Exploring efficient methods for precision QCD calculations on the lattice

by

Lucius Nabil Bushnaq

Dissertation

Presented to the

University of Dublin, Trinity College

in fulfillment

of the requirements

for the Degree of

Doctor of Philosophy

University of Dublin, Trinity College

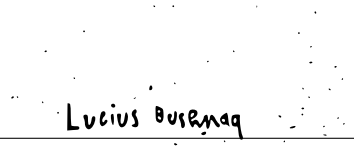
March 2023

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

I, the undersigned, agree to deposit this thesis in the University's open access institutional repository or allow the library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

I consent to the examiner retaining a copy of the thesis beyond the examining period, should they so wish (EU GDPR May 2018).

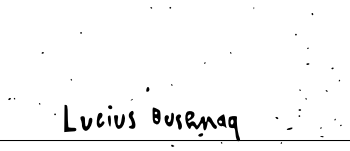


Lucius Nabil Bushnaq

August 22, 2023

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.



Lucius Nabil Bushnaq

August 22, 2023

Acknowledgments

I would like to express my deepest gratitude to my thesis supervisor, Mike Peardon, who always made time for me. He provided guidance, unwavering support, and encouragement throughout my Ph.D. journey, and his creative ideas were the basis for all the lines of research presented here. His expertise and insightful feedback have been instrumental in shaping my research work, and this thesis could not have existed without him.

I am also deeply grateful to my co-supervisor, Marina Marinkovic, for her guidance, expertise, and support. Her valuable feedback and constructive criticism have been invaluable, and the code that resulted in what is now chapter 2 might not be running even today if it was not for her help.

I would further like to thank others who gave advice that furthered the research presented here, such as Javad Komijani, who often offered very helpful insights on machine learning in discussions we had when we were both working in Zürich and Agostino Patella and the whole RC* Collaboration, for their help and support.

I am also grateful to the Trinity College Dublin lattice group, who created an excellent work environment, provided me with valuable resources, and facilitated a good atmosphere for academic inquiry.

I am also thankful to the EuroPLEx research network for bringing me in touch with other researchers in the field across Europe, which has been a great learning experience.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 813942. The research of AC, JL and AP is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - Projektnummer 417533893/GRK2575 "Rethinking Quantum Field Theory". The work was supported by the North-German Supercomputing Alliance (HLRN) with the project bep00085. The work was supported by the Poznan Supercomputing and Networking Center (PSNC) through grant numbers 450 and 466. The authors acknowledge

access to Piz Daint at the Swiss National Supercomputing Centre, Switzerland under the ETHZ's share with the project IDs go22 and go24. Computations were also performed on the Seagull cluster maintained by the Trinity Centre for High Performance Computing (TCHPC).

Lucius Nabil Bushnaq

University of Dublin, Trinity College
March 2023

Abstract

Exploring efficient methods for precision QCD calculations on the lattice

Lucius Nabil Bushnaq, Ph.D.
University of Dublin, Trinity College, 2023

Supervisor: Mike Peardon

We study a new technique for stochastic noise reduction in the calculation of propagators by implementing it in OpenQ*D for two ensembles with $\mathcal{O}(a)$ improved Wilson fermion action, with periodic boundary conditions and pion masses of 437 MeV and 331 MeV, for the connected and disconnected vector and pseudoscalar correlators. The technique performs low-mode averaging using the low-lying spectrum of the spatial Laplacian operator instead of the Dirac operator. We find that the technique yields no speedup compared to traditional methods, owing to the failure of its underlying assumption that the spectra of the spatial Laplacian and Dirac operators are sufficiently similar for the technique's purposes. However, the overlap is also found to be sensitive to rescaling the Laplacian by a real number, significantly enhancing performance. While still falling short of cost-effectiveness, we conclude that this suggests that the overlap of the low-lying operator spectrum with physical observables is sensitive to small changes, and future modifications may yet yield a cheaper alternative to the Dirac operator for low-mode averaging.

We further tested the ability of generative flow-based models based on neural networks to sample from the Lüscher action of the 2D U(1) Schwinger model. The Lüscher action approximates the fermion determinant through a polynomial, resulting in a local action of n_l additional multi-boson fields. Flow-based models we trained for $< 20,000$ update steps on a single Google Colab GPU proved able to sample from the Lüscher action for the Schwinger model on a $L/a = 8$ lattice with $n_L = 12$ multi-boson fields, and on a $L/a = 16$ lattice with $n_L = 4$ multi-boson fields. The models use the same fermionic layers for all the multi-boson fields. Thus, their total parameter count is not extensive in n_L. We conclude that with further algorithmic improvement, domain decomposition using the Lüscher action seems to be a promising avenue for enabling the sampling of significantly larger lattice volumes with flow-based models.

Contents

List of Tables

List of Figures

List of Publications

This thesis is based on the following peer-reviewed publications:

- **Implementing noise reduction techniques into the OpenQ*D package**

Chapter 1

Introduction

Historically, most calculations of observables in the Standard Model have been performed perturbatively. The only currently available alternative, evaluating the path integral numerically, has, for the most part, been deployed in cases where perturbative calculations are difficult or not possible at all. Perhaps the most well-known cases of this are quantum chromodynamics (QCD) Calculations at low energies. Due to the running of the strong coupling g_s with the renormalization energy scale, it is large at the scales relevant for quark confinement, glueball physics, nuclear spectroscopy, and other low-energy phenomena involving the strong force, making perturbative evaluation of the path integral impossible. The field of lattice QCD founded the numerical approach to Quantum Field Theory to calculate such physics. The Standard Model action for the QCD sector, written in the Euclidean metric convention, is [1]:

$$
\begin{aligned}
&S_{\mathrm{QCD}}[\psi,\overline{\psi},A] = S_F[\psi,\overline{\psi},A] + S_G[A]\,, \qquad \text{with}\\
&S_F[\psi,\overline{\psi},A] = \int d^4x \sum_{f=1}^{n_f} \overline{\psi}^{(f)}(x) D^{(f)}(x) \psi^{(f)}(x)\\
&S_G[A] = \frac{1}{4g_s^2} \sum_{i=1}^{8} \int d^4x F_{\mu\nu}^{(i)}(x) F_{\mu\nu}^{(i)}(x)\\
&D^{(f)}(x) = \left(\gamma_\mu(\partial_\mu + iA_\mu(x)) + m^{(f)}\right)\\
&F_{\mu\nu}^{(i)}(x) = \partial_\mu A_\nu^{(i)}(x) - \partial_\nu A_\mu^{(i)}(x) - f_{ijk} A_\mu^{(j)}(x) A_\nu^{(k)}(x)\\
&A_\mu(x) = \sum_{i=1}^{8} A_\mu^{(i)}(x) T_i\,,
\end{aligned}
\tag{1.1}
$$

where T are a basis for the fundamental representation of SU(3) and f_{ijk} are the SU(3) structure constants. To approximate the path integral, the theory is placed on a discrete Euclidean space-time lattice of finite extent. Points on this lattice are separated by the lattice parameter a, which also serves as an ultraviolet regulator [2]. Typical lattice sizes in state-of-the-art calculations can be in the range of 128×96^3 points [3]. Any physics that takes place above the energy scale of the lattice cut-off, or in volumes that do not fit into the lattice volume V, cannot be captured by the calculation. Historically, this has greatly limited the precision and applicability of numeric path integral evaluation. At the border of the lattice, a boundary condition for the theory must be chosen. There are many possible choices for the lattice action, constrained only by the need to recover the action and results of the continuum theory when taking the limit $a \to 0$. The gauge part of the lattice action can be written as

$$S_G[U] = \frac{\beta}{3} \sum_{n \in V} \sum_{\mu < \nu} \mathrm{Re\ tr}[\mathbb{1} - P_{\mu,\nu}(n)]\,, \tag{1.2}$$

with $\beta = \frac{6}{g_s^2}$ and P the "plaquette"

$$P_{\mu,\nu}(n) = U_\mu(n) U_\nu(n+\hat{\mu}) U_\mu^\dagger(n+\hat{\nu}) U_\nu^\dagger(n)\,. \tag{1.3}$$

Here, U are the gauge link transporter variables connecting fields on adjacent lattice sites. They are related to the gauge fields A of the continuum through $U_\mu(n) = \mathbb{1} + iaA_\mu(n) + \mathcal{O}(a^2)$. In contrast to the continuum case, the path integral on the lattice is thus defined over variables that take values in SU(3). Common actions in use differ, among other things, in the formulation of the lattice fermions. A naive choice of lattice fermion action fails to recover the continuum physics due to the fermion doubling problem [1]. One possible action, used in chapter two of this work, uses the Wilson fermion formulation

$$\begin{aligned} S_F[\psi, \overline{\psi}, U] &= a^4 \sum_{n,m \in V} \sum_{f=1}^{n_f} \overline{\psi}^{(f)}(n) D^{(f)}(n|m) \psi^{(f)}(m) \\ D^{(f)}(n|m) &= \mathbb{1} - \kappa^{(f)} \sum_\mu \Big((\mathbb{1} - \gamma_\mu) U_\mu(n) \delta_{n+\hat{\mu},m} + (\mathbb{1} + \gamma_\mu) U_\mu^\dagger(n-\hat{\mu}) \delta_{n-\hat{\mu},m} \Big) \\ \text{with} \quad \kappa^{(f)} &= \frac{1}{2(am^{(f)} + 4)}\,. \end{aligned} \tag{1.4}$$

To set the correct value for the lattice parameters a, a scale setting procedure needs to be performed, typically by calculating some observable on the lattice that is also available to

high precision from experiment, or a different calculation. After an observable is calculated on the lattice, the calculation is repeated for lattice actions with smaller lattice parameters a, and different lattice sizes, keeping the physical volume constant. The continuum value of the observable at $a = 0$ is then extrapolated by fitting a curve to these points.

For example, to calculate a two-point correlation function of the form

$$\begin{aligned}\langle O(n)\overline{O}(m)\rangle &= \langle\overline{\psi}^{(f_1)}(n)\Gamma\psi^{(f_2)}(n)\overline{\psi}^{(f_2)}(m)\Gamma\psi^{(f_1)}(m)\rangle \\ &= \frac{1}{Z}\int \mathcal{D}[U]\mathcal{D}[\psi,\overline{\psi}]e^{-S_G[U]-S_F[\psi,\overline{\psi},U]}\overline{\psi}^{(f_1)}(n)\Gamma\psi^{(f_2)}(n)\overline{\psi}^{(f_2)}(m)\Gamma\psi^{(f_1)}(m) \\ \text{with } Z &= \int \mathcal{D}[U]\mathcal{D}[\psi,\overline{\psi}]e^{-S_G[U]-S_F[\psi,\overline{\psi},U]}\,,\end{aligned} \tag{1.5}$$

on the lattice, where Γ is an operator of arbitrary spin structure, one first performs the integral over the fermionic Grassmann variables to obtain an expression that can be evaluated numerically

$$\langle O(n)\overline{O}(m)\rangle = \frac{1}{Z}\int \mathcal{D}[U]\det[\prod_f D^{(f)}]e^{-S_G[U]}\mathrm{tr}[\Gamma(D^{(f_1)})^{-1}(n|m)\Gamma(D^{(f_2)})^{-1}(m|n)]\,. \tag{1.6}$$

Notably, while the original action is local, this expression is effectively not, due to the presence of the terms $\det[D^{(f)}]$ in the integrand. The primary computational challenge now lies in sampling the integrand efficiently to perform Monte Carlo integration. Current leading approaches use Markov-chain methods and the Hybrid Monte Carlo (HMC) algorithm [4], which attempts to find a new Markov candidate in the space of gauge configurations by evolving the current configuration with a molecular dynamics equation in a Markov time τ. Generating enough well-distributed gauge configurations for state-of-the-art lattice calculations requires significant computational investment and the resulting Markov chains are often stored for years for use by various lattice collaborations. Current lattice projects frequently look to utilize resources on the exascale [5]. Once the configurations are generated, the observables need to be calculated. As shown above, this often necessitates inverting the Dirac operator D, another numerically expensive operation. If the full inverse operator was needed, the cost of inverting a $n_c \times n_s \times N_0 \times N^3$ dimensional operator, where $n_c = 3$ and $n_s = 4$ are the numbers of colors and spin states and N_0 and N are the temporal and spatial extent of the lattice, would be outright prohibitive. However, in many applications, only a few entries of the inverse, such as one or a few columns, are required. Then, one can choose a source term,

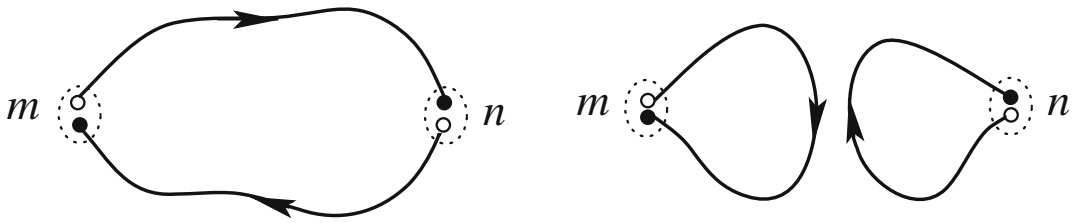


Figure 1.1: Connected and disconnected contributions to mesonic correlators, credit [1].

such as a point source,

$$\sum_{n'} D(n|n')\phi(n') = \delta_{n,0}\,, \tag{1.7}$$

drastically reducing the required computational resources. However, some observables require evaluation of so-called disconnected contributions,

$$\frac{1}{Z}\int \mathcal{D}[U]\det[D]e^{-S_G[U]}\mathrm{tr}[\Gamma(D^{(f_1)})^{-1}(n|n)]\mathrm{tr}[\Gamma(D^{(f_2)})^{-1}(m|m)]\,, \tag{1.8}$$

see also figure 1.1. For this, one can use stochastic sources

$$\begin{aligned}\sum_{n'} D^{(f_1)}(n|n')\phi^{(r)}(n') &= \eta^{(r)}(n)\\ \lim_{N_r\to\infty}\frac{1}{N_r}\sum_r \eta^{(r)}(n)(\eta^{(r)})^\dagger(m) &= \delta_{n,m}\,,\end{aligned} \tag{1.9}$$

where the entries of the noise vectors $\eta^{(r)}(n)$ are samples from a distribution, such as a Gaussian. Solutions $\phi^{(r)}$ to the equation can then be used to evaluate expressions

$$\mathrm{tr}[\Gamma(D^{(f_1)})^{-1}(n|n)] = \lim_{N_r\to\infty}\frac{1}{N_r}\sum_r \mathrm{tr}[(\eta^{(r)})^\dagger(n)\Gamma\phi^{(r)}(n)] \tag{1.10}$$

for finite N_r, introducing stochastic noise in addition to the gauge noise from sampling the integrand, but making the calculation tractable. The exact form of the noise vector varies and can be chosen through schemes like dilution [6] to minimize the stochastic noise for a particular observable.

Over the years, advances in both computing hardware and algorithms have allowed lattice QCD as a field to carry out calculations on larger lattices at higher precision. Calculations used to be carried out at higher physical pion masses and only extrapolated to the physical point. The physical pion mass was too light, increasing correlation lengths on the lattice

beyond what the methods of the time could handle. Nowadays, it is becoming increasingly common for state-of-the-art calculations to be performed at the physical point. In the past, the determinant of the Dirac operator on the path integral was often set to one, an operation equivalent to neglecting interactions mediated through the quark fields. Current calculations often use 2+1 or even 2+1+1 dynamic quark flavors, see e.g. [7], neglecting only interactions through the extremely heavy top and bottom quarks. Contributions from electromagnetic interactions used to not be treated on the lattice, now they increasingly are, using methods like $C^\star$ boundary conditions [8, 9]. As a result, lattice QCD is now occasionally deployed in circumstances where a different evaluation method would also be feasible.

A recent example of this comes from calculations of the magnetic moment of the muon. At present, there are possible discrepancies between the numerical value of the magnetic moment of the muon $a_\mu = \frac{(g_\mu - 2)}{2}$ measured in experiment [10, 11] and derived through some theoretical calculations [12].

In the standard model, a_μ receives contributions from the QED, electroweak, and hadronic sectors

$$a_\mu = a_\mu^{QED} + a_\mu^{EW} + a_\mu^{HAD} \,. \tag{1.11}$$

The QED and electroweak contributions can be evaluated purely through direct perturbative calculation. a_μ^{HAD} involves hadronic loops with contributions from the infrared momentum region, where perturbative treatments of QCD break down. The leading contribution of this type is the hadronic vacuum polarization a_μ^{HVP}, at $\mathcal{O}(\alpha^2)$ in the fine structure constant α, [13], pictured in figure 1.2. This contribution cannot be neglected, as the error on a_μ^{HAD} currently contributes a significant share of the total error on a_μ. It must thus be evaluated through alternate means. One way of evaluating a_μ^{HAD} is to do so indirectly. Using dispersion relations, a_μ^{HVP} may be related to the R-ratio of $e + e- \rightarrow$ hadrons, allowing determination from experimental data. a_μ calculated through this method differs from the value of a_μ measured experimentally by 4.3σ. The recent improvements in lattice QCD precision calculations have allowed for an alternative, direct evaluation of the hadronic sector contributions. In particular, the BMW collaboration achieved a calculation of a_μ^{HVP} with a relative error of 0.008. The resulting estimate of a_μ differs from the experimentally measured value by only 1.5σ [14] and differs from the value calculated from dispersion relations by 2.1σ, see figure 1.3. Since then, other lattice collaborations have provided estimates of similar precision for parts of the HVP contribution, see figure 1.4, [24, 3]. So far, all of these lattice QCD estimates are

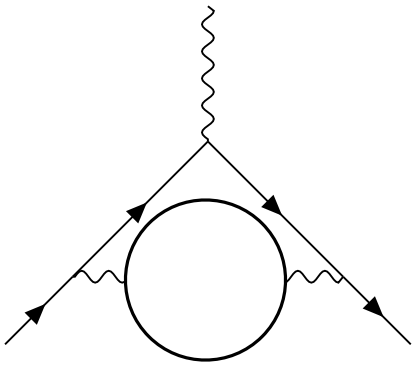


Figure 1.2: Hadronic vacuum polarization diagram, credit [13], contributing to a_μ at $\mathcal{O}(\alpha^2)$ in the fine structure constant α. A similar contribution with two hadronic loops in the photon propagator appears at $\mathcal{O}(\alpha^3)$.

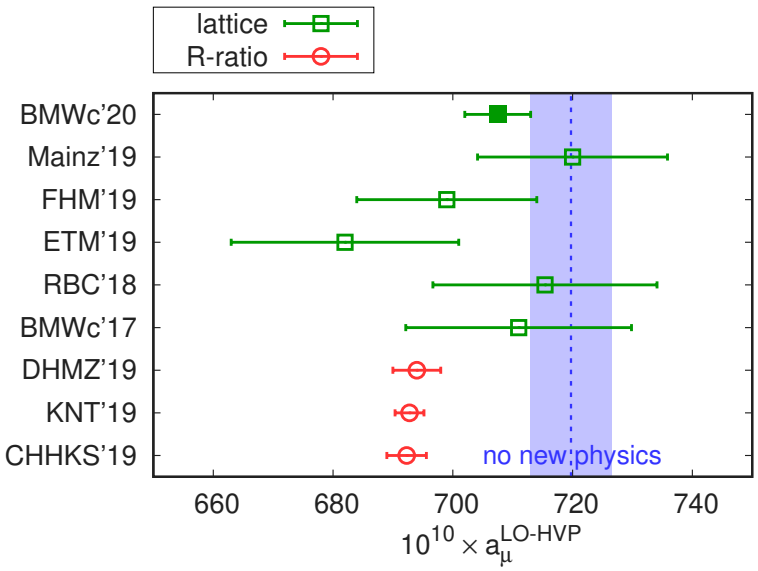


Figure 1.3: Standard model predictions for a_μ^{HVP} calculated by various groups. The green bars represent lattice calculations. From top to bottom, [14], which is also the source of this graphic, [15], [16], [17], [18], [19]. The red bars show dispersive results, [20], [21], [22], [23]. The blue-shaded region indicates the region of compatibility with the BNL results.

