-4687. DOI: 10.1038/nature07128.

[13] H. Häffner, C. F. Roos, and R. Blatt, "Quantum computing with trapped ions," *Physics Reports*, vol. 469, no. 4, pp. 155–203, Dec. 1, 2008, ISSN: 0370-1573. DOI: 10.1016/j.physrep.2008.09.003.

[14] I. Bloch, "Ultracold quantum gases in optical lattices," *Nature Physics*, vol. 1, no. 1, pp. 23–30, Oct. 2005, ISSN: 1745-2473, 1745-2481. DOI: 10.1038/nphys138.

[15] T. Schenkel *et al.*, "Solid state quantum computer development in silicon with single ion implantation," *Journal of Applied Physics*, vol. 94, no. 11, pp. 7017–7024, Dec. 2003, ISSN: 0021-8979, 1089-7550. DOI: 10.1063/1.1622109.

[16] R. C. Brower, D. Berenstein, and H. Kawai, "Lattice gauge theory for a quantum computer," in *Proceedings of 37th International Symposium on Lattice Field Theory — PoS(LATTICE2019)*, Feb. 2, 2020, p. 112. DOI: 10.22323/1.363.0112.

[17] J. Preskill, "Simulating quantum field theory with a quantum computer," in *Proceedings of The 36th Annual International Symposium on Lattice Field Theory — PoS(LATTICE2018)*, May 29, 2019, p. 024. DOI: 10.22323/1.334.0024.

[18] T. Byrnes and Y. Yamamoto, "Simulating lattice gauge theories on a quantum computer," *Physical Review A*, vol. 73, no. 2, p. 022 328, Feb. 17, 2006, ISSN: 1050-2947, 1094-1622. DOI: 10.1103/PhysRevA.73.022328.

[19] N. Klco and M. J. Savage, "Digitization of scalar fields for quantum computing," *Physical Review A*, vol. 99, no. 5, p. 052 335, May 23, 2019, ISSN: 2469-9926, 2469-9934. DOI: 10.1103/PhysRevA.99.052335.

[20] N. Klco, M. J. Savage, and J. R. Stryker, "SU(2) non-abelian gauge field theory in one dimension on digital quantum computers," *Physical Review D*, vol. 101, no. 7, p. 074 512, Apr. 21, 2020, ISSN: 2470-0010, 2470-0029. DOI: 10.1103/PhysRevD.101.074512.

[21] Y. Meurice, R. Sakai, and J. Unmuth-Yockey, "Tensor lattice field theory for renormalization and quantum computing," *Reviews of Modern Physics*, vol. 94, no. 2, p. 025 005, May 26, 2022, ISSN: 0034-6861, 1539-0756. DOI: 10.1103/RevModPhys.94.025005.

[22] M. S. Anis *et al.*, "Qiskit: An open-source framework for quantum computing," 2021. DOI: 10.5281/zenodo.2573505.

[23] "IBM quantum." (2021), [Online]. Available: https://quantum-computing.ibm.com.

[24] K. Temme, S. Bravyi, and J. M. Gambetta, "Error mitigation for short-depth quantum circuits," *Physical Review Letters*, vol. 119, no. 18, p. 180 509, Nov. 3, 2017, ISSN: 0031-9007, 1079-7114.

DOI: 10.1103/PhysRevLett.119.180509.

[25] T. Giurgica-Tiron, Y. Hindy, R. LaRose, A. Mari, and W. J. Zeng, "Digital zero noise extrapolation for quantum error mitigation," in *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, Denver, CO, USA: IEEE, Oct. 2020, pp. 306–316, ISBN: 978-1-72818-969-7. DOI: 10.1109/QCE49297.2020.00045.

[26] B. M. Terhal, "Quantum error correction for quantum memories," *Reviews of Modern Physics*, vol. 87, no. 2, pp. 307–346, Apr. 7, 2015, ISSN: 0034-6861, 1539-0756. DOI: 10.1103/RevModPhys.87.307.

[27] E. T. Campbell, B. M. Terhal, and C. Vuillot, "Roads towards fault-tolerant universal quantum computation," *Nature*, vol. 549, no. 7671, pp. 172–179, Sep. 14, 2017, ISSN: 0028-0836, 1476-4687. DOI: 10.1038/nature23460.

[28] L. Grover and T. Rudolph, "Creating superpositions that correspond to efficiently integrable probability distributions," 2002, Publisher: arXiv Version Number: 1. DOI: 10.48550/ARXIV.QUANT-PH/0208112.