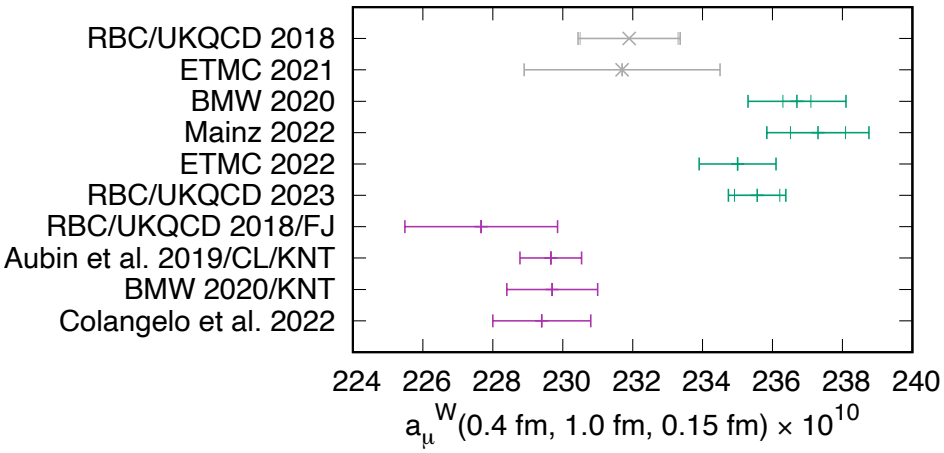


Figure 1.4: The Euclidean window HVP contribution a_μ^W, calculated by various groups. a_μ^W was defined in [18] to exclude the more difficult parts of the correlator for small and large Euclidean times, enabling easier comparison between different results. Lattice calculations are depicted in gray and green, with the first error bar showing statistical uncertainties, and the second statistic and systematic uncertainties added in quadrature. Starting from the top, [18, 25, 14], [26, 24, 3] which is also the source of the graphic. In purple are dispersive results adjusted to the window, [18, 27, 14, 28].

in agreement with each other, and by extension, in disagreement with the dispersive result. Debates on which calculation is correct are ongoing.

As the HVP calculation served as the original physical motivation for this work, before it expanded into a more general effort to increase the efficiency of lattice QCD calculations, its basic outline will be briefly recounted here. The leading order vacuum polarization can be expressed as an integral over the electromagnetic vector current correlator, and is related to a_μ^{HVP} through convolution with a known kernel function, determined purely by QED interactions [29] [30]. In the time momentum representation [31], this can be written as

$$\begin{aligned} V_\mu(x) &= Z_V \sum_f Q_f \overline{\psi}_f(x)\gamma_\mu\psi_f(x) \quad \text{(local formulation)} \\ G(t) &= \frac{1}{3}\sum_{\mu=1,2,3}\int d^3x \langle V_\mu(x)V_\nu(0)\rangle \\ a_\mu^{HVP} &= \alpha^2 \int_0^\infty dt\, K(t)G_{1\gamma I}(t)\,, \end{aligned} \tag{1.12}$$

where $G_{1\gamma I}(t)$ is the one photon irreducible part of $G(t)$, and $K(t)$ is the QED kernel function

integrated over momenta

$$K(t) = \int_0^\infty \frac{dQ^2}{m_\mu^2} \frac{\left(\frac{Q^2}{m_\mu^2} + 2 - \sqrt{\frac{Q^2}{m_\mu^2}\left(\frac{Q^2}{m_\mu^2}+4\right)}\right)^2}{\sqrt{\frac{Q^2}{m_\mu^2}\left(\frac{Q^2}{m_\mu^2}+4\right)}} \left(t^2 - \frac{4}{Q^2}\sin\left(\frac{Qt}{2}\right)\right), \tag{1.13}$$

with m_μ being the muon mass. The task of lattice calculations is thus to evaluate the one photon irreducible part of the vector-vector correlator $\int d^3x \langle V_\mu(x) V_\nu(0)\rangle$ on the lattice. The short distance region of the correlator presents a challenge on the lattice due to the cutoff but can be calculated in perturbation theory. The long-distance region is difficult to evaluate due to the exponentially increasing relative error as the correlator falls off, but it can be estimated using information about the correlator's finite-volume spectrum since only low-lying states contribute to it.

The vector correlator receives both connected and disconnected contributions. The disconnected contribution is considerably more difficult to evaluate, in part because its gauge noise stays constant with Euclidean time, while the correlator itself decays with $e^{-E_i t_0}$, where E_i are the energy eigenstates of the correlator. To reduce the computational cost required to evaluate the contributions, and thus increase precision, state-of-the-art calculations make use of several algorithmic tricks. One such trick is called low-mode averaging [32, 33]. It aims to reduce the number of stochastic sources needed to evaluate D^{-1}. At longer distances t_0, most of the high-lying modes of the correlator with greater energy have decayed. One could thus expect that most of the signal should be formed by the lowest eigenmodes of the Dirac operator. So, if one solves for the first few eigenmodes of the Dirac operator explicitly, the remainder of the correlator can be small, necessitating far fewer stochastic sources to calculate it to sufficient precision. The cost of this approach is that the Dirac eigenmodes are expensive to evaluate. The first part of this work catalogs an attempt to improve on this technique by using the eigenmodes of the distillation operator [34] instead. The low-lying modes of the distillation operator are conjectured to have significant overlap with the low energy modes of correlators, suggesting that it could be used in the same manner as the Dirac operator, while its eigenmodes are cheaper to calculate due to its smaller size of $n_c \times N^3$.

The second part of this work is concerned with a more exploratory attempt at improving lattice calculations in a manner that diverges from the current dominant paradigm. In recent years, attempts to sample from the QCD action using trivializing maps [35] learned through

generative neural network models, have gone from a conceptual proposal to the first, very small scale, implementations [36, 37]. In this approach, one uses a parameterized function $f(\theta)$ to map from some trivial distribution p_0 over field configurations to the distribution given by the action e^{-S}. The function class $f(\theta)$ is chosen as a neural network architecture, and the parameters θ are set through numeric optimization. In machine learning, this is also referred to as training the network.

In this way, the cost of generating gauge configurations is front-loaded into training a generative neural network adept at this task. Once a network is trained, configurations can often be sampled at a comparatively rather negligible cost. These configurations can then be used as proposals in a Metropolis-Hastings algorithm, replacing the HMC proposals in the current paradigm. Current implementations are exploratory and take place at small scales. [38] recently published a network trained to sample a QCD action with $n_f = 2$ flavors on a $4D$ lattice with $N_0 = N = 4$. In contrast to HMC samplers, neural network methods propose new field configurations without reference to any previous configurations. As a consequence, they typically have much less trouble with sampling from different topological sectors and do not necessarily suffer from the critical slowing down HMC algorithms experience at fine lattice spacing, where correlation lengths are large, and the auto-correlation times of the Markov chain thus become very long. However, current models do not yet sample at large lattice volumes. In the second part of this work, the idea of improving the volume scaling of these methods by factorizing the path integral into many smaller lattice volumes with the Lüscher action [39] is explored. The Lüscher action approximates the inverse Dirac operator D^{-1} through a polynomial expansion, resulting in an integrand with only nearest-neighbor interactions. When first conceived in the 1990s, this technique for localizing the action proved largely unworkable with classic sampling techniques, but neural network-based samplers may not be subject to the same restriction.

In so far as these two halves of the work have a unifying theme, it is attempts to push the efficiency of lattice calculations further forward, into an era of high precision and broad applicability. The goal is enabling more precise determinations of quantities like the eta prime mass [40] and the glue physics to which it is coupled, the hadronic vacuum polarization, the parton distributions of hadrons such as the proton [41], and perhaps even the calculation of larger scale nuclear physics involving bound states of multiple hadrons from first principles.

Chapter 2

Distillation low mode averaging

2.1 Introduction

The distillation operator [34] is a tool originally conceived for the smearing of quark sources for hadron creation operators, and has seen widespread success in this application. When using lattice QCD determinations of two-point correlation functions $\langle O_2(t)O_1(0)\rangle \propto \sum_i a_i e^{-E_i t}$, to determine the energies of a particle p in its ground state E_0, or one of its low-lying excited states, the precision of the determination relies on choosing interpolators O such that the correlator has high overlap with the propagator of p at the desired low energies.

Exact formulations such that, e.g., $\langle O_2(t)O_1(0)\rangle \propto e^{-E_0 t}$ are usually unavailable. Instead, the choice of interpolators is made heuristically. The construction of interpolators customarily begins by forming quark bilinears on a time slice, $\overline{q}_1(n,t)\Gamma q_2(n,t), n \in L^3, t \in L_0$, with the quark content and the spin structure Γ chosen to match the quantum numbers of p. Since such bilinears are located at a single lattice point, the states they create have a high overlap with states of high energy. This typically leads to significant contributions from excited states to the correlator. The extraction of low-lying state energies thus becomes challenging or even infeasible. To alleviate this, one attempts to "smear out" the bilinears over many lattice points, to obtain a spatial wave-function with lower contributions from high energy states, and thus greater overlap with the low-lying spectrum of p.

There are many possible ways to obtain a spatial profile well-matched to the bilinear. Distillation is one such technique, intended to optimize the computational efficiency of the

smearing procedure. One applies the distillation operator $\Box(t)$ to each quark field

$$O(t) = \Box(t)\overline{q}_1\Gamma\Box(t)q_2. \tag{2.1}$$

The distillation operator is formed out of the eigenmodes of the spatial Laplacian

$$\begin{aligned} \bigtriangledown^2_{ms,ns'}(t) &= -6\delta_{ss'}\delta_{mn} + \delta_{ss'}\sum_{j=1}^{3}\left(U_j(m,t)\delta_{m+\hat{j},n} + U_j^\dagger(m-\hat{j},t)\delta_{m-\hat{j},n}\right) \\ \Box(t) &= V(t)V^\dagger(t) = \sum_{k=1}^{N_v} v^{(k)}(t)v^{(k)\dagger}(t)\,, \end{aligned} \tag{2.2}$$

where $v^{(k)}(t)$ are the eigenvectors of $\bigtriangledown^2$, N_v is the number of eigenvectors calculated, and $V(t)$ is a matrix with the eigenvectors as column entries. Since the operator does not affect the spin space, the eigenvectors come in spin-degenerate sets of four.

This smearing procedure seems to succeed at projecting the resulting correlator into the subspace of low-lying Dirac modes, excluding the majority of contributions from high lying states $E_i, i \gg 1$. This could lead one to suspect that the low-lying spectrum of the spatial Laplacian has substantial overlap with the low-lying spectrum of the Dirac operator. Since the spatial Laplacian has a lower dimensionality than the Dirac operator, $n_c \times N^3$ compared to $n_c \times n_s \times N_0 \times N^3$, its eigenspectrum is also cheaper to calculate. Thus, if the overlap between the two operators' spectra were "close enough", it could be computationally efficient to substitute the Distillation spectrum for the Dirac spectrum in many applications where the latter is required.

A potential practical use case and test of this notion could be "distillation low-mode averaging". Conventional low-mode averaging is a technique used to reduce the computational effort involved in solving the Dirac equation with stochastic sources. The first N_v low-lying modes $|i\rangle$ of the Dirac spectrum are calculated explicitly. Then, correlators can be broken up into contributions from the N_v low-lying modes and the higher-lying modes. For a connected

mesonic observable, for example:

$$
\begin{aligned}
&C = C_l + C_h + C_{hl} + C_{lh}\,, \quad \text{with} \\
&C_l = \sum_{i=0,j=0}^{N_v} \mathrm{tr}[\Gamma_a D^{-1}(n,m)|i\rangle\langle i|\Gamma_b|j\rangle\langle j|D^{-1}(m,n)] \\
&C_h = \mathrm{tr}[\Gamma_a D^{-1}(n,m)\left(1 - \sum_{i=0}^{N_v}|i\rangle\langle i|\right)\Gamma_b\left(1 - \sum_{j=0}^{N_v}|j\rangle\langle j|\right)D^{-1}(m,n)] \\
&C_{hl} = \mathrm{tr}[\Gamma_a D^{-1}(n,m)\left(1 - \sum_{i=0}^{N_v}|i\rangle\langle i|\right)\Gamma_b\sum_{j=0}^{N_v}|j\rangle\langle j|D^{-1}(m,n)] \\
&C_{lh} = \mathrm{tr}[\Gamma_a D^{-1}(n,m)\sum_{i=0}^{N_v}|i\rangle\langle i|\Gamma_b\left(1 - \sum_{j=0}^{N_v}|j\rangle\langle j|\right)D^{-1}(m,n)]\,.
\end{aligned}
\tag{2.3}
$$

Since $|i\rangle$ are eigenvectors of D^{-1}, C_l is now directly available without stochastic noise. C_h, C_{hl} and C_{lh} can be calculated through stochastic inversion as normal. If the low-lying Dirac modes make up the majority contribution to the observable, $\frac{C_l}{C} \approx 1$, this can yield a significantly smaller statistical error $\sigma(C)$ on the estimate of the correlator C than investing computational resources into more stochastic sources. Among other applications, low-mode averaging has been used to calculate the vector correlator that forms the HVP contribution to the muon $g-2$. Currently, [14] is the most precise lattice determination of the HVP contribution, and uses low-mode averaging (among other noise reduction techniques).

Here, the idea is to do the same, but using the spectrum of the distillation operator instead. If the overlap between the low-lying part of the distillation and Dirac operator spectra is high, this would reduce stochastic noise just as with conventional low-mode averaging, while using up less time to calculate the operator eigenmodes. In this chapter, a mathematical and computational setup for calculating the connected and disconnected contributions to the vector correlator with distillation low-mode averaging is described, and results for computations on 30 gauge configurations on a 64×32^3 lattice are presented. As a further test case, the same is done for the pseudoscalar correlator. We find that distillation low-mode averaging falls short of yielding computationally cost-effective noise reduction for both observables in our setup. This seems to be due to the technique's underlying assumption, that the low-lying distillation spectrum forms the largest contribution to these observables just as with the Dirac spectrum, being falsified.

2.2 Mathematical setup

To implement this method for the calculation of the connected contributions, we begin by splitting the Hilbert space into the low-lying space of the Laplacian modes, and the "rest-space" at both source and sink by inserting $1 - \Box(t) + \Box(t)$, giving

$$\begin{aligned}
&\langle \overline{\psi}^A(t)\Gamma_\mu \psi^B(t)\overline{\psi}^B(t')\Gamma_\nu\psi^A(t')\rangle_F \\
&\quad = \langle \overline{\psi}^A(t)\Gamma_\mu \left(1-\Box(t)+\Box(t)\right)\psi^B(t)\overline{\psi}^B(t')\Gamma_\nu\left(1-\Box(t')+\Box(t')\right)\psi^A(t')\rangle_F \\
&\quad = \langle \overline{\psi}^A(t)\Gamma_\mu \left(1-\Box(t)\right)\psi^B(t)\overline{\psi}^B(t')\Gamma_\nu\left(1-\Box(t')\right)\psi^A(t')\rangle_F \\
&\qquad + 2\langle \overline{\psi}^A(t)\Gamma_\mu\Box(t)\psi^B(t)\overline{\psi}^B(t')\Gamma_\nu\left(1-\Box(t')\right)\psi^A(t')\rangle_F \\
&\qquad + \langle \overline{\psi}^A(t)\Gamma_\mu\Box(t)\psi^B(t)\overline{\psi}^B(t')\Gamma_\nu\Box(t')\psi^A(t')\rangle_F \\
&\quad =: C_{rest}(t) + 2C_{cross}(t) + C_{dist}(t)\,.
\end{aligned} \tag{2.4}$$

We have used time-reversal symmetry to combine the two mixed contributions. If we were working with conventional low-mode averaging, and $\Box(t)$ consisted of eigenmodes of the Dirac operator, $C_{cross}(t)$ would vanish since $\Box$ and the propagator would commute. But here, we need to calculate it on top of the other two contributions. Now we perform a one-end insertion [42] to introduce the stochastic sources

$$\begin{aligned}
C_{dist}(t) &= \mathrm{tr}[\gamma_5\Gamma_\mu V^\dagger(t)D^{-1}(t,t')V(t')\Gamma_\nu\gamma_5 V^\dagger(t')D^{-1^\dagger}(t',t)V(t)] \\
C_{rest}(t) &= \frac{1}{N_r}\mathrm{tr}[\gamma_5\Gamma_\mu\left(1-\Box(t)\right)D^{-1}(t,t')\eta^{(r)}(t')\eta^{\dagger(r)}(t')\Gamma_\nu\gamma_5\left(1-\Box(t')\right)D^{-1^\dagger}(t',t)] \\
C_{cross}(t) &= \frac{1}{N_r}\mathrm{tr}[\gamma_5\Gamma_\mu\Box(t)D^{-1}(t,t')\eta^{(r)}(t')\eta^{\dagger(r)}(t')\Gamma_\nu\gamma_5\left(1-\Box(t')\right)D^{-1^\dagger}(t',t)]\,.
\end{aligned} \tag{2.5}$$

Here, $\mathbb{Z}(2)\times\mathbb{Z}(2)$ [43] stochastic wall sources located on a single time slice are used. Note that to avoid calculating the Dirac equation for the distillation modes on multiple time slices, it is necessary to place all stochastic sources on the same time slice. This can increase the gauge noise. To calculate $C_{dist}(t)$ we then need the vectors $\phi^{(k)} := D^{-1}(t,t')v^{(k)}(t')$ which form the perambulator [34]. Thus, we need to solve the Dirac equation with the eigenvectors of the spatial Laplacian at the desired source time as sources.

To calculate $C_{rest}(t)$ and $C_{cross}(t)$, we need

$$\begin{aligned}\phi^{(r)} &:= D^{-1}(t,t')\eta^{(r)}(t') \\ \overline{\phi}^{(r)} &:= D^{-1}(t,t')\left(1-\Box(t')\right)\eta^{(r)}(t')\,,\end{aligned} \tag{2.6}$$

making for $N_v + 2N_R$ needed solutions to the Dirac equation in total. The two-point correlation function for any Γ_μ, Γ_ν can then be formed through the contraction of these solutions by calculating the spin structure explicitly. This is detailed in section 2.3.1, as well as [44]. For the technique to be worth the cost of the increased number of solutions needed per independent stochastic source, as well as the cost of calculating the distillation modes themselves, $C_{cross}(t)$ and $C_{rest}(t)$ need to be small relative to $C_{dist}(t)$.

Analogously, to implement the technique for disconnected contributions to two-point correlators, the Hilbert space is split into the low-lying space of the Laplacian modes, and the "rest-space" in both traces, again by inserting $1-\Box(t)+\Box(t)$, giving

$$\begin{aligned}&\langle\overline{\psi}(t)\Gamma_\mu\psi(t)\rangle_F\langle\overline{\psi}(t_0)\Gamma_\nu\psi(t_0)\rangle_F = \mathrm{tr}[\Gamma_\mu D^{-1}(t,t)]\mathrm{tr}[\Gamma_\nu D^{-1}(t',t')] \\ &= \sum_{r,r'}\mathrm{tr}[\eta^{\dagger(r)}\left(1-\Box(t)\right)\Gamma_\mu D^{-1}(t,t)\eta^{(r)}]\mathrm{tr}[\eta^{\dagger(r')}\left(1-\Box(t')\right)\Gamma_\mu D^{-1}(t',t')\eta^{(r')}] \\ &+ \sum_{k,k'}\mathrm{tr}[v^{(k)\dagger}(t)\Gamma_\mu D^{-1}(t,t)v^{(k)}(t)]\mathrm{tr}[v^{(k')\dagger}(t')\Gamma_\mu D^{-1}(t',t')v^{(k')}(t')] \\ &+ \sum_{r,k}(\mathrm{tr}[\left(1-\Box(t)\right)\Gamma_\mu D^{-1}(t,t)\eta^{(r)}\eta^{\dagger(r)}]\mathrm{tr}[v^{(k)\dagger}(t')\Gamma_\mu D^{-1}(t',t')v^{(k)}(t')] \\ &+ \mathrm{tr}[v^{(k)\dagger}(t)\Gamma_\mu D^{-1}(t,t)v^{(k)}(t)]\mathrm{tr}[\eta^{\dagger(r)}\left(1-\Box(t')\right)\Gamma_\mu D^{-1}(t',t')\eta^{(r)}]) \\ &= C_{rest}(t) + C_{dist}(t) + C_{cross}(t)\,.\end{aligned} \tag{2.7}$$

Thus, to calculate C_{rest} and $C_{cross}(t)$ for the vector and pseudoscalar operators, we need

$$\phi^{(r)}(t) := D^{-1}(t,t)\eta^{(r)}\,, \tag{2.8}$$

with separate stochastic sources $\eta^{(r)}$ for the two traces, and

$$\phi^{(k)}(t) := D^{-1}(t,t)v^{(k)}(t)\,, \tag{2.9}$$

calculated for all time slices. Thus, $N_0 \times N_v + 2N_R$ Dirac inversions are required in total, compared to $2N_R$ without distillation low mode averaging.

2.3 Simulation setup, implementation, and testing

The technique was implemented for the connected and disconnected vector and pseudoscalar correlators in OpenQ*D, using the PRIMME eigensolver [45] to obtain the distillation eigenmodes. The spatial covariant Laplacian is passed to PRIMME as a callable function, parallelized with MPI. The solver then returns the specified number of low-lying eigenmodes. This is repeated for the spatial Laplacian on every time slice of the lattice for every configuration. No preconditioning for the operator is employed. We tested the performance of the method on two CLS data sets with two flavors of dynamical $\mathcal{O}(a)$ improved Wilson fermions. The computation used periodic boundary conditions. The determinations of the lattice spacing and other configuration parameters were taken from [46, 47]. The code can be found at https://github.com/LuciusBushnaq/PhDQCD under the GNU General Public License v3.0.

Ensemble	$N_0 \times N^3$	β	κ	$m_\pi L$	a [fm]	m_π [MeV]
E5	$64x32^3$	5.30	0.13625	4.7	0.0658(7)(7)	437
A5	$64x32^3$	5.20	0.13594	4.0	0.0755(9)(7)	331

Table 2.1: Ensemble parameters

2.3.1 Setup and tests of OpenQ*D

In this section, the implementation of the normal, non-distilled correlators into OpenQ*D is described, and some tests of the simulation setup are carried out. Using $\mathbb{Z}(2)\times\mathbb{Z}(2)$ stochastic wall sources,

$$\lim_{N_r\to\infty}\frac{1}{N_r}\sum_r \eta_\mu^{\alpha(r)}(n,t)\eta_\nu^{\beta(r)\dagger}(m,t') = \delta_{n,m}\delta_{t,t_0}\delta_{t',t_0}\delta_{\mu,\nu}\delta_{\alpha,\beta}$$

$$\eta(n,t) = \delta_{t,t_0}\begin{bmatrix}\mathcal{D}(n,c1,s1)\\ \mathcal{D}(n,c2,s1)\\ \dots\\ \mathcal{D}(n,c3,s4)\end{bmatrix}, \qquad \mathcal{D} = \frac{1}{\sqrt{2}}(\pm 1 \pm i) \tag{2.10}$$

$$D^{-1}(m,t|n,t') =: \frac{1}{N_r}\sum_r \phi^{(r)}(m,t)\eta^{(r)}(n,t')^\dagger\,,$$

the connected pseudoscalar

$$\begin{aligned}\frac{1}{N^3}\sum_{n,m}\langle P(n,t)P(m,t_0)\rangle_{F,C} &= \frac{1}{N^3}\sum_{n,m}\langle\overline{\psi}(n,t)\gamma_5\psi(n,t)\overline{\psi}(m,t_0)\gamma_5\psi(m,t_0)\rangle_{F,C}\\ &= -\frac{1}{N^3}\sum_{n,m}\mathrm{tr}[\gamma_5 D^{-1}(n,t|m,t_0)\gamma_5 D^{-1}(m,t_0|n,t)]\\ &= -\frac{1}{N^3}\sum_{n,m}\lim_{N_r\to\infty}\frac{1}{N_r}\sum_r \mathrm{tr}[D^{-1}(n,t|m,t)\eta^{(r)}(m,t)\\ &\qquad\qquad \eta^{(r)\dagger}(m',t')D^{-1\dagger}(n,t|m',t')]\\ &= -\frac{1}{N^3}\sum_{n}\lim_{N_r\to\infty}\frac{1}{N_r}\sum_r \phi^{(r)\dagger}(n,t)\phi^{(r)}(n,t)\,,\end{aligned} \tag{2.11}$$

and axial vector correlators

$$\begin{aligned}\frac{1}{N^3}\sum_{n,m}\langle A_\mu(n,t)P(m,t_0)\rangle_{F,C} &= \frac{1}{N^3}\langle\overline{\psi}(n,t)\gamma_5\gamma_\mu\psi(n,t)\overline{\psi}(m,t_0)\gamma_5\psi(m,t_0)\rangle_{F,C}\\ &= -\frac{1}{N^3}\sum_{n,m}\mathrm{tr}[\gamma_5\gamma_\mu D^{-1}(n,t|m,t_0)\gamma_5 D^{-1}(m,t_0|n,t)]\\ &= -\frac{1}{N^3}\sum_{n}\lim_{N_r\to\infty}\frac{1}{N_r}\sum_r \phi^{(r)\dagger}(n,t)\gamma_\mu\phi^{(r)}(n,t)\,,\end{aligned} \tag{2.12}$$

were both calculated using the one-end insertion trick [42] for noise reduction. $\langle ...\rangle_{F,C}$ here refers to the connected part of the fermionic expectation value. The configurations used were taken from the CLS data set E5, with a pion mass of 437 MeV, periodic boundary conditions, and $\mathcal{O}(a)$ improved Wilson fermions, as shown in table 2.1. Throughout this chapter, error analyses were conducted with Jackknife resampling. For details, see section B.1 in the appendix.

Effective mass curves for the axial and pseudoscalar correlator are shown in figures 2.1 and 2.2. The effective mass $\mathrm{M_{eff}}$ here is defined as [1]

$$\frac{C(t)}{C(t+a)} = \frac{\cosh\left(\mathrm{M_{eff}}(t/a - L_0/(2a))\right)}{\cosh\left(\mathrm{M_{eff}}(t/a+1 - L_0/(2a))\right)} \tag{2.13}$$

for the pseudoscalar and

$$\frac{C(t)}{C(t+a)} = \frac{\sinh\left(\mathrm{M_{eff}}(t/a - L_0/(2a))\right)}{\sinh\left(\mathrm{M_{eff}}(t/a+1 - L_0/(2a))\right)} \tag{2.14}$$

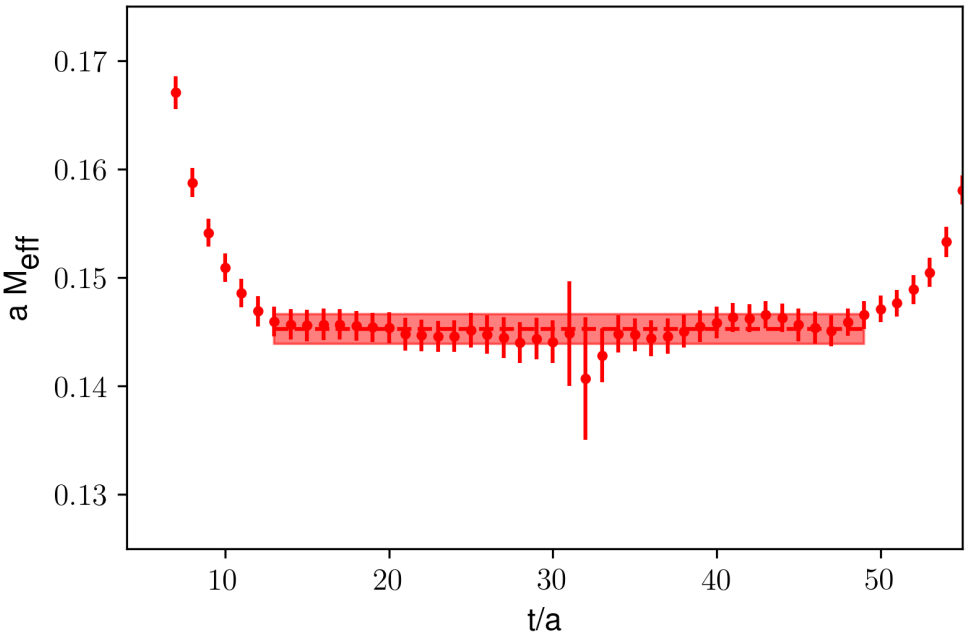


Figure 2.1: The plateau of the effective masses of the connected pseudoscalar correlator calculated on 30 gauge configuration taken from the E5 data set. The shading indicates the one-sigma region of the fit to the plateau. The plateau matches the value for am_π calculated on the E5 set in [47] inside two standard deviations.

for the axial vector correlator. Fitting a function

$$A \cosh\left(m(t/a - L_0/(2a))\right) \tag{2.15}$$

to the plateau in the effective mass of the pseudoscalar yields a mass in lattice units of 0.1453 ± 0.0014, matching the literature value given in [47] for the same configuration set inside the error. Next, the connected contribution to the vector correlator was implemented into the program. Again, one-end insertion is used for the stochastic sources

$$\begin{aligned}
\frac{1}{N^3}\sum_{n,m}\langle V_\mu(n,t)V_\nu(m,t_0)\rangle_{F,C} &= \frac{1}{N^3}\sum_{n,m}\langle\overline{\psi}(n,t)\gamma_\mu\psi(n,t)\overline{\psi}(m,t_0)\gamma_\nu\psi(m,t_0)\rangle_F \\
&= -\frac{1}{N^3}\sum_{n,m}\mathrm{tr}[\gamma_\mu D^{-1}(n,t|m,t_0)\gamma_\nu D^{-1}(m,t|n,t_0)] \\
&= \frac{1}{N^3}\sum_{n}\lim_{N_r\to\infty}\frac{1}{N_r}\sum_{r}{\phi_2^{(r)}}^\dagger(n,t)\gamma_5\gamma_\mu\phi_1^{(r)}(n,t)\,, \quad \text{with} \\
\phi_1^{(r)}(n,t) &= \sum_z D^{-1}(n,t|z,t')\eta^{(r)}(z,t') \\
\phi_2^{(r)}(n,t) &= \sum_z D^{-1}(n,t|z,t')\gamma_\nu\gamma_5\eta^{(r)}(z,t')\,.
\end{aligned} \tag{2.16}$$

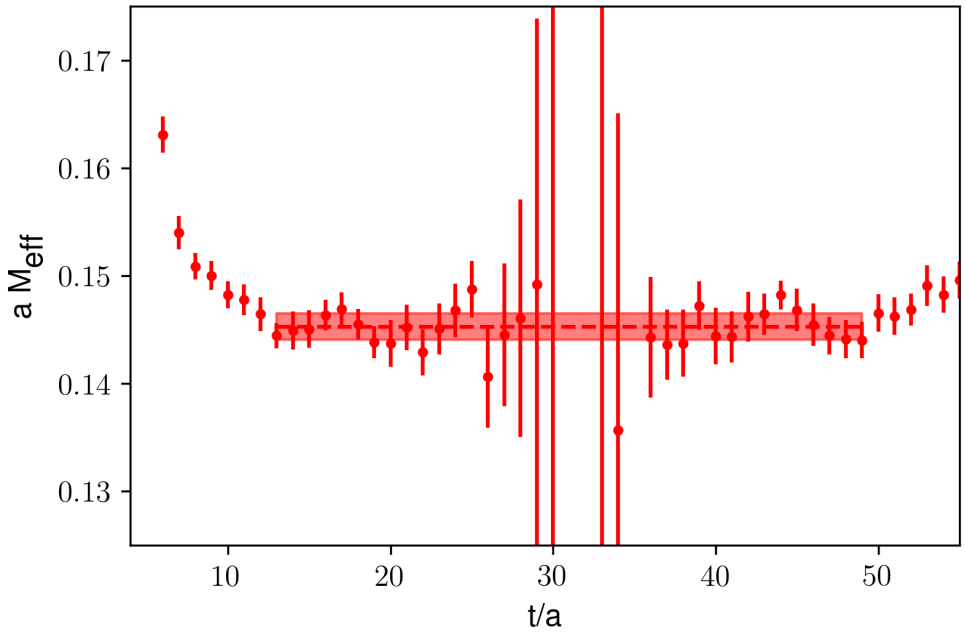


Figure 2.2: Effective masses of the zeroth component of the axial vector correlator, calculated on the E5 data set. The shading indicates the one-sigma region of the fit to the plateau.

Here, $\phi_1(n,t)$ and $\phi_2(n,t)$ are solutions to the Dirac equation with stochastic sources $\eta(z)$, $\gamma_\nu\gamma_5\eta(z)$ respectively. Thus, we would naively need to solve the Dirac equation five times to calculate all 16 components of the vector correlator. However, following the implementation in [44], we can cut this down to four solutions. These solutions can also be used to calculate other connected two-point correlation functions with arbitrary spin structure $\langle\overline{\psi}_s(n)\Gamma_1\psi_s(n)\overline{\psi}_s(m)\Gamma_2\psi_s(m)\rangle_F$. This is achieved by using linked spin-diluted sources. That is, we take

$$\eta'_i(n) = \delta_{i,s}\xi(n),\ i \in \{1,2,3,4\}\,. \tag{2.17}$$

We then solve the Dirac equation for each of these sources

$$D(n,z)\phi'_s(z) = \eta'_s(n)\,. \tag{2.18}$$

Due to the linearity of solutions, we can then form the solutions to Dirac equations with stochastic sources of the general form

$$D(n,z)\phi(z) = \Gamma\xi(n) = \Gamma\sum_i \eta'^{(r)}_i(n)\,, \tag{2.19}$$

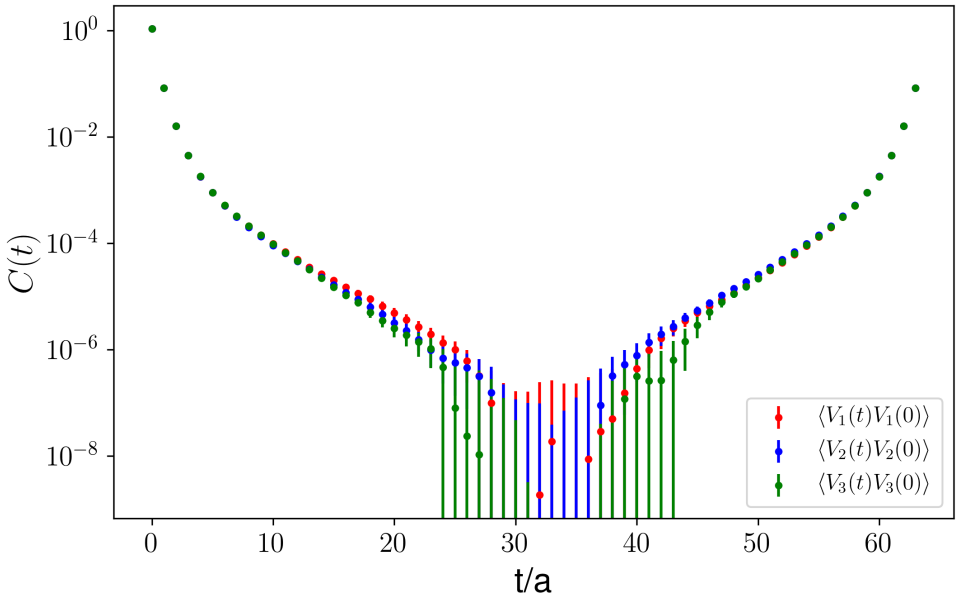


Figure 2.3: The three diagonal entries of the connected vector correlator, calculated on 30 gauge configurations from the E5 set with 48 stochastic sources. The shading indicates the one-sigma regions of the fits to the respective plateaus. There is good agreement inside the error bars.

using linear combinations

$$\phi(n) = \sum_i \Gamma\, \phi_i'(n)\,. \tag{2.20}$$

Plots for the vector correlator are shown in figures 2.3 and 2.4, plots for its effective mass in figures 2.5 and 2.6, for 48 and 384 sources respectively. The effective mass is defined the same as in 2.13 for the pseudoscalar. Fitting 2.15 to the plateau regions yields results that agree inside the errors for all three diagonal components of the vector correlator. The off-diagonal components of the vector correlator do not show a signal. This serves as a further test of the validity of the Dirac solutions since it relies on the cancellation between different gauge configurations. The signal-to-noise ratio shrinks considerably in the long-distance part of the vector correlator.

Disconnected correlation functions

The disconnected pseudoscalar and vector correlators present a bigger computational challenge than their connected counterparts. While the signal on the disconnected correlators decays exponentially at larger times, their variance stays constant, leading to a rapidly increasing relative error $\frac{\sigma(C)}{C}$. Since the propagator already needs to be calculated on all time slices for disconnected observables, statistical power can be further increased through trans-

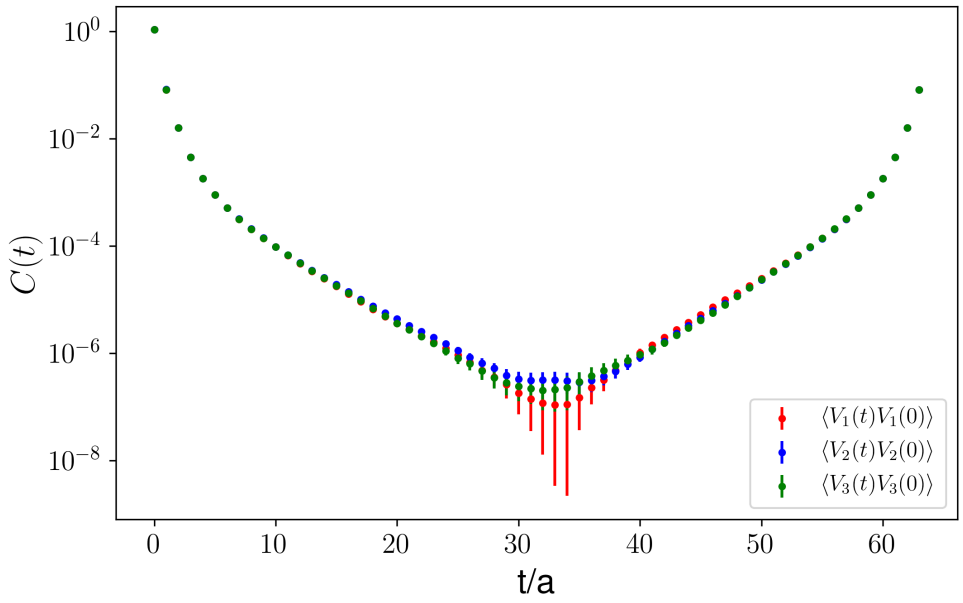


Figure 2.4: The three diagonal entries of the connected vector correlator, calculated on 30 gauge configurations from the E5 set with 384 stochastic sources. The shading indicates the one-sigma regions of the fits to the respective plateaus. There is good agreement inside the error bars, and the noise is reduced significantly compared to figure 2.3.

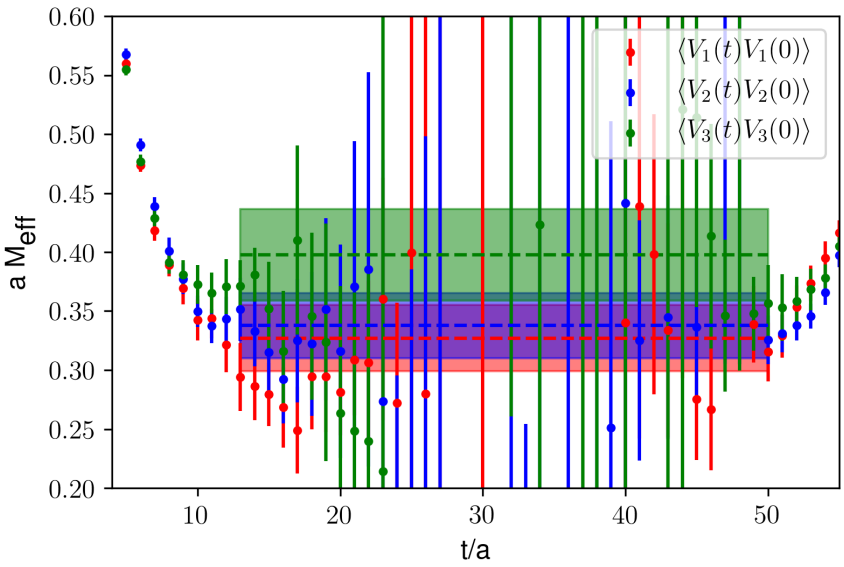


Figure 2.5: Effective masses for the three diagonal components of the connected vector correlator by time, calculated on 30 gauge configurations from the E5 set with 48 stochastic sources. The estimates agree with each other inside the error bars.