[29] A. N. Soklakov and R. Schack, "Efficient state preparation for a register of quantum bits," *Physical Review A*, vol. 73, no. 1, p. 012 307, Jan. 9, 2006, ISSN: 1050-2947, 1094-1622. DOI: 10.1103/PhysRevA.73.012307.

[30] P. A. M. Casares, "Circuit implementation of bucket brigade qRAM for quantum state preparation," 2020, Publisher: arXiv Version Number: 1. DOI: 10.48550/ARXIV.2006.11761.

[31] S. Lloyd, M. Mohseni, and P. Rebentrost, "Quantum algorithms for supervised and unsupervised machine learning," 2013, Publisher: arXiv Version Number: 2. DOI: 10.48550/ARXIV.1307.0411.

[32] A. Peruzzo *et al.*, "A variational eigenvalue solver on a photonic quantum processor," *Nature Communications*, vol. 5, no. 1, p. 4213, Sep. 2014, ISSN: 2041-1723. DOI: 10.1038/ncomms5213.

[33] J. Tilly *et al.*, "The variational quantum eigensolver: A review of methods and best practices," *Physics Reports*, vol. 986, pp. 1–128, Nov. 5, 2022, ISSN: 0370-1573. DOI: 10.1016/j.physrep.2022.08.003.

[34] I. Hamamura and T. Imamichi, "Efficient evaluation of quantum observables using entangled measurements," *npj Quantum Information*, vol. 6, no. 1, p. 56, Dec. 2020, ISSN: 2056-6387. DOI: 10.1038/s41534-020-0284-2.

[35] J. Schwinger, "The theory of quantized fields. i," *Physical Review*, vol. 82, no. 6, pp. 914–927, Jun. 15, 1951, ISSN: 0031-899X. DOI: 10.1103/PhysRev.82.914.

[36] J. Schwinger, "The theory of quantized fields. II," *Physical Review*, vol. 91, no. 3, pp. 713–728, Aug. 1, 1953, ISSN: 0031-899X. DOI: 10.1103/PhysRev.91.713.

[37] J. Schwinger, "Gauge invariance and mass," *Physical Review*, vol. 125, no. 1, pp. 397–398, Jan. 1, 1962, ISSN: 0031-899X. DOI: 10.1103/PhysRev.125.397.

[38] J. Schwinger, "Gauge invariance and mass. II," *Physical Review*, vol. 128, no. 5, pp. 2425–2429, Dec. 1, 1962, ISSN: 0031-899X. DOI: 10.1103/PhysRev.128.2425.

[39] S. Coleman, R. Jackiw, and L. Susskind, "Charge shielding and quark confinement in the massive schwinger model," *Annals of Physics*, vol. 93, no. 1, pp. 267–275, Sep. 1975, ISSN: 00034916. DOI: 10.1016/0003-4916(75)90212-2.

[40] S. R. Coleman, "More about the massive schwinger model," *Annals Phys.*, vol. 101, p. 239, 1976. DOI: 10.1016/0003-4916(76)90280-3.

[41] N. Manton, "The schwinger model and its axial anomaly," *Annals of Physics*, vol. 159, no. 1, pp. 220–251, Jan. 1985, ISSN: 00034916. DOI: 10.1016/0003-4916(85)90199-X.

[42] J. E. Hetrick and Y. Hosotani, "QED on a circle," *Physical Review D*, vol. 38, no. 8, pp. 2621–2624, Oct. 15, 1988, ISSN: 0556-2821. DOI: 10.1103/PhysRevD.38.2621.

[43] J. E. Hetrick, Y. Hosotani, and S. Iso, "The massive multi-flavor schwinger model," *Physics Letters B*, vol. 350, no. 1, pp. 92–102, May 4, 1995, ISSN: 0370-2693. DOI: 10.1016/0370-2693(95)00310-H.

[44] C. Adam, "Massive schwinger model within mass perturbation theory," *Annals of Physics*, vol. 259, no. 1, pp. 1–63, Sep. 1997, ISSN: 00034916. DOI: 10.1006/aphy.1997.5697.

[45] C. Adam, "Normalization of the chiral condensate in the massive schwinger model," *Physics Letters B*, vol. 440, no. 1, pp. 117–122, Nov. 1998, ISSN: 03702693. DOI: 10.1016/S0370-2693(98)01070-3.

[46] E. A. Martinez *et al.*, "Real-time dynamics of lattice gauge theories with a few-qubit quantum computer," *Nature*, vol. 534, no. 7608, pp. 516–519, Jun. 23, 2016, ISSN: 0028-0836, 1476-4687. DOI: 10.1038/nature18318.