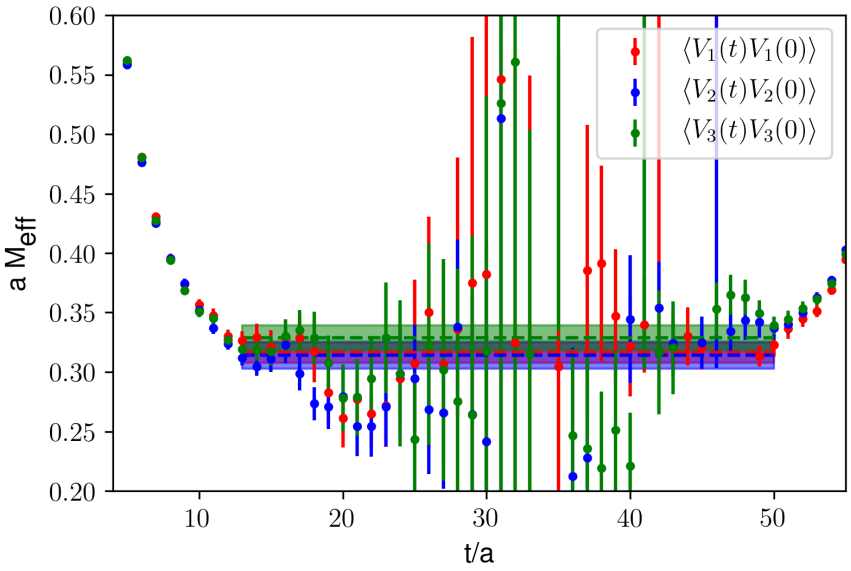


Figure 2.6: Effective masses for the three diagonal components of the connected vector correlator by time, calculated on 30 gauge configurations from the E5 set with 384 stochastic sources. Compared to figure 2.5, the error on the mass plateau is greatly reduced. The estimates agree inside the error bars, both with each other and with the value of am_ρ calculated on the E5 set in [47].

lation invariance

$$\langle O_a(t)\overline{O}_b(t_0)\rangle = \sum_{t_0}\langle O_a(t+t_0)\overline{O}_b(t_0)\rangle. \tag{2.21}$$

Further, for the traces in the pseudoscalar correlator, $\mathrm{tr}[\gamma_5 D^{-1}(t,t)]$, the gamma matrices can be chosen such that $\gamma_5^\dagger = \gamma_5$ holds, in which case the trace is real. So we can write

$$\mathrm{tr}[\gamma_5 D^{-1}(t,t)] = \frac{1}{2}\frac{1}{N_r}\sum_r \left(\mathrm{tr}[\eta^{\dagger(r)}\gamma_5 D^{-1}(t,t)\eta^{(r)}] + \mathrm{tr}[(\gamma_5 D^{-1}(t,t)\eta^{(r)})^\dagger \eta^{(r)}]\right), \tag{2.22}$$

removing all stochastic noise from the imaginary part of the trace. Since $1-\Box(t)$ is also hermitian, the same can be done with $\mathrm{tr}[(1-\Box(t))\gamma_5 D^{-1}(t,t)]$ for the calculation of C_{rest} and C_{cross}.

For the vector correlator, γ_i can be chosen such that $\gamma_i^\dagger = -\gamma_i$, making the trace anti-hermitian. Thus, calculating

$$\mathrm{tr}[\gamma_i D^{-1}(t,t)] = \frac{1}{2}\frac{1}{N_r}\sum_r \left(\mathrm{tr}[\eta^{\dagger(r)}\gamma_i D^{-1}(t,t)\eta^{(r)}] - \mathrm{tr}[(\gamma_i D^{-1}(t,t)\eta^{(r)})^\dagger \eta^{(r)}]\right), \tag{2.23}$$

removes all stochastic noise from the real part of the trace without requiring additional Dirac inversion or other relevant additional computational costs. This was implemented with

$\mathbb{Z}(2) \times \mathbb{Z}(2)$ noise sources for both disconnected observables.

Dilution

To optimize the signal-to-noise ratio for the observables, some dilution schemes [6] for the stochastic sources were implemented and tested. In a stochastic dilution scheme, stochastic sources covering the whole volume are split into sources that only have support on a part of it. For example, sources may be separated into their three color and four spin components,

$$\begin{aligned} \eta^{(r)} &= \sum_{d=0}^{N_d-1} \eta^{(r)(d)} \\ \eta^{(r)} &= \begin{bmatrix} \mathcal{D}(c1,s1) \\ 0 \\ \dots \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \mathcal{D}(c2,s1) \\ \dots \\ 0 \end{bmatrix} + \dots \end{aligned} \tag{2.24}$$

$$D^{-1}(m|n) = \frac{1}{N_r}\frac{1}{N_d}\sum_r \sum_d \phi^{(r)(d)}(m)\eta^{(r)(d)}(n)^\dagger\,, \tag{2.25}$$

or into even and odd points $(n_0 + n_1 + n_2 + n_3) \bmod 2$ on the lattice. The solutions to the Dirac equation for each of these split sources yield an unbiased estimator for the propagator. In the limit of diluting sources completely, so they only have support on a single point, all stochastic noise is eliminated as one is effectively performing a non-stochastic calculation with point sources.

Whether dilution is more cost-effective than increasing the number of stochastic sources depends on the observable and lattice setup under consideration. For the connected correlators, we tested the performance of dilution in the even-odd positions on the lattice, in the color components, and in a combination of both, for the connected vector correlator. Since we already make use of linked spin-diluted sources as explained above, further spin dilution for the connected correlators is not possible in this implementation. None of these dilution schemes yielded a significant gain compared to using more stochastic sources in the simulations on our data set. Figure 2.7 shows the ratio between the errors on the V_{11} component of the vector correlator, calculated with different dilution schemes, relative to applying no dilution scheme at all. It is an equal cost comparison with 96 sources, meaning the baseline used 96 normal wall sources, the color dilution used 32 sources in every color, the even-odd

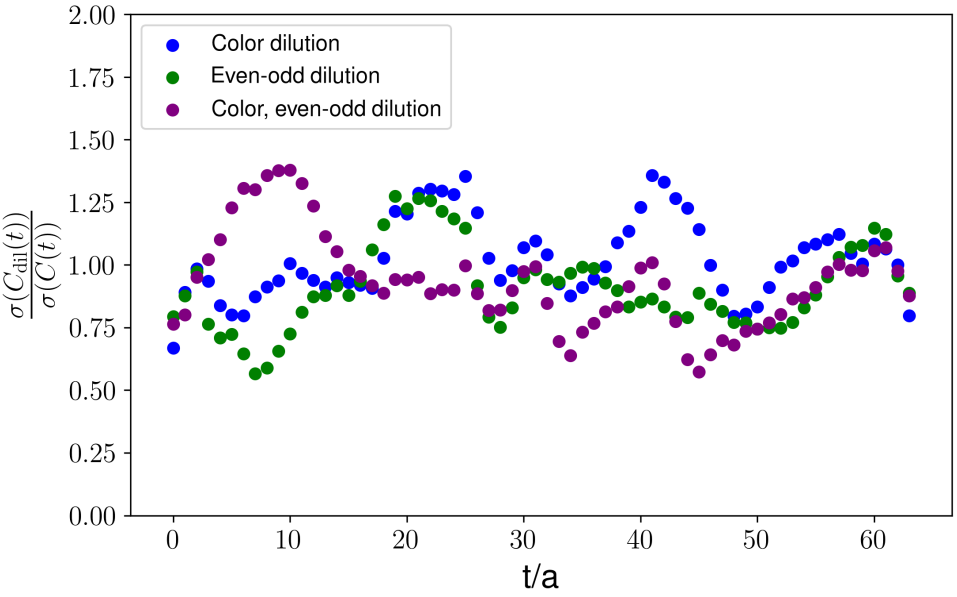


Figure 2.7: Error ratios $\frac{\sigma(C_{\mathrm{dil}}(t))}{\sigma(C(t))}$ for the connected vector correlator, for different dilution schemes of equal cost, relative to a calculation with 96 $\mathbb{Z}(2) \times \mathbb{Z}(2)$ sources and no dilution, calculated on 30 configurations from the E5 data set.

dilution 48 in both parities and the color-even-odd dilution 16 for every parity split color component. None of the dilution schemes demonstrate an error ratio below 0.9 relative to the baseline. We conclude that these dilution schemes are cost-neutral for our implementation of the connected vector correlator.

Since the disconnected correlators receive contributions from all time slices, $\mathbb{Z}(2) \times \mathbb{Z}(2)$ volume sources with support on all time slices might be expected to be a viable alternative to wall sources. In this framing, wall sources are time-diluted volume sources. An equal-cost comparison between $\mathbb{Z}(2) \times \mathbb{Z}(2)$ wall sources and volume sources was carried out for the disconnected pseudoscalar correlator. Volume sources significantly outperformed wall sources. Spin diluting the volume sources had no notable effect on precision. Neither did dilution in even-odd lattice position and color. This is pictured in figures 2.8 and 2.9, plotting the correlators and relative errors over time for the various dilution schemes. For the disconnected vector-vector correlator, which is significantly noisier, not enough computational resources were available to carry out a full time-series comparison of dilution schemes as for the pseudoscalar. However, since the error on the disconnected vector-vector correlator is constant in time, averaging the error on the correlator over all times still provides a useful estimate. The following table shows the time-averaged error on the disconnected vector correlator calculated with 30 gauge configurations from the E5 set, on 640 $\mathbb{Z}(2) \times \mathbb{Z}(2)$ stochastic sources, relative

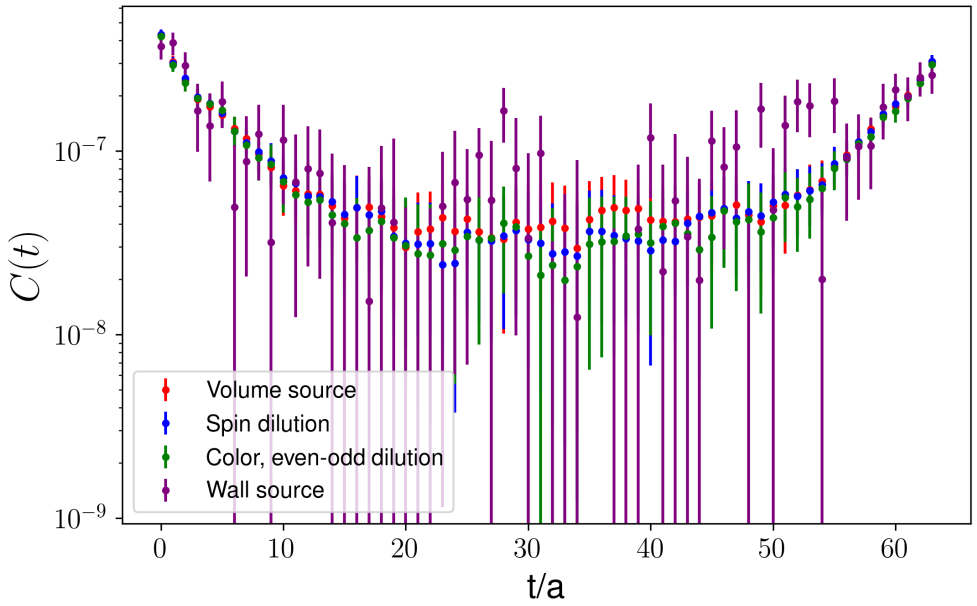


Figure 2.8: The disconnected pseudoscalar correlator calculated with 640 $\mathbb{Z}(2) \times \mathbb{Z}(2)$ sources. The different colors correspond to wall sources, volume sources, and volume sources diluted by spin and even-odd position and color, respectively. 30 configurations from the E5 data set were used. Wall sources lead to higher errors. Other dilution schemes appear to have little effect.

to the baseline of the volume sources.

Source type	$\overline{\sigma}(C)$ for 640 sources, divided by $\overline{\sigma}(C)$ for volume sources
No dilution	1
Spin diluted	1.18
Color, even-odd dilution	1.49
Wall source	69.97

Plain volume sources outperform wall sources by almost two orders of magnitude and marginally outperform the other diluted volume sources. Thus, volume sources are used for all subsequent calculations of disconnected observables in this work.

2.3.2 Testing Primme

To ensure the accuracy of the Laplacian implementation, a test of the Primme solver was carried out. One can form a relation between the trace of the spatial Laplacian operator to

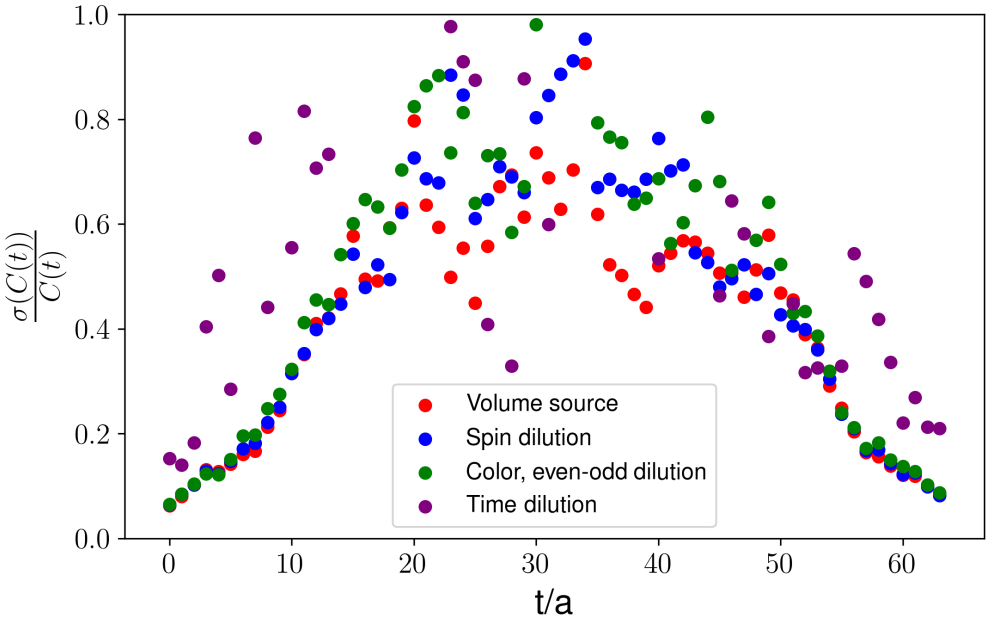


Figure 2.9: Relative errors $\frac{\sigma(C(t))}{C}$ of the disconnected pseudoscalar correlators from figure 2.8. Time dilution significantly decreases precision.

the fourth power, and the average spatial plaquette on a configuration

$$\begin{aligned} \nabla^2_{m,n} &= -6\,\delta_{m,n} + \sum_{j=1}^{3} \left(U_j(m,t)\delta_{m+j,n} + U_j^\dagger(m-j,t)\delta_{m-j,n} \right) \\ \Rightarrow \mathrm{tr}[\nabla^4] &= \sum_n \left(4 \sum_{\{j,k=1|j\neq k\}}^{3} \mathrm{tr}[P_{j,k}(n)] + 2\times 6^4\,\mathrm{tr}_U[\mathbb{1}] + 6\times 11 \mathrm{tr}_U[\mathbb{1}] \right) . \end{aligned} \tag{2.26}$$

Here, we assume that the spatial lattice satisfies $L > 4$ in every direction, otherwise, additional terms from Wilson lines wrapping around the lattice appear. $\mathrm{tr}[\nabla^4]$ can be calculated by summing the fourth powers of every eigenvalue found by PRIMME, while the average plaquette can be calculated directly from the configuration. The implementation was found to satisfy this relation for a configuration chosen from the E5 ensemble.

2.4 Results

2.4.1 Connected correlation functions

Figures 2.10 and 2.11 compare the vector correlator calculated with distillation-low-mode-averaging and without it, on 30 configurations drawn from the E5 and A5 ensembles respectively, with four independent color and even-odd diluted wall sources and 640 Laplacian

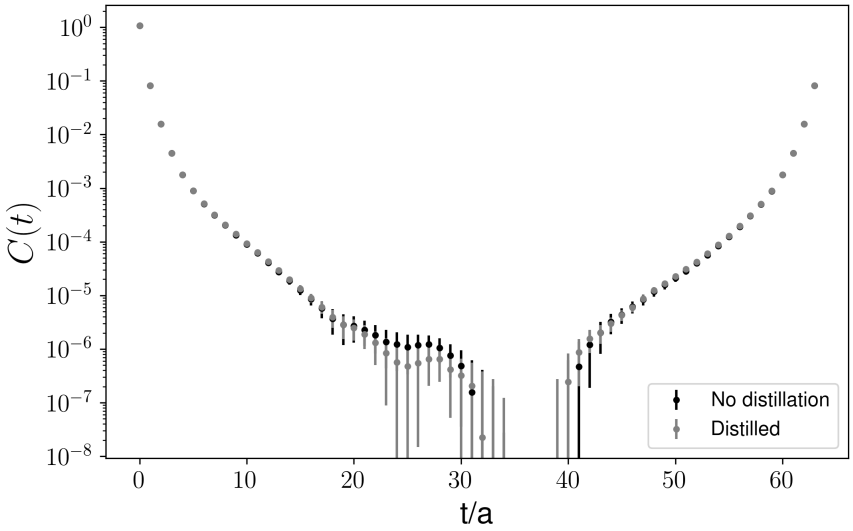


Figure 2.10: The connected vector correlator, calculated on 30 gauge configurations from the E5 data set with 24 stochastic wall sources using distillation low-mode averaging with 160×4 Laplacian eigenvectors. The correlator without distillation is shown in black. There is good agreement inside the error bars.

modes. Figures 2.12 and 2.13 show the same for the pseudoscalar correlator. The results are in good agreement inside the error bars. Figure 2.14 shows the three individual contributions to the total distilled result in the E5 data set. As one would hope, the C_{rest} contribution quickly falls off in importance relative to C_{dist}. However, C_{cross} does not fall off to the same extent and stays close to C_{dist} in size even in the long-distance part of the correlator. This is a fundamental problem for the technique. If C_{cross} does not fall off quickly, the method is not efficient in terms of the computing costs invested. This would seem to indicate that our guess was incorrect, and the spectra of the distillation and Dirac operators are not similar enough to make this idea less numerically expensive after all. To show this clearly, plots 2.15 and 2.16 graph $\frac{C_{dist}}{C}$, the contribution of the distillation subspace to the whole correlator on the two data sets for different numbers of distillation modes. At low numbers of modes, $\frac{C_{dist}}{C}$ scales linearly with the size of the subspace. But this scaling quickly falls off, and even a computational investment of 1280 distillation modes falls short of the 0.8 mark on both ensembles. 2.17 and 2.18 show the same for the connected pseudoscalar correlator. Here, the total contribution of the distilled subspace to the observable is even smaller than in the case of the vector correlator. The A5 ensemble also exhibits worse scaling overall than the E5 ensemble. We speculate that this is due to the larger size of the physical volume increasing the number of distillation modes below any given threshold [48]. Figures 2.20 and 2.21 plot the relative errors $\frac{\sigma(C(t))}{C}$ of the total correlators with and without the distillation technique

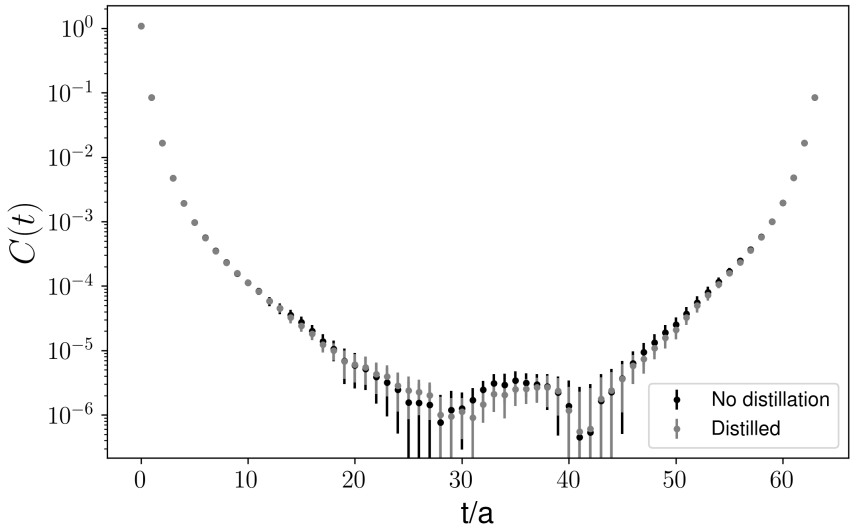


Figure 2.11: The connected vector correlator, calculated on 30 gauge configurations from the A5 data set with 24 stochastic wall sources using distillation low-mode averaging with 160×4 Laplacian eigenvectors. The correlator without distillation is shown in black. There is good agreement inside the error bars.

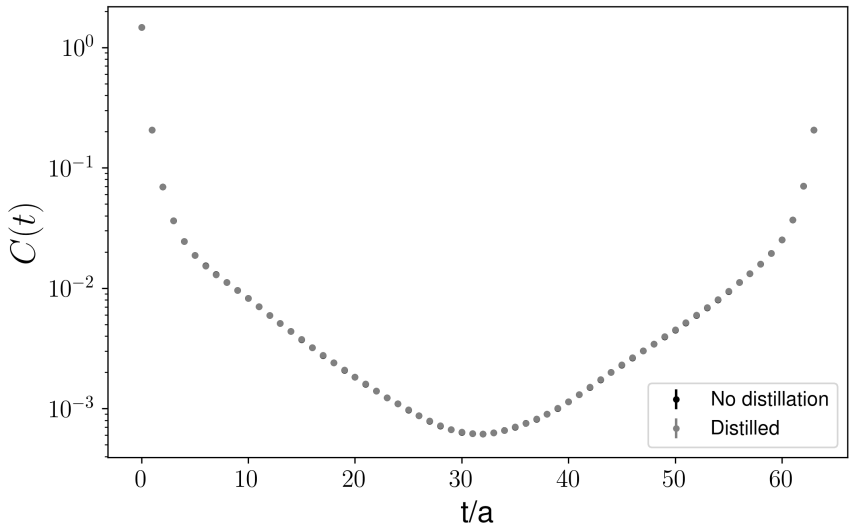


Figure 2.12: The connected pseudoscalar correlator, calculated on 30 gauge configurations from the E5 data set with 24 stochastic wall sources using distillation low-mode averaging with 160×4 Laplacian eigenvectors. The correlator without distillation is shown in black. There is good agreement inside the error bars, which are too small to be visible.

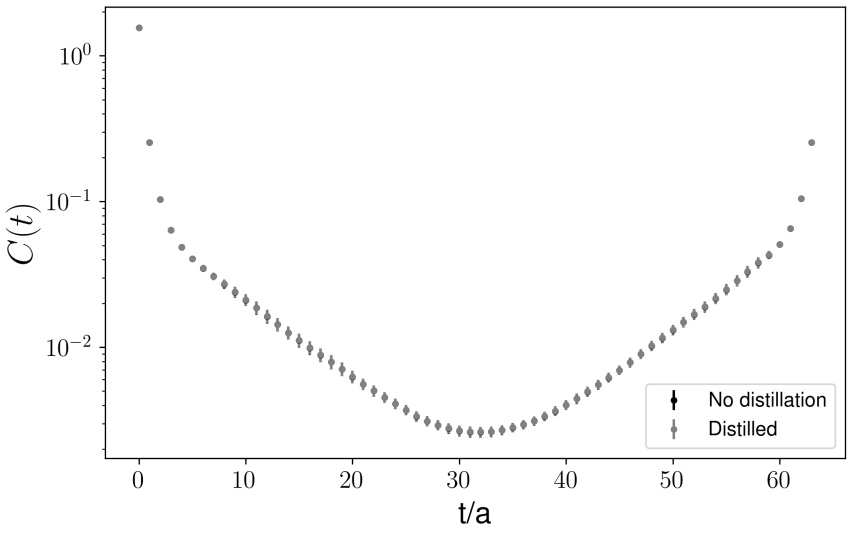


Figure 2.13: The connected pseudoscalar correlator, calculated on 30 gauge configurations from the A5 data set with 24 stochastic wall sources using distillation low-mode averaging with 160×4 Laplacian eigenvectors. The correlator without distillation is shown in black. There is good agreement inside the error bars, which are too small to be clearly visible.

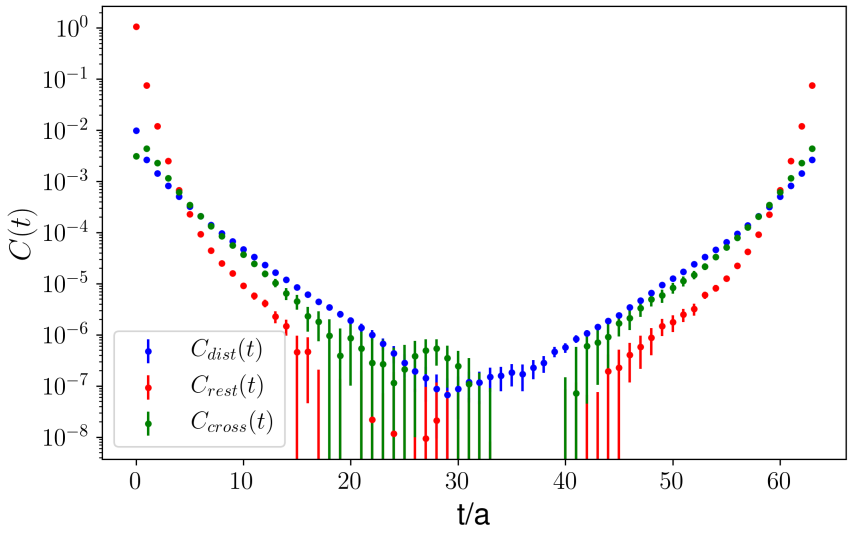


Figure 2.14: The separate distilled, rest, and cross contributions to the connected vector correlator, calculated on 30 gauge configurations from E5 data set with 24 stochastic sources and 160×4 Laplacian modes. The rest contribution declines quickly in importance, as hoped, but the mixed contribution does not.

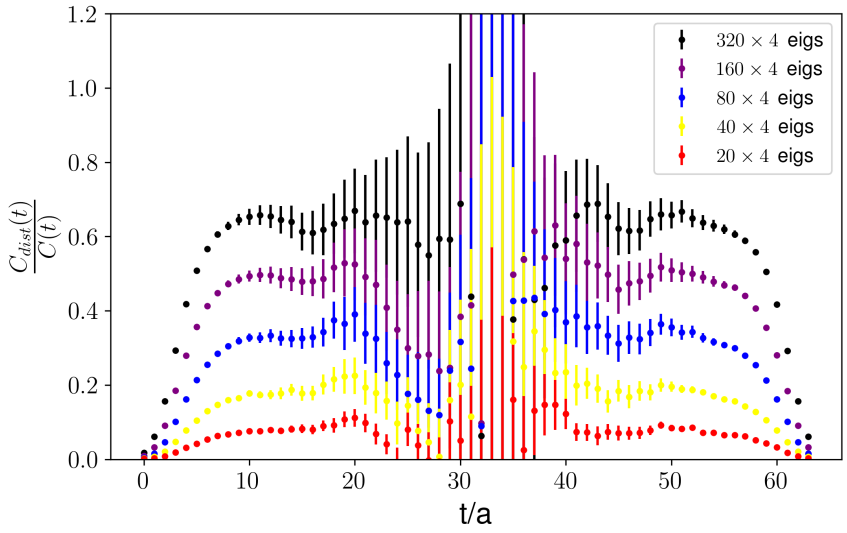


Figure 2.15: Fractional contribution $\frac{C_{dist}}{C}$ of the distilled correlator to the total vector correlator for different numbers of distillation eigenmodes, calculated on 30 gauge configurations taken from the E5 data set with 24 stochastic sources. The contribution scales linearly with mode count at the start, but quickly drops off. A prohibitive number of eigenmodes seems necessary to capture > 0.75 of the total correlator.

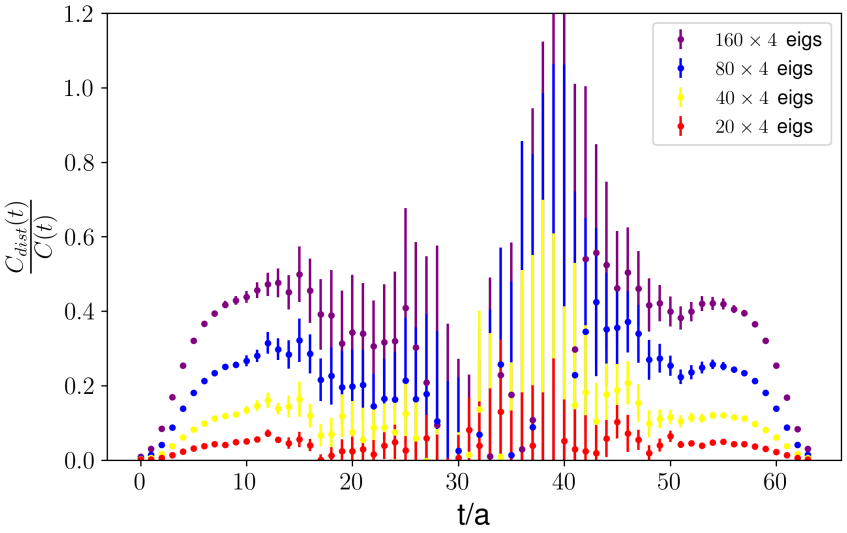


Figure 2.16: Fractional contribution $\frac{C_{dist}}{C}$ of the distilled correlator to the total vector correlator for different numbers of distillation eigenmodes, calculated on 30 gauge configurations taken from the A5 data set with 24 stochastic sources. Compared to figure 2.15, the overlap is smaller. This is likely due to the larger physical lattice volume in this configuration set.

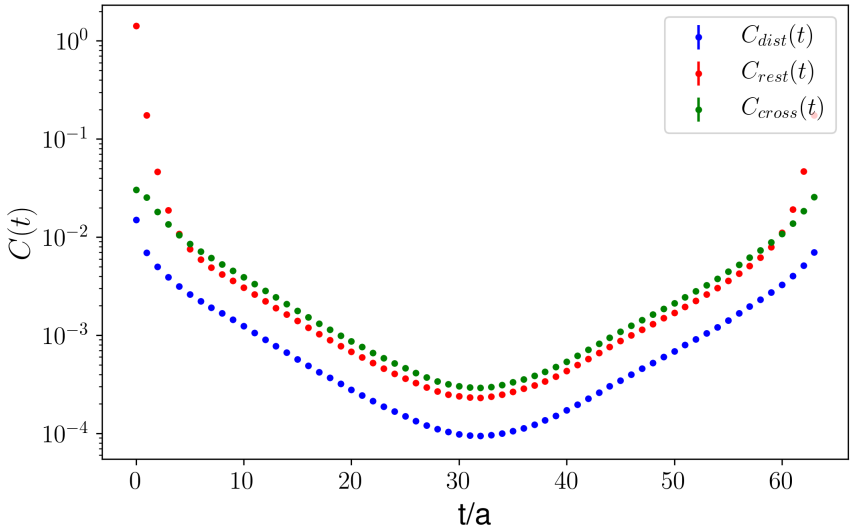


Figure 2.17: The separate distilled, rest, and cross contributions to the connected pseudoscalar correlator, calculated on 30 gauge configurations from E5 data set with 24 stochastic sources and 160×4 Laplacian modes. The distilled contribution stays subdominant even at large times. Notably, the three contributions appear to follow a similar exponential curve after the first five time steps.

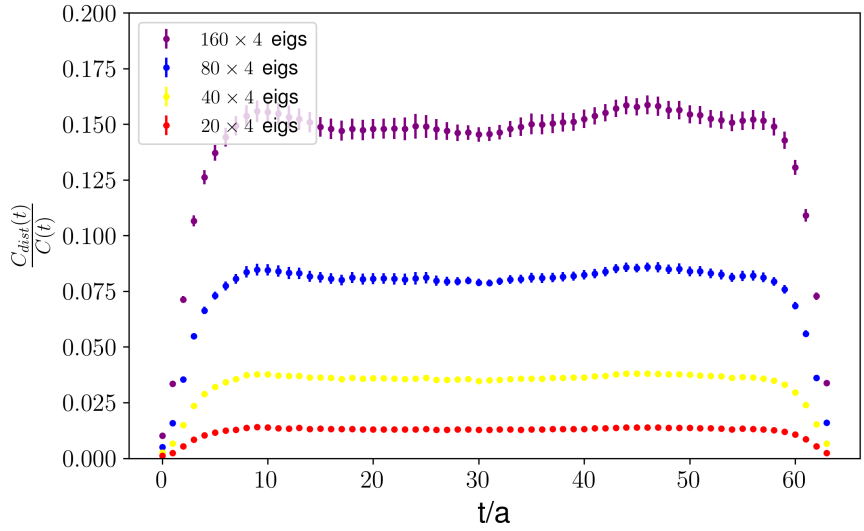


Figure 2.18: Fractional contributions of the distilled correlator $\frac{C_{dist}}{C}$ to the total connected pseudoscalar correlator for different numbers of distillation eigenmodes, calculated on 30 gauge configurations from the E5 data set. The cost of capturing even > 0.5 of the total correlator is completely prohibitive.

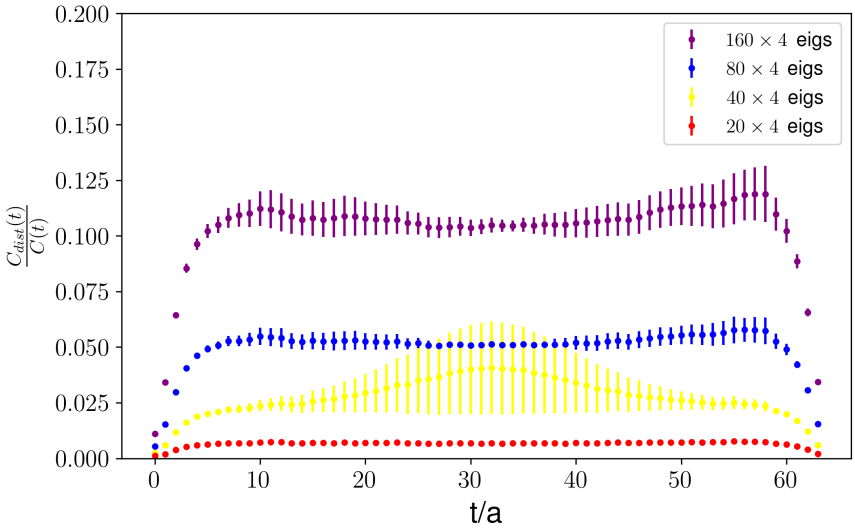


Figure 2.19: Fractional contributions of the distilled correlator $\frac{C_{dist}}{C}$ to the connected pseudoscalar for different numbers of distillation eigenmodes, calculated on 30 gauge configurations from the A5 data set. As with the vector correlator, the overlap is smaller compared to E5.

applied, for 160 spin-degenerate distillation modes, 160×4. As might be expected from the analysis above, there is a factor of about $\frac{1}{\sqrt{2}}$ between the two, which is consistent with reducing the stochastic noise on approximately half of the total correlator to zero, since

$$\sigma(C) = \sqrt{\sigma^2(C_{dist}) + \sigma^2(C - C_{dist})}\,. \tag{2.27}$$

This means that if the contribution $\sigma(C - C_{dist})$ to the total stochastic error budget simply scaled linearly with the size of $C - C_{dist}$, one would expect a $\frac{1}{\sqrt{2}}$ factor for $\frac{C_{dist}}{C} = 0.5$, which is what we observe. Similar results are observed for the pseudoscalar correlator, except that the scaling of $\frac{C_{dist}}{C}$ with the number of modes is even more unfavorable. Calculating enough modes to reach $\frac{C_{dist}}{C} = 0.5$ thus proved impractical. 2.23 and 2.19 instead picture the precision gain for the pseudoscalar for 160×4 eigenmodes, with roughly $\frac{C_{dist}}{C} = 0.12$, $\frac{C_{dist}}{C} = 0.11$ respectively. As would be expected from equation 2.27, the relative errors shrink by less than 10%.

Computational costs

To compare computational costs for the connected contribution, table 2.2 shows the time needed to calculate the distillation eigenmodes with Primme, and C_{dist} with OpenQ*D, on a single configuration, for different numbers of eigenmodes. At low numbers of distillation

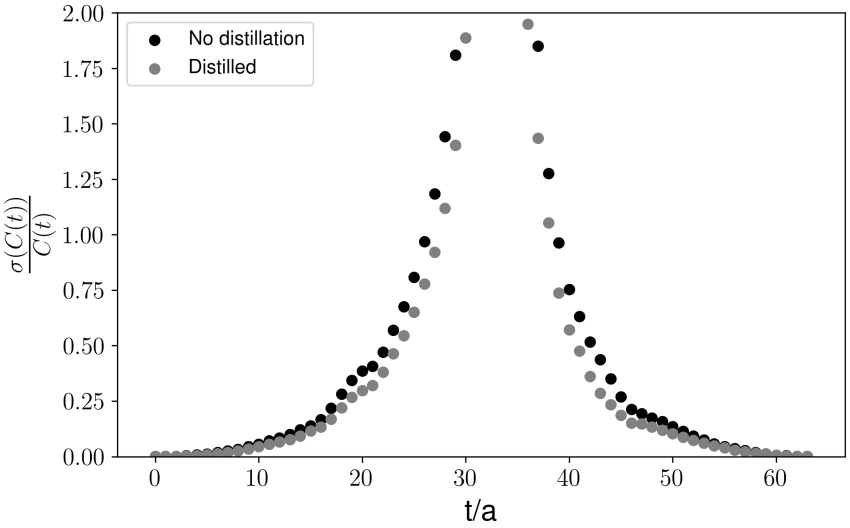


Figure 2.20: Relative error $\frac{\sigma(C(t))}{C(t)}$ for the vector correlator calculated with 160×4 eigenmodes and 24 stochastic sources, on 30 gauge configurations from the E5 data set. At larger times, the distilled method reduces the relative error by a factor of ca. $\frac{1}{\sqrt{2}}$. This is consistent with 2.15 showing the distilled correlator to contribute ca. 0.5 of the whole.

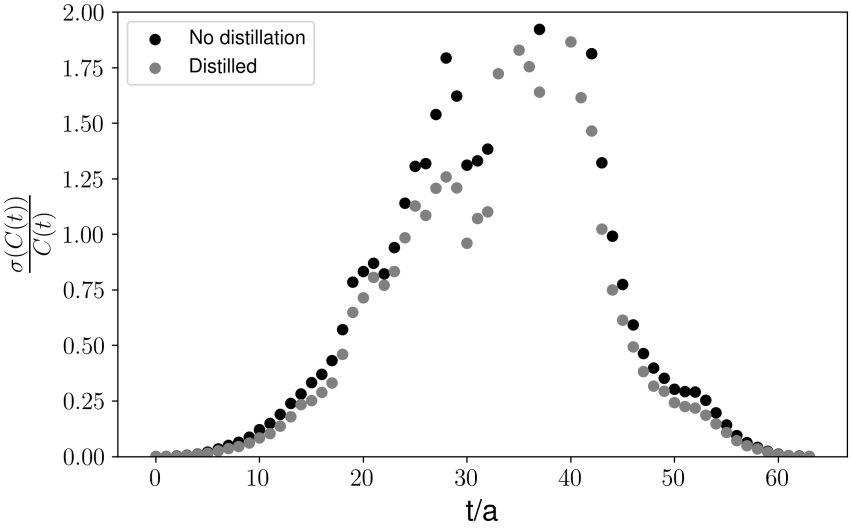


Figure 2.21: Relative error $\frac{\sigma(C(t))}{C(t)}$ for the vector correlator calculated with 160×4 eigenmodes and 24 stochastic sources, on 30 gauge configurations from the A5 data set. At larger times, the distillation method reduces the relative error by a factor of ca. $\frac{1}{\sqrt{2}}$. This is consistent with 2.16 showing the distilled correlator to contribute ca. 0.5 of the whole.

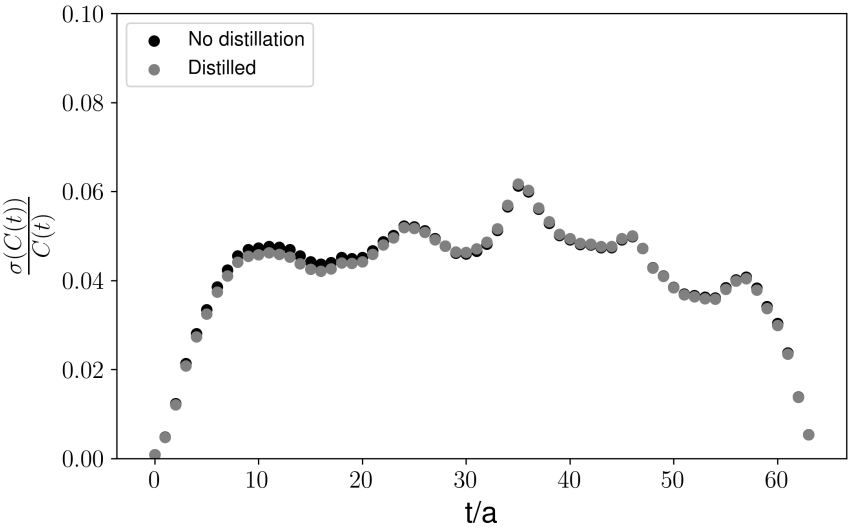


Figure 2.22: Relative error $\frac{\sigma(C(t))}{C(t)}$ for the pseudoscalar correlator, calculated with 160×4 eigenmodes and 24 sources, on 30 gauge configurations from the E5 data set. Consistent with the small contribution of ca. 0.15 the distilled correlator contributes to the whole, as shown in figure 2.18, the error reduction is barely visible even at large times.

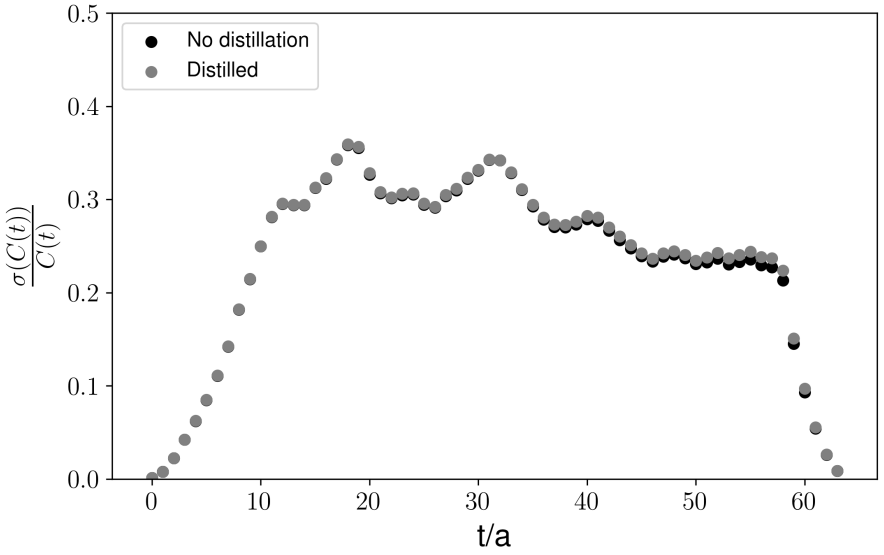


Figure 2.23: Relative precision $\frac{\sigma(C(t))}{C(t)}$ for the pseudoscalar correlator, calculated with 160×4 eigenmodes and 24 sources, on 30 gauge configurations from the A5 data set. Consistent with the small contribution of ca. 0.15 the distilled correlator contributes to the whole, as shown in figure 2.19, the error reduction is barely visible even at large times.