[47] K. Fujikawa, "Path-integral measure for gauge-invariant fermion theories," *Physical Review Letters*, vol. 42, no. 18, pp. 1195–1198, Apr. 30, 1979, ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.42.1195.

[48] R. Roskies and F. Schaposnik, "Comment on fujikawa's analysis applied to the schwinger model," *Physical Review D*, vol. 23, no. 2, pp. 558–560, Jan. 15, 1981, ISSN: 0556-2821. DOI: 10.1103/PhysRevD.23.558.

[49]   J. Kogut and L. Susskind, "Hamiltonian formulation of wilson's lattice gauge theories," *Physical Review D*, vol. 11, no. 2, pp. 395–408, Jan. 15, 1975. DOI: 10.1103/PhysRevD.11.395.

[50]   L. Susskind, "Lattice fermions," *Physical Review D*, vol. 16, no. 10, pp. 3031–3039, Nov. 15, 1977, ISSN: 0556-2821. DOI: 10.1103/PhysRevD.16.3031.

[51]   C. J. Hamer, Z. Weihong, and J. Oitmaa, "Series expansions for the massive schwinger model in hamiltonian lattice theory," *Physical Review D*, vol. 56, no. 1, pp. 55–67, Jul. 1, 1997, ISSN: 0556-2821, 1089-4918. DOI: 10.1103/PhysRevD.56.55.

[52]   C. Muschik *et al.*, "U(1) wilson lattice gauge theories in digital quantum simulators," *New Journal of Physics*, vol. 19, no. 10, p. 103 020, Oct. 23, 2017, ISSN: 1367-2630. DOI: 10.1088/1367-2630/aa89ab.

[53]   B. Chakraborty, M. Honda, T. Izubuchi, Y. Kikuchi, and A. Tomiya, "Classically emulated digital quantum simulation of the schwinger model with a topological term via adiabatic state preparation," *Physical Review D*, vol. 105, no. 9, p. 094 503, May 13, 2022, ISSN: 2470-0010, 2470-0029. DOI: 10.1103/PhysRevD.105.094503.

[54]   T. Banks, L. Susskind, and J. Kogut, "Strong-coupling calculations of lattice gauge theories: (1 + 1)-dimensional exercises," *Physical Review D*, vol. 13, no. 4, pp. 1043–1053, Feb. 15, 1976, ISSN: 0556-2821. DOI: 10.1103/PhysRevD.13.1043.

[55]   P. Jordan and E. Wigner, "Über das paulische äquivalenzverbot," *Zeitschrift f. Physik*, vol. 47, no. 9, pp. 631–651, Sep. 1928, ISSN: 1434-6001, 1434-601X. DOI: 10.1007/BF01331938.

[56]   N. Hatano and M. Suzuki, "Finding exponential product formulas of higher orders," in *Quantum Annealing and Other Optimization Methods*, ser. Lecture Notes in Physics, A. Das and B. K. Chakrabarti, Eds., red. by R. Beig *et al.*, vol. 679, Nov. 16, 2005, pp. 37–68, ISBN: 978-3-540-27987-7.

[57]   M. Born and V. Fock, "Beweis des Adiabatensatzes," *Zeitschrift f. Physik*, vol. 51, no. 3, pp. 165–180, Mar. 1928, ISSN: 1434-6001, 1434-601X. DOI: 10.1007/BF01343193.

[58]   M. H. S. Amin, "Consistency of the adiabatic theorem," *Physical Review Letters*, vol. 102, no. 22, p. 220 401, Jun. 3, 2009, ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.102.220401.

[59]   T. Albash and D. A. Lidar, "Adiabatic quantum computation," *Reviews of Modern Physics*, vol. 90, no. 1, p. 015 002, Jan. 29, 2018, ISSN: 0034-6861, 1539-0756. DOI: 10.1103/RevModPhys.90.015002.

[60]   T. Kato, "On the adiabatic theorem of quantum mechanics," *Journal of the Physical Society of Japan*, vol. 5, no. 6, pp. 435–439, Nov. 15, 1950, ISSN: 0031-9015, 1347-4073. DOI: 10.1143/

JPSJ.5.435.

[61] S. Jansen, M.-B. Ruskai, and R. Seiler, "Bounds for the adiabatic approximation with applications to quantum computation," *Journal of Mathematical Physics*, vol. 48, no. 10, p. 102 111, Oct. 2007, ISSN: 0022-2488, 1089-7658. DOI: 10.1063/1.2798382.