N_v	C_{dist} Dirac solver	Lap. solver	Lap. solver Ortho.	Lap. solver MatVec
20×4	1.18×10^2 s	4.63×10^1 s	3.5 s	3.11×10^1 s
40×4	2.53×10^2 s	1.04×10^2 s	1.08×10^1 s	6.87×10^1 s
80×4	4.84×10^2 s	2.43×10^2 s	4.18×10^1 s	1.52×10^2 s
160×4	1.12×10^3 s	7.77×10^2 s	2.51×10^2 s	3.46×10^2 s
320×4	2.82×10^3 s	2.14×10^3 s	1.14×10^3 s	7.40×10^2 s

Table 2.2: Computational costs of distillation for the connected correlators, on a single configuration taken from the E5 set.

modes, the Dirac solver dominates the computational cost. However, as the number of modes is increased, the time needed by the Primme Laplacian solver for orthogonalization increases quadratically. This represents another fundamental obstacle to the technique. Since a very large number of eigenmodes seems needed for C_{dist} to dominate C_{cross} in the correlator, the time needed to solve for all the eigenmodes of the Laplacian will eventually scale quadratically with the number of modes, potentially destroying the supposed advantage over normal low-mode-averaging, that individual modes take less time to calculate. If we were to use the computation time invested in distillation for additional stochastic sources instead, we would expect

$$t_{\frac{N_v}{4},l} + t_{N_v,D} + 2t_{N_R,D} = t_{\hat{N}_R}\,, \tag{2.28}$$

where N_R is the number of stochastic sources with distillation, $\hat{N}_R$ the number possible for the same cost without distillation, $t_{N_v,D}$ the time required to solve the Dirac equation for the given number of distillation modes N_v, $t_{N_R,D}$ the time required to do the same for the stochastic sources and $t_{\frac{N_v}{4},l}$ the time required to run the Laplacian solver for the given number of distillation modes N_v. The factor of 2 reflects that introducing distillation forces us to solve the Dirac equation for the stochastic sources twice.

Assuming roughly equal time needed to solve the equation for any kind of Dirac source, this implies

$$\hat{N}_R > 2N_R + N_v\,, \tag{2.29}$$

since the relative variance of the correlator scales as

$$\frac{\sigma^2(C)}{C} \propto \frac{1}{N_r}\,, \tag{2.30}$$

we would thus need

$$1 - \frac{C_{dist}}{C} \leq \frac{N_r}{\hat{N}_R} \leq \frac{N_r}{2N_r + N_v} = \frac{1}{\frac{N_v}{N_r} + 2}\,, \tag{2.31}$$

even in the limit of assuming the time needed to find the Laplacian modes is negligible, which

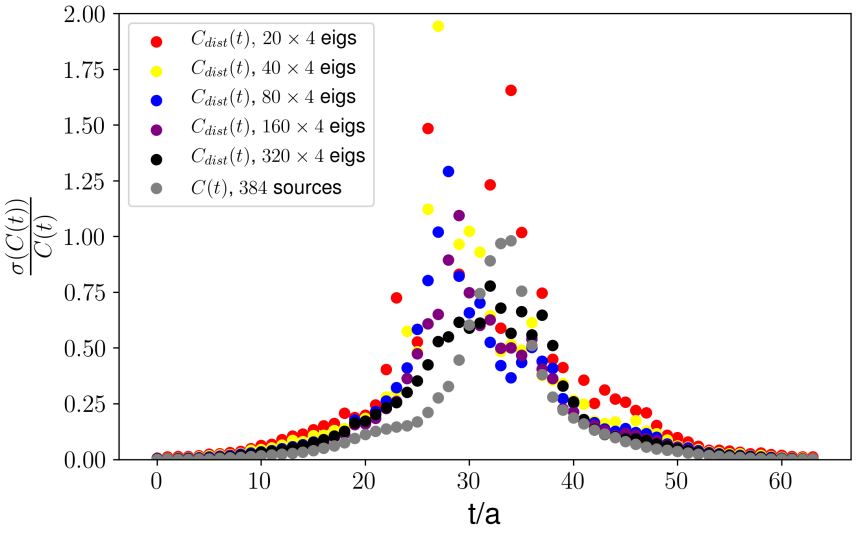


Figure 2.24: Relative error on the distilled vector correlator contribution calculated with different numbers of distillation eigenmodes, on 30 configurations taken from the E5 data set with 24 stochastic sources. The points in gray are the relative errors on the whole correlator calculated with 384 sources, providing an estimate for the total gauge noise.

is not the case. In our existing tests plotted in figures 2.15, 2.18, 2.16, 2.19, this relation is never fulfilled.

Gauge noise

The combined variance of the three individual contributions to the total vector correlator is larger than the variance of the whole correlator.

$$\sigma^2(C_{dist}) + \sigma^2(C_{rest}) + 2\sigma^2(C_{cross}) > \sigma^2(C) = \sigma^2\left(C_{dist} + C_{rest} + 2C_{cross}\right) \tag{2.32}$$

In particular, $\frac{\sigma(C_{dist})}{C_{dist}}$ is higher for smaller numbers of eigenmodes used. Figure 2.24 pictures the relative error of the distillation subspace contribution to the vector correlator over time, on the $E5$ configuration set, for different sizes of the subspace. The gray line is the error of the total vector correlator calculated without distillation, using 384 independent stochastic sources. This amount is sufficient to suppress the majority of the stochastic contribution to the total error budget, allowing it to serve as an effective baseline for the gauge noise alone.

As the size of the distillation subspace is increased, the gauge error on the subspace contribution shrinks, presumably reaching the original relative gauge error in the limit of a maximal distillation space. This suggests that there are cancellations in the gauge noise between the different contributions. In the next section, a way to exploit this correlation

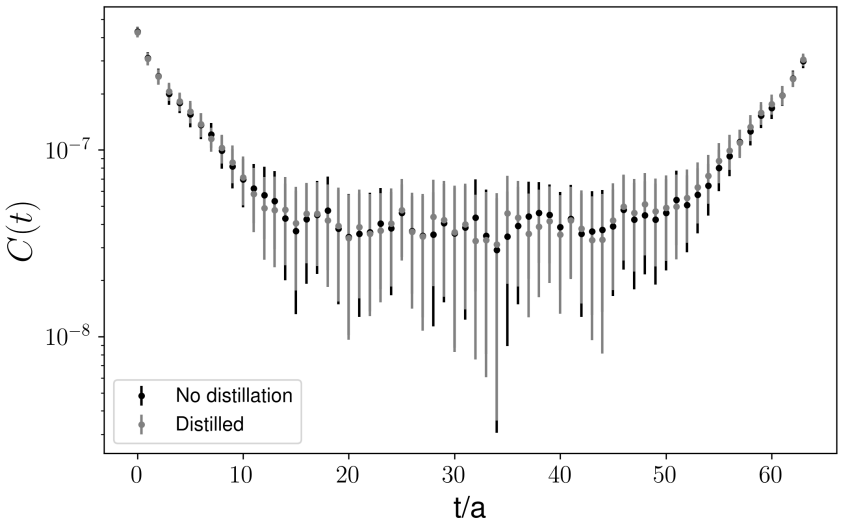


Figure 2.25: Comparison of the disconnected pseudoscalar correlator calculated with and without distillation using 20×4 Laplacian eigenmodes on every time slice, on 30 configurations taken from the E5 data set, with 640 stochastic volume sources. The results are in good agreement inside the error bars.

to increase the spectrum overlap between the distillation operator and the correlator will be explored.

2.4.2 Disconnected correlation functions

Figure 2.25 compares the disconnected pseudoscalar correlator calculated with distillation-low-mode averaging and without it. The results are in good agreement inside the error bars. Figure 2.26 shows the three individual contributions $C_{dist}, C_{rest}, C_{cross}$ to the total distilled result. As with the connected pseudoscalar correlator, the distilled contribution is far too small to be of relevance. The relative size of the distilled contribution for different numbers of eigenmodes is pictured in figure 2.27. The growth of the contribution with the number of eigenmodes used included seems far too small to justify the large computational investment required for calculating the Laplacian modes on every time slice.

On the computation and time budget available for this work, the disconnected vector correlator is too noisy to allow for most of the time series comparisons used for the other observables. The correlator becomes too small to distinguish from the noise background past $t/a = 0$. However, since the error on the disconnected observables stays constant in time, a simple analysis of the precision gained by distillation is still possible. Table 2.3 shows the error averaged over all time slices for different numbers of distillation modes. There is no relevant gain in precision, indicating that the distilled contribution is too small at these mode

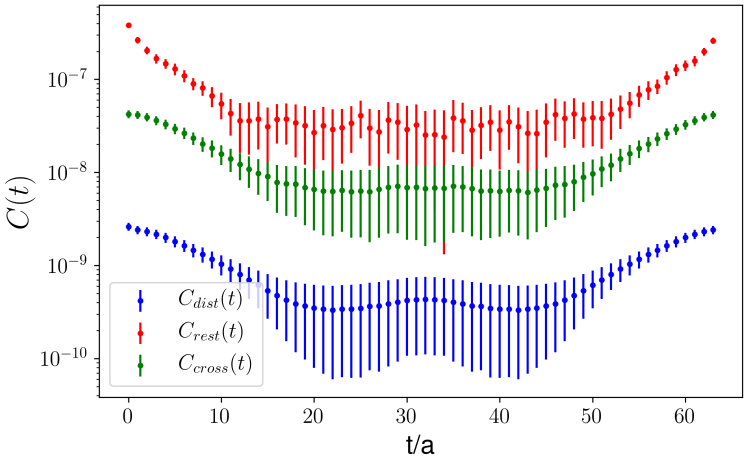


Figure 2.26: The separate distilled, rest, and cross contributions to the disconnected pseudoscalar correlator, calculated on 30 gauge configurations from the E5 data set with 640 stochastic volume sources and 20×4 Laplacian modes on every time slice. The distilled contribution stays subdominant even at large times.

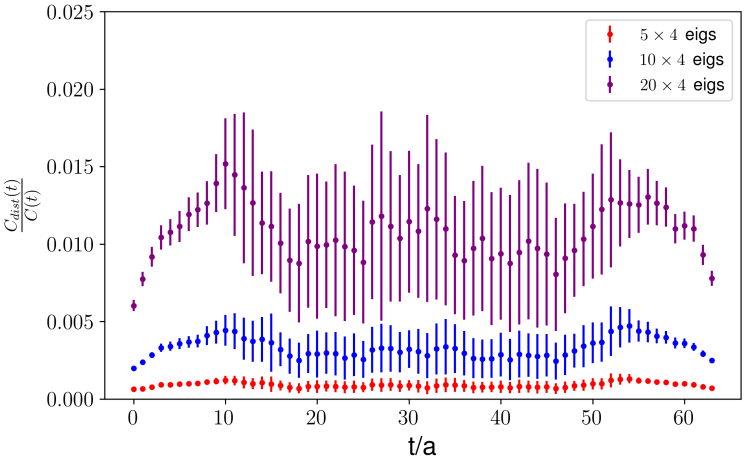


Figure 2.27: Fractional contributions of the distilled correlator $\frac{C_{dist}}{C}$ to the disconnected pseudoscalar for 5×4, 10×4 and 20×4 distillation eigenmodes on every time slice, calculated on 30 gauge configurations from the E5 data set, with 640 stochastic sources. Compared to the connected observables, the distilled contribution is even smaller here, barely reaching 0.01.

counts to provide a benefit.

N	$\frac{1}{64}\sum_{t=0a}^{64a}\sigma\left(C_{VV}(t)\right)$	$\frac{1}{32}\sum_{t=16a}^{48a}\sigma\left(C_{VV}(t)\right)$
0	$7.83\times10^{-10}\pm1.12\times10^{-11}$	$7.91\times10^{-10}\pm1.67\times10^{-11}$
5×4	$7.70\times10^{-10}\pm1.15\times10^{-11}$	$7.81\times10^{-10}\pm1.77\times10^{-11}$
10×4	$7.60\times10^{-10}\pm1.15\times10^{-11}$	$7.68\times10^{-10}\pm1.78\times10^{-11}$
20×4	$7.63\times10^{-10}\pm1.09\times10^{-11}$	$7.69\times10^{-10}\pm1.62\times10^{-11}$

Table 2.3: The average error on the disconnected vector correlator, calculated on 30 gauge configurations taken from the E5 data set, for different numbers of distillation modes. The distillation fails to provide a gain in precision visible at $> 1\sigma$, even when the average is restricted to the second half of the correlator to give higher modes time to decay.

2.5 Rescaling

While the low-lying spectrum of the distillation operator does not appear to contribute sufficiently to our observables, one could still hope to salvage the idea behind the distillation low-mode averaging technique by using a slightly different operator, related to the distillation operator through some simple transformation that does not substantially increase the computational resources necessary to calculate its eigenspectrum.

Perhaps the simplest such transformation is a rescaling. Instead of the distillation operator $\Box$, $\alpha\Box$ with $\alpha \in \mathbb{R}^+$ could be used instead. Looking at the plot 2.17 for the connected pseudoscalar, one may be tempted to think such a rescaling could significantly increase the contribution of the distilled subspace to the observable. It appears as if the distilled contribution is following the same exponential behavior as the other contributions, just shifted by a prefactor. This could be taken as an indicator that the three contributions actually consist of the same spectrum of particle eigenstates E_i, making the gauge noise on them correlated.

2.5.1 Connected correlators

Figures 2.28 and 2.29 show the C_{dist} and $C_{rest} + C_{cross}$ contributions to the connected pseudoscalar from 30 different gauge configurations on a single time slice. There is indeed a visible correlation in the data. Thus, setting $\alpha > 1$ should subtract this shared spectrum and decrease $C_{rest} + C_{cross}$ as well as $\sigma(C_{rest})$ and $\sigma(C_{cross})$, increasing precision. At $t/a = 10$, there is no clearly visible correlation for the connected vector correlator, see figures 2.30, 2.31, though the shape for the A5 data set does look suggestive. By $t/a = 43$, after more of the higher lying modes have decayed, some more hints of correlation become visible in the vector correlator, see figures 2.34 and 2.35. Correlation for the pseudoscalar remains very

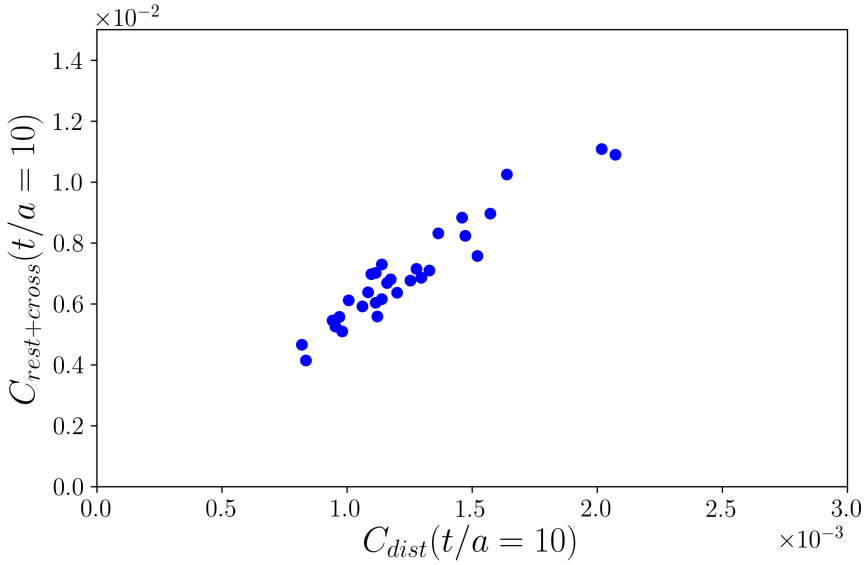


Figure 2.28: The stochastic and distilled contribution to the connected pseudoscalar correlator at $t/a = 10$, calculated with 160×4 distillation modes, for 30 configurations drawn from the E5 data set. There is a clear correlation in the data. Note that since the distilled contribution is much smaller, the axes have different scales.

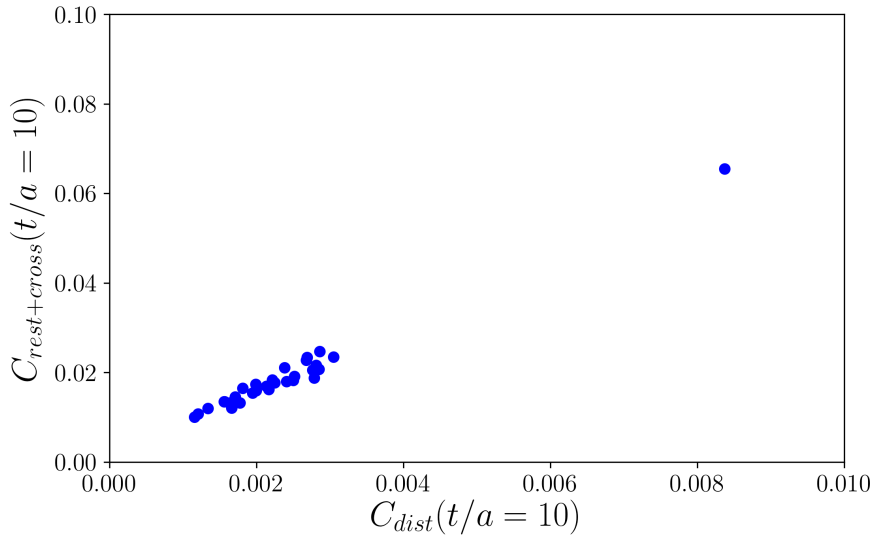


Figure 2.29: The stochastic and distilled contribution to the connected pseudoscalar correlator at $t/a = 10$, calculated with 160×4 distillation modes, for 30 configurations drawn from the A5 data set. There is a clear correlation in the data. Note that since the distilled contribution is much smaller, the axes have different scales. Despite its peculiar deviation from the rest, there appears to be nothing overtly wrong with the outlier point.

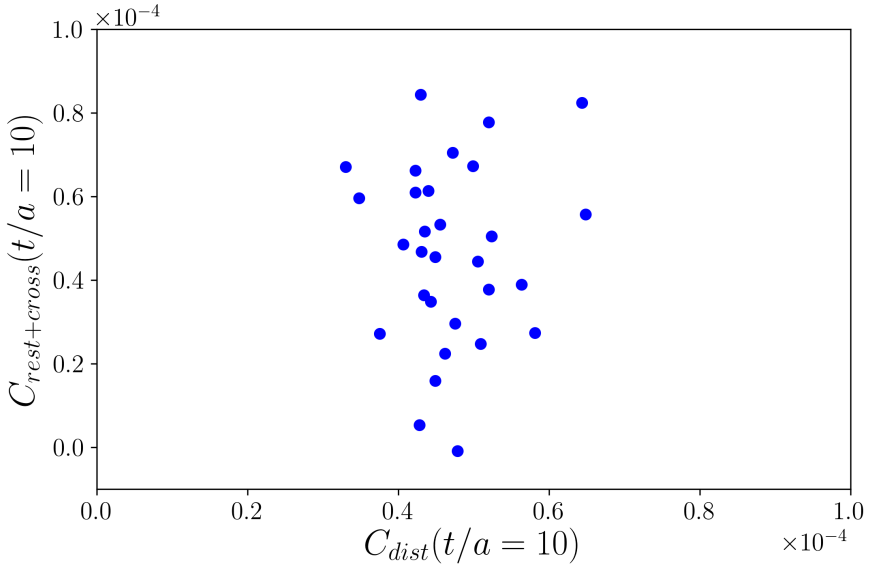


Figure 2.30: The stochastic and distilled contributions to the connected vector correlator at $t/a = 10$, calculated with 160×4 distillation modes, for 30 configurations drawn from the E5 data set. To the bare eye, there is no correlation visible in the data.

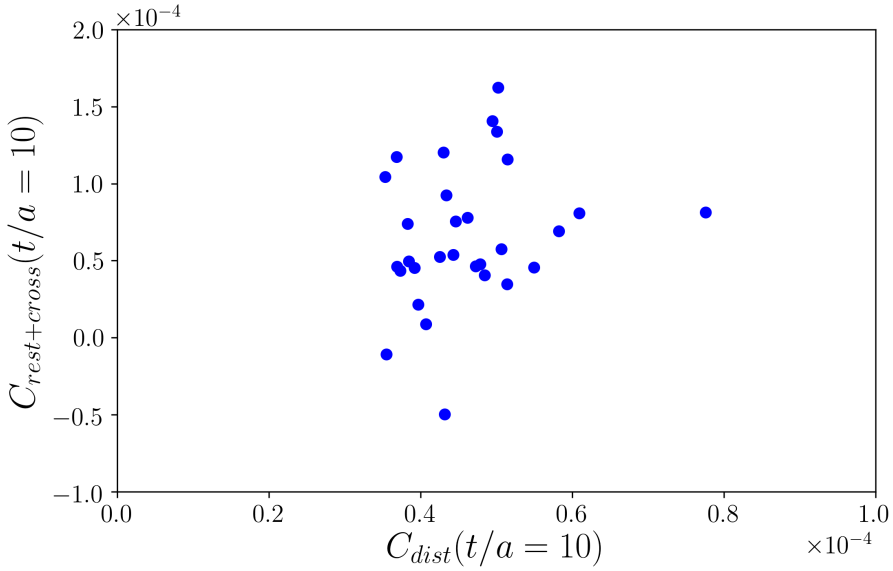


Figure 2.31: The stochastic and distilled contributions to the connected vector correlator at $t/a = 10$, calculated with 160×4 distillation modes, for 30 configurations drawn from the A5 data set. It is difficult to discern with the bare eye whether the data shows a correlation or not.

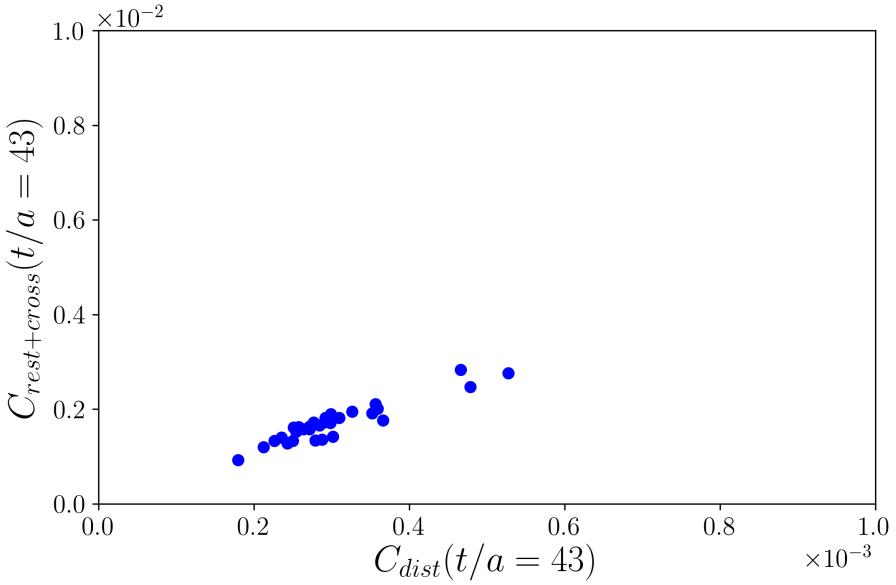


Figure 2.32: The stochastic and distilled contributions to the connected pseudoscalar correlator at $t/a = 43$, calculated with 160×4 distillation modes, for 30 configurations drawn from the E5 data set. There is a clear correlation in the data.

strong, see figures 2.32 and 2.33. We might thus expect low-mode averaging with $\alpha\Box, \alpha > 1$ to improve the spectrum overlap for both the connected vector and pseudoscalar correlators, with stronger effects for the latter than the former. For the connected correlators, inserting $1 - \alpha\Box + \alpha\Box$ instead of $1 - \Box + \Box$ into equation 2.4 means we need to calculate

$$\begin{aligned} \phi^{(k)} &:= \alpha D^{-1}(t,t')v^{(k)}(t') & \phi^{(r)} &:= D^{-1}(t,t')\eta^{(r)}(t') \\ \overline{\phi}^{(r)} &:= D^{-1}(t,t')\left(1 - \alpha\Box(t')\right)\eta^{(r)}(t')\,. \end{aligned} \tag{2.33}$$

$\phi^{(k)}$ is only rescaled, but $\overline{\phi}^{(r)}$ needs to be calculated again for every different value of α used.

Figures 2.36 and 2.37 show correlation scatter plots for the vector correlator calculated with 160 and 80 distillation modes and different α. How much the distilled contribution can be scaled up before the correlation disappears completely varies with the number of eigenvectors used. The more eigenvectors there are, the less exploitable correlation there is. This trend continues for 40 and 20 eigenvectors, and also holds for the pseudoscalar correlator, as pictured in figures 2.39 and 2.40. The pseudoscalar correlator has a greater amount of correlation to exploit than the vector correlator, so a far higher multiplier α can be deployed for it. Figures 2.42 and 2.43 show relative precision gains for the vector and pseudoscalar correlators for different multipliers, at 160×4 and 20×4 distillation modes respectively. As expected, bigger multipliers further improve precision, since they reduce the size of the

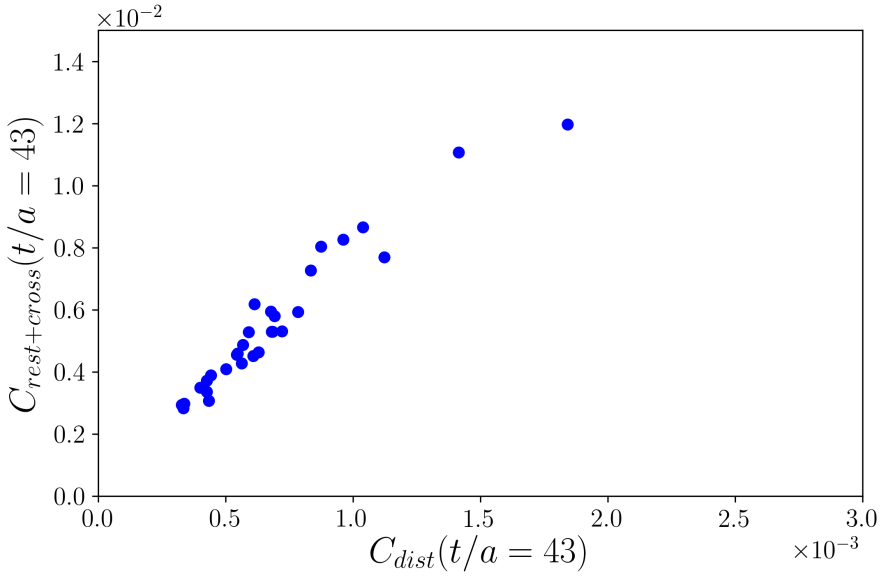


Figure 2.33: The stochastic and distilled contributions to the connected pseudoscalar correlator at $t/a = 43$, calculated with 160×4 distillation modes, for 30 configurations drawn from the A5 data set. There is a clear correlation in the data.

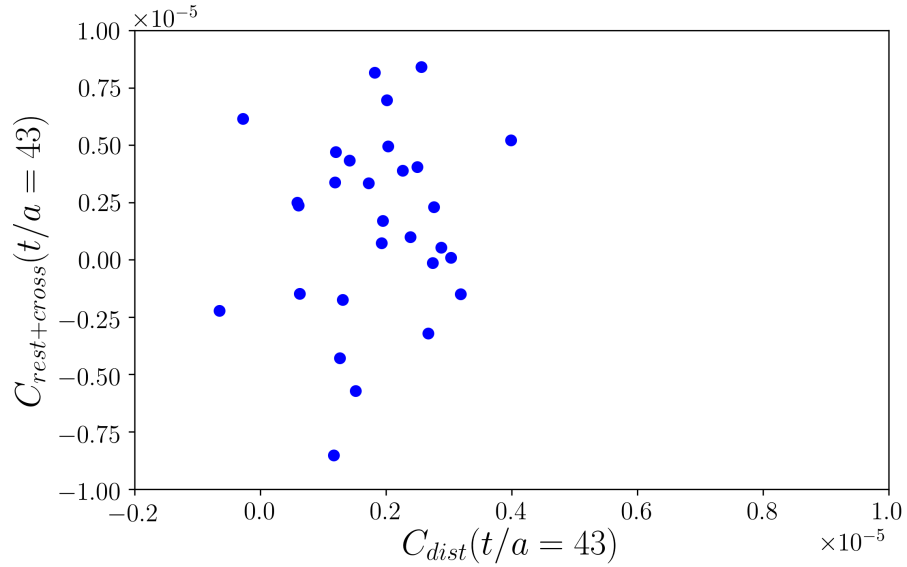


Figure 2.34: The stochastic and distilled contributions to the connected vector correlator at $t/a = 43$, calculated with 160×4 distillation modes, for 30 configurations drawn from the E5 data set. Correlation in the data or lack thereof is difficult to discern with the bare eye.

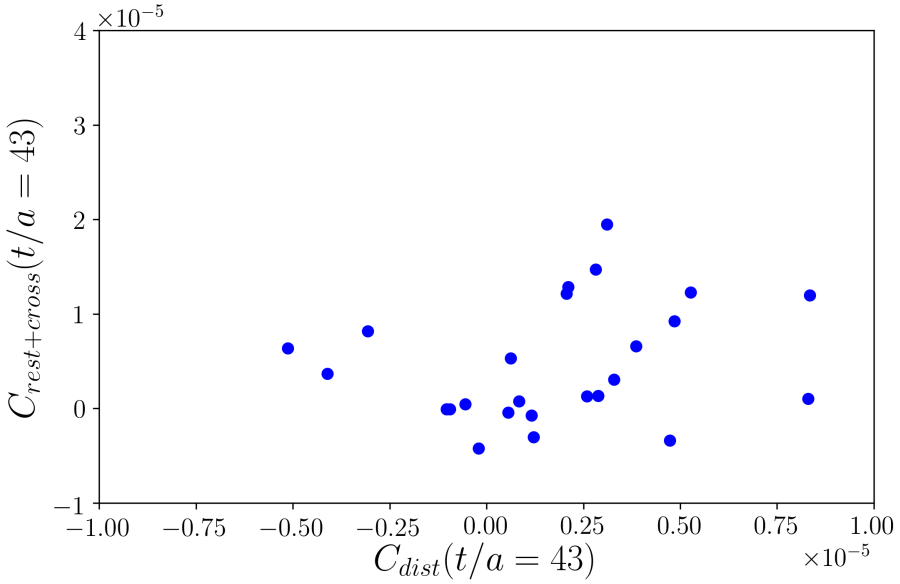


Figure 2.35: The stochastic and distilled contributions to the connected vector correlator at $t/a = 43$, calculated with 160×4 distillation modes, for 30 configurations drawn from the A5 data set. There is some visual suggestion of correlation in the data.

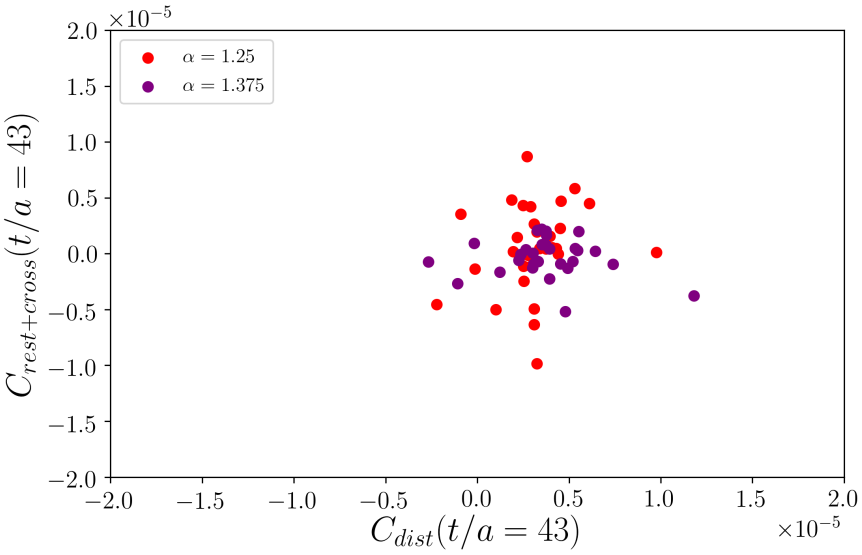


Figure 2.36: The stochastic and distilled contributions to the connected vector correlator at $t/a = 43$, with 160×4 distillation modes and $\alpha = 1.25, 1.375$, for 30 configurations. The correlation appears mostly exploited at $\alpha = 1.375$, implying an expected improvement in the size of the distilled contribution of about $1.375^2 = 1.89$.

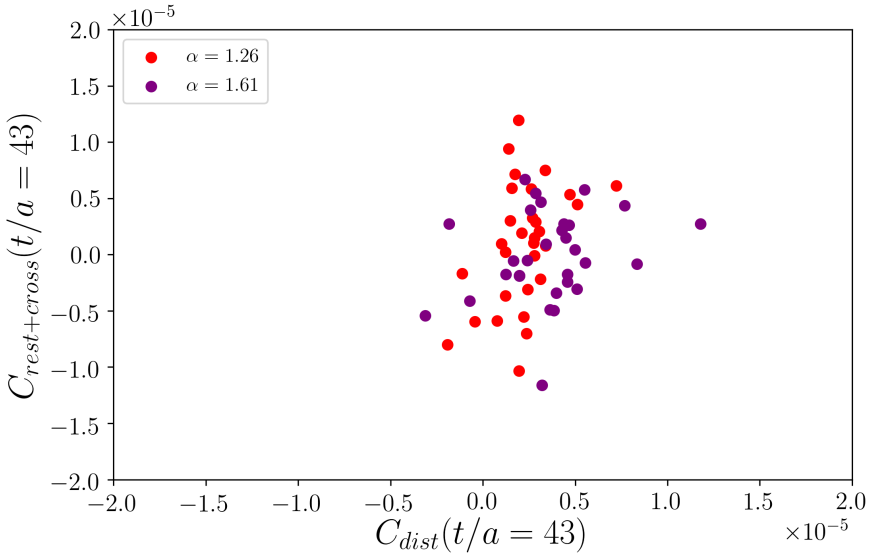


Figure 2.37: The stochastic and distilled contributions to the connected vector correlator at $t/a = 43$, with 80×4 distillation modes and $\alpha = 1.26, 1.61$, for 30 configurations. There is more exploitable correlation present than with 160 modes, compare figure 2.36. This trend also holds for 40 and 20 modes, with the correlation growing as the mode count shrinks.

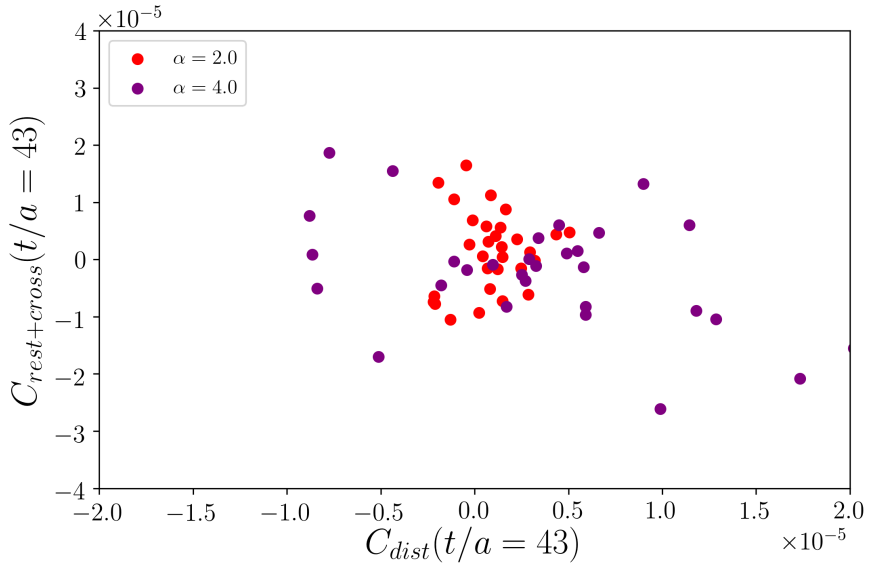


Figure 2.38: The stochastic and distilled contributions to the connected vector correlator at $t/a = 43$, with 20×4 distillation modes and $\alpha = 2.0, 4.0$, for 30 configurations. The 4.0 multiplier is too large, leading to a negative correlation.

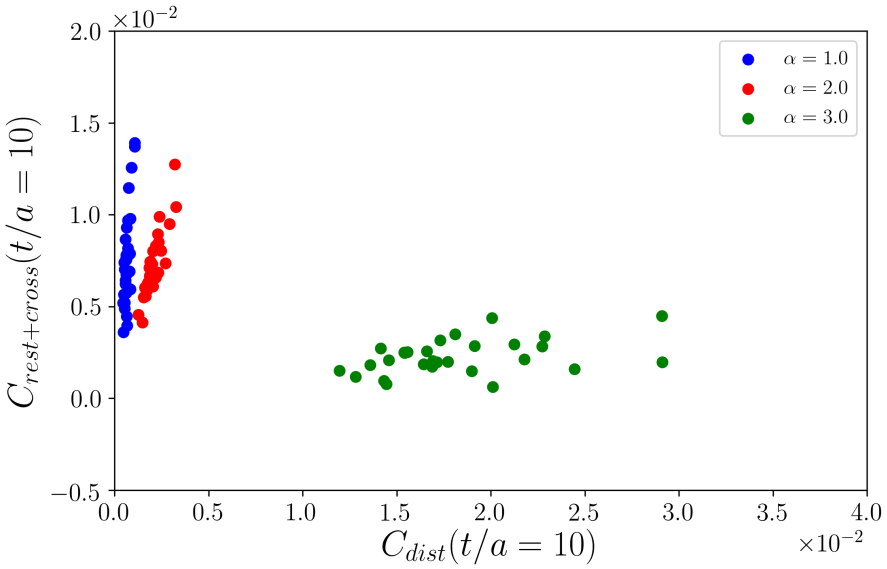


Figure 2.39: The stochastic and distilled contributions to the connected pseudoscalar correlator at $t/a = 10$, calculated 80×4 distillation modes with $\alpha = 1.0, 2.0, 3.0$. Each point shows the value for one of the 30 gauge configurations used. The correlation between the contributions seems mostly gone at $\alpha = 3.0$, implying a possible useful increase in the size of the distilled contribution by $3.0^2 = 9.0$ compared to the blue points in figure 2.18.

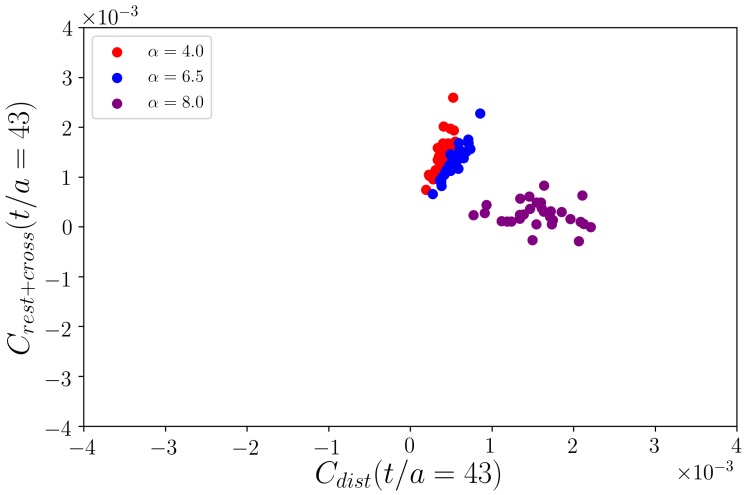


Figure 2.40: The stochastic and distilled contributions to the connected pseudoscalar correlator at $t/a = 43$, calculated with 20×4 distillation modes and $\alpha = 4.0, 6.5, 8.0$. Each point shows the value for one of the 30 gauge configurations used. The correlation between the contributions is mostly gone at $\alpha = 8.0$, implying a possible increase in the size of the distilled contribution by $8.0^2 = 64$ compared to the red points in figure 2.18. As with the vector correlator, the amount of exploitable correlation is larger at lower numbers of distillation modes. Compare figure 2.39. This trend also holds for 160 and 40 distillation modes.

stochastic contributions. These results seem to suggest that a lot of the same overlap with the correlator spectrum is shared between lower and higher modes of the distillation operator. Thus, calculating more distillation modes is to some extent redundant, and comparable precision gain can be achieved with fewer eigenmodes using a greater multiplier α. However, there is still a ceiling on how much of the correlator can be extracted this way. Experimenting with very roughly varying the mode count and multiplier for both observables seems to suggest a plateau of technique efficiency, where calculating further modes leads to little additional gain. This plateau lies, very roughly, at ca. 160×4 modes with $\alpha = 1.375$ for the vector correlator, and 20×4 modes with $\alpha = 8.0$ for the pseudoscalar correlator on the $E5$ data set. The plateaus were identified by varying mode counts by factors of 2, and increasing the multipliers for each mode number until the correlation in the scatter plots vanished.

Naively, looking back at equation 2.28, one might then expect that distillation with rescaling could be more efficient than pure stochastic sources at large times. 160 distillation modes lead to an overlap of ca. 0.5 for the vector correlator 2.15, so $\alpha = 1.375$ should increase this to ca. 0.945. For calculation with more than $160 \times 4 = 640$ stochastic sources, the cost of the distillation would then seem to be easily recouped. Specifically, for 192 stochastic sources, we would expect that distillation would yield a reduction in the stochastic error similar to using $\frac{192}{1-0.945} \approx 3490$ sources while costing

$$t_{160,l} + 640 t_D + 2 \times 192 t_D = 1024\, t_D + t_{160,l} \tag{2.34}$$

in computational resources. Since the time for obtaining the Laplacian eigenmodes, $t_{160,l}$, is still dominated by the Dirac solver time at 160 modes, see table 2.2, a calculation with distillation ought to be more efficient. However, in practice, the error at such large numbers of sources becomes quickly dominated by the systematic introduced when locating all the sources on a single time slice. As pictured in figure 2.45, increasing the number of stochastic sources located on a single time slice past 192 provides no benefit to the vector correlator for the 30 gauge configurations used in these tests. The same is true for the pseudoscalar, pictured in 2.46. Since the distillation low-mode averaging technique for connected correlators effectively prohibits making use of sources on different time slices, it is not cost-efficient here. The lack of precision gain from distillation for 192 sources for the vector correlator is shown in figure 2.44. The same holds for distillation with 20×4 modes and $\alpha = 8.0$ for the pseudoscalar, for the same reason.

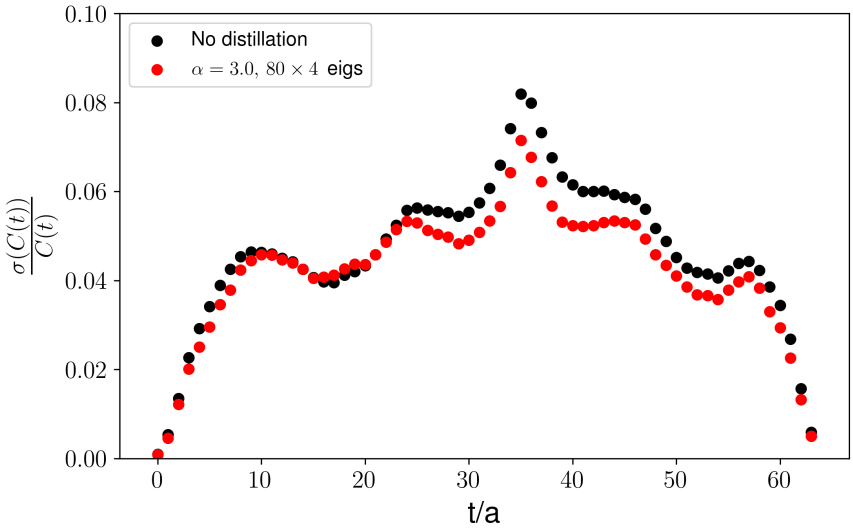


Figure 2.41: Relative precision $\frac{\sigma(C(t))}{C(t)}$ for the connected pseudoscalar correlator calculated on 30 configurations from the E5 data set, with 12 stochastic sources with and without distillation, with $\alpha = 3.0$.

The rescaling improves the overlap of the distillation spectrum with the connected vector correlator considerably, and improves the overlap for the connected pseudoscalar correlator even more, bringing the overlap for the two observables closer together. But for both observables, the overlap remains too small to make distillation low-mode averaging useful. For the technique to be cost-effective, the overlap would need to be so large with so few modes as to make solving the Dirac equation for distillation modes located on every time slice a worthwhile investment. That a modification of the distillation operator as simple as a scalar prefactor can already have such a big impact on its overlap for different observables seems encouraging, however. It suggests that the original idea, of finding a simple, cheaper operator, whose modes can substitute for the Dirac modes in low mode averaging, may be feasible after all, even though the specific approach of using the distillation operator for this did not work.

2.5.2 Disconnected correlators

Just as with the connected pseudoscalar, the distilled and stochastic contributions to the disconnected pseudoscalar correlator are strongly correlated, as can be seen in figure 2.47. Just as with the connected observables, this correlation is larger at lower numbers of distillation operator modes. To exploit this correlation, we scale the distillation operator by inserting $1 - \alpha\Box + \alpha\Box$ into equation 2.7. In contrast to the connected observables, α only appears in

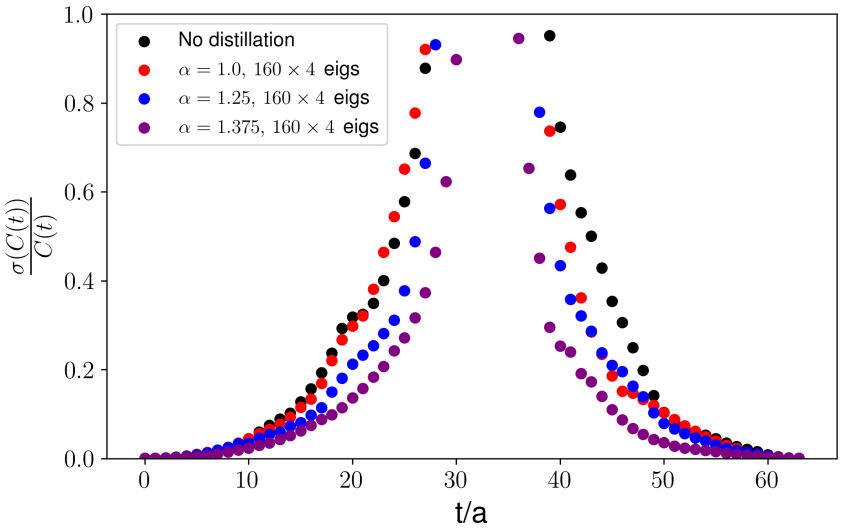


Figure 2.42: Relative precision $\frac{\sigma(C(t))}{C(t)}$ for the connected vector correlator, calculated on 30 configurations from the E5 data set, with 24 stochastic sources, 160×4 distillation modes and $\alpha = 1.0, 1.25, 1.375$. As would be expected from figure 2.36, larger α multipliers improve precision, since they decrease the size of the stochastic contributions.

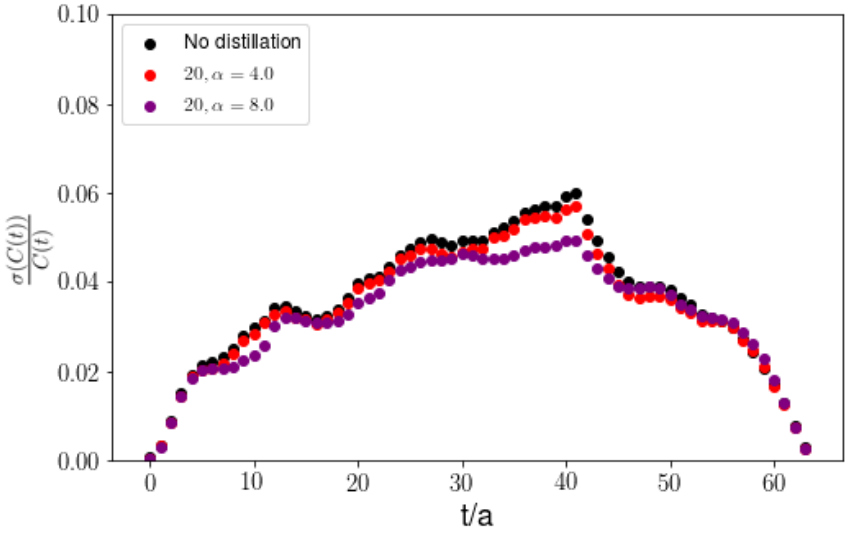


Figure 2.43: Relative precision $\frac{\sigma(C(t))}{C(t)}$ for the connected pseudoscalar correlator, calculated on 30 configurations from the E5 data set, with 24 stochastic sources, 20×4 distillation modes and $\alpha = 4.0, 8.0$. As would be expected from figure 2.40, larger α improve precision, since they decrease the size of the stochastic contributions.

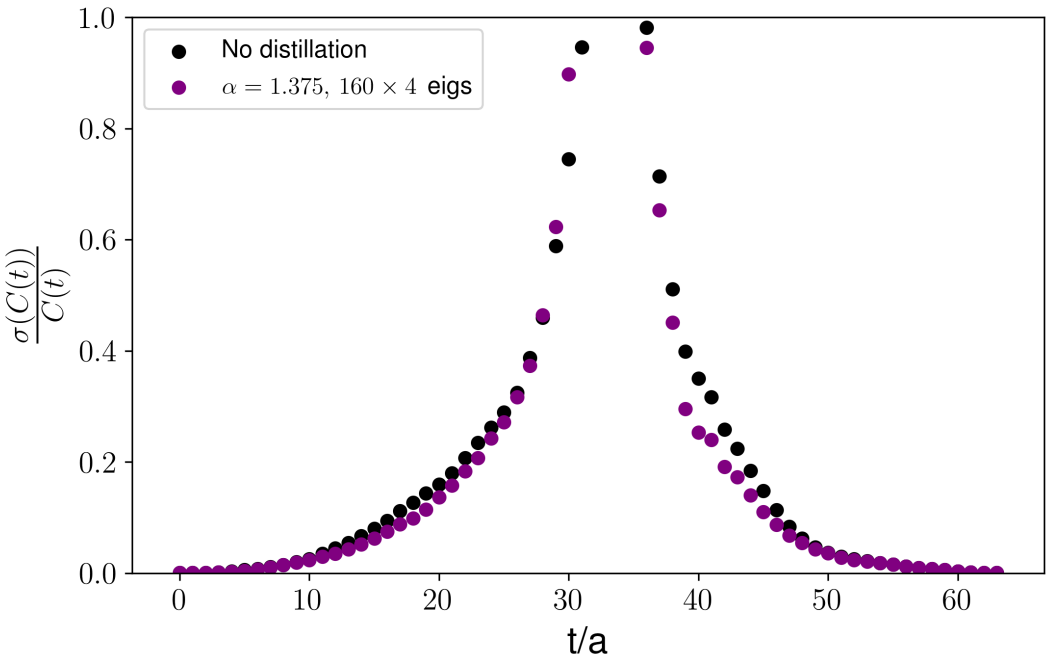


Figure 2.44: Relative precision $\frac{\sigma(C(t))}{C(t)}$ for the connected vector correlator, calculated on 30 configurations from the E5 data set, with 192 stochastic sources, 160×4 distillation modes and $\alpha = 1.375$. Unlike what might naively be expected from figure 2.42, showing the same calculation with 24 stochastic sources, the distillation fails to provide a relevant gain in precision. This is due to the distillation low-mode averaging requiring all stochastic sources to be located on a single time slice. The higher gauge noise resulting from this dominates the regular stochastic variance when employing many sources. See figure 2.45.

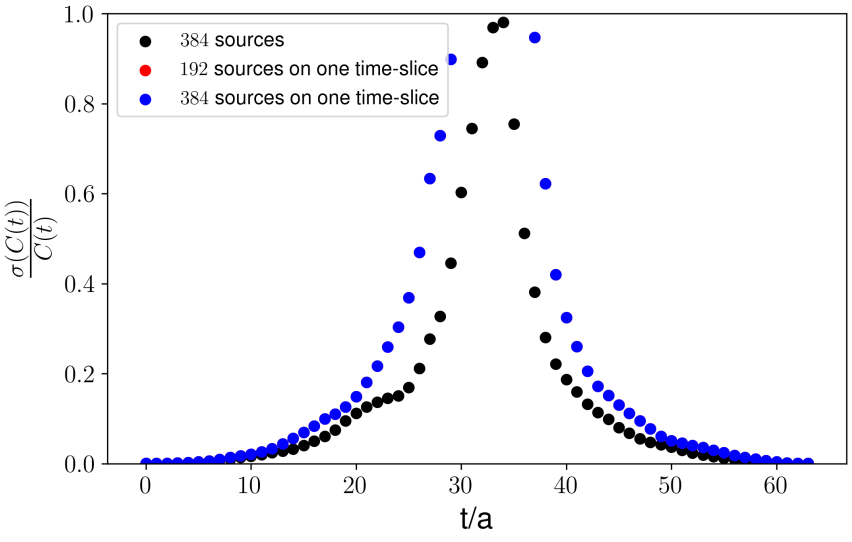


Figure 2.45: Relative precision $\frac{\sigma(C(t))}{C(t)}$ for the connected vector correlator, calculated on 30 configurations from the E5 data set, with 192 and 384 stochastic sources respectively located on one time slice, compared to 384 stochastic sources located on random time slices. The sources located on a single time slice lead to lower precision, and their relative errors match exactly, leaving the red points with 192 sources invisible. This indicates that the precision for sources located on one time slice saturates early, due to the higher gauge noise.

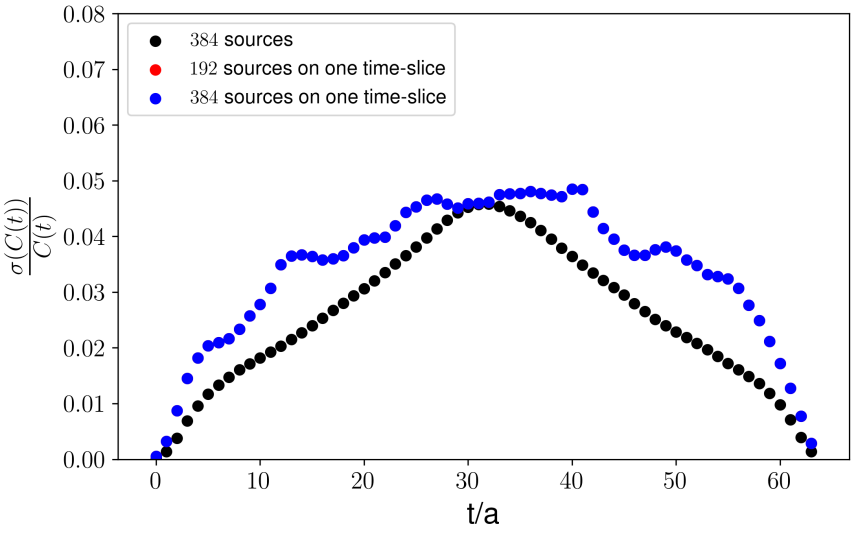


Figure 2.46: Relative precision $\frac{\sigma(C(t))}{C(t)}$ for the connected pseudoscalar correlator, calculated on 30 configurations from the E5 data set, with 192 and 384 stochastic sources respectively located on one time slice, compared to 384 stochastic sources located on random time slices. As in figure 2.45, the precision for sources located on one time slice saturates early, due to the higher gauge noise.

the term contracting the Dirac solutions here, not the solutions themselves. Thus, tests of the impact of using different α may be carried out without recalculating the Dirac solutions. Figure 2.48 shows the correlation for different choices of α for 20×4 modes, suggesting that α can be set as high as 8.0. Combining this information with figure 2.27 suggests that the distilled contribution could make up about $0.6 - 0.7$ of the complete correlator. Due to the low number of gauge configurations used here, the error on the pseudoscalar correlator is mostly dominated by the gauge noise, so the distillation does not notably decrease it.

However, this does suggest that the distillation would be theoretically cost-effective compared to pure stochastic sources for calculations with more gauge configurations and a very high number of stochastic sources per configuration, more than $64 \times 20 \times 4 = 5120$, to recoup the investment into the distillation modes. Extrapolating from the connected observables, using more distillation modes may enhance this advantage further. But it would also relegate it to calculations with even greater numbers of stochastic sources. For the noisier disconnected vector-vector correlator, resources were insufficient to carry out a correlation analysis as for the pseudoscalar. However, if a meaningful amount of correlation were present in the data, increasing α to exploit it should decrease the error on the total correlator, which is constant over time. Even if the contribution from the distilled spectrum were too small to make a detectable difference, or the gauge noise on the spectrum was of comparable size to

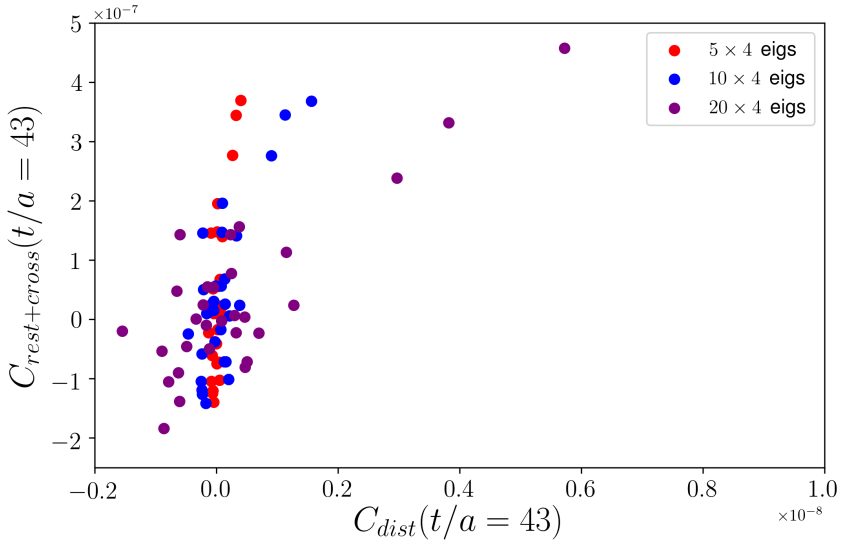


Figure 2.47: The stochastic and distilled contributions to the disconnected pseudoscalar correlator at $t/a = 43$, calculated with 5×4,10×4 and 20×4 distillation modes respectively. Each point shows the value for one of the 30 gauge configurations used. As with the connected observables, the correlation is larger at lower numbers of distillation modes.

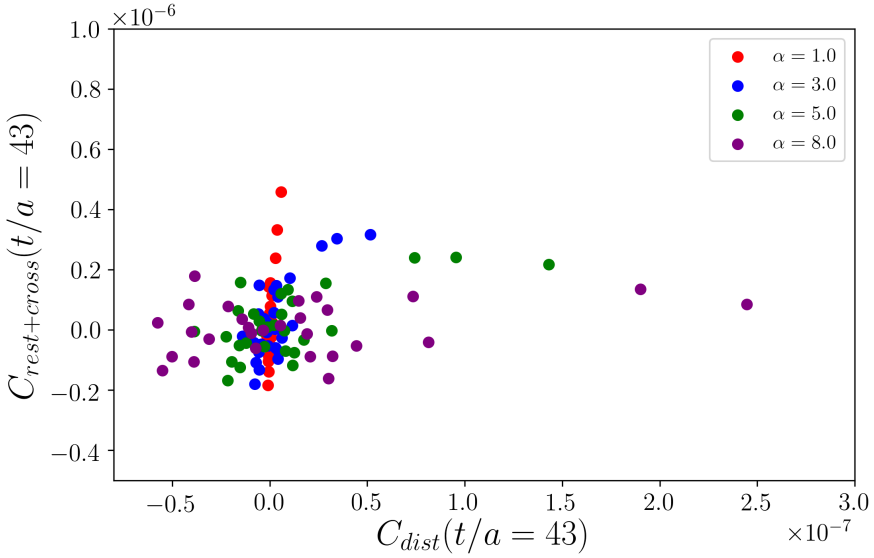


Figure 2.48: The stochastic and distilled contributions to the disconnected pseudoscalar correlator at $t/a = 43$, calculated with 20×4 distillation modes and different prefactors α. The correlation seems to be fully exploited at $\alpha = 8.0$. Looking at figure 2.27, this suggests a possible increase in $\frac{C_{dist}}{C}$ by ca. 64 from ca. 0.01 to about $0.6 - 0.7$.

the stochastic noise, increasing α should at the very least not increase the error on the total correlator until the optimal value to exploit the correlation is passed. So looking at the error averaged over time slices can provide a rough estimate or at the very least an upper limit, on the optimal α for the disconnected vector correlator. Table 2.4 shows this average error calculated with different values of α. Even when the average is restricted to later times to

α	$\frac{1}{64}\sum_{t=0}^{64a}\sigma\left(C_{VV}(t)\right)$	$\frac{1}{32}\sum_{t=16a}^{48a}\sigma\left(C_{VV}(t)\right)$
0	$7.83\times10^{-10}\pm1.12\times10^{-11}$	$7.91\times10^{-10}\pm1.67\times10^{-11}$
1.0	$7.63\times10^{-10}\pm1.09\times10^{-11}$	$7.69\times10^{-10}\pm1.62\times10^{-11}$
1.5	$7.77\times10^{-10}\pm1.12\times10^{-11}$	$7.80\times10^{-10}\pm