[62] E. Farhi, J. Goldstone, S. Gutmann, and L. Zhou, "The quantum approximate optimization algorithm and the sherrington-kirkpatrick model at infinite size," *Quantum*, vol. 6, p. 759, Jul. 7, 2022, ISSN: 2521-327X. DOI: 10.22331/q-2022-07-07-759.

[63] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May 13, 1983, ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.220.4598.671.

[64] S. J. Glaser *et al.*, "Training schrödinger's cat: Quantum optimal control: Strategic report on current status, visions and goals for research in europe," *The European Physical Journal D*, vol. 69, no. 12, p. 279, Dec. 2015, ISSN: 1434-6060, 1434-6079. DOI: 10.1140/epjd/e2015-60464-1.

[65] K. Choi, D. Lee, J. Bonitati, Z. Qian, and J. Watkins, "Rodeo algorithm for quantum computing," *Physical Review Letters*, vol. 127, no. 4, p. 040 505, Jul. 23, 2021, ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.127.040505.

[66] M. Bee-Lindgren *et al.*, "Rodeo algorithm with controlled reversal gates," 2022, Publisher: arXiv Version Number: 1. DOI: 10.48550/ARXIV.2208.13557.

[67] Z. Qian, J. Watkins, G. Given, J. Bonitati, K. Choi, and D. Lee, "Demonstration of the rodeo algorithm on a quantum computer," 2021, Publisher: arXiv Version Number: 1. DOI: 10.48550/ARXIV.2110.07747.

[68] T. Toffoli, "Reversible computing," in *Automata, Languages and Programming*, J. Bakker and J. Leeuwen, Eds., red. by G. Goos *et al.*, vol. 85, Series Title: Lecture Notes in Computer Science, Berlin, Heidelberg: Springer Berlin Heidelberg, 1980, pp. 632–644, ISBN: 978-3-540-10003-4. DOI: 10.1007/3-540-10003-2_104.

[69] N. Yu, R. Duan, and M. Ying, "Five two-qubit gates are necessary for implementing the toffoli gate," *Physical Review A*, vol. 88, no. 1, p. 010 304, Jul. 30, 2013, ISSN: 1050-2947, 1094-1622. DOI: 10.1103/PhysRevA.88.010304.

CHAPTER 5

SUMMARY AND OUTLOOK

Lattice QCD is a mature field of research that, thanks to developments in HPC, is now capable of producing precision results for QCD at the hadronic scale. These results all rely on MCMC methods and are numerically very challenging. In particular, the highest computational cost is given by the iterative linear systems solvers needed both in generating ensembles of lattice gauge field configurations and in quark propagators for measuring observables. In this work, we presented three vastly different attempts to improve LQCD calculations.

In Chapter 2, we have presented a strategy to include ML methods into LQCD by reducing the cost of computing two-point hadron correlators by using data from quark propagators computed with different stopping criteria in the BiCGSTAB solver. The results for the GNN models indicate that the approach is not suited for accurately reconstructing the correlator's distribution. On the contrary, the SPNN models we presented indicate that the underlying idea is valid. The correlator calculated using ML, while much less expensive numerically, has a comparable distribution to that coming from a fully precise calculation. However, the ML procedure does lose some of the correlation between the different values of t, which make the determination of derived quantities, such as the effective mass, problematic. Thus, going forward, this issue needs to be addressed, possibly shifting the method to an approach similar to the TSM, where the ML model is used on a single configuration with data coming from multiple stochastic sources computed on it.

The OPENLAT Initiative aims to state-of-the-art ensembles of lattice gauge field configurations using SWF. In doing so, we want also to define and uphold quality standards for the ensembles and provide high-quality metadata to users. In Chapter 3, we have presented some examples of these procedures, one of which is the Bayesian analysis framework for hadron spectroscopy. The results of the method are reliable and stable under all the parameter space we have applied it to. The spectrum results also show the excellent scaling properties of the SWF ensembles, which make them good candidates for future studies of LQCD.

Quantum Computing is a relatively new approach in the context of lattice gauge theories, but it could revolutionize the field once the hardware advances allow large-scale simulations because it allows calculations in real-time, while not being plagued by the sign problem. Currently, there is no well-established way to deal with the $SU(3)$ symmetry group or a four-dimensional space-time on a QC, so research is focused on simpler models, which can give valuable insights on how to approach full QCD eventually. In Chapter 4, we investigated three algorithms for state preparation for the Schwinger model and discussed their strengths, weaknesses and potential applicability to NISQ machines. These algorithms are general enough to be applied to any model. In particular, the RA is very promising for after the NISQ era because it allows the preparation of both the ground state and excited states with high precision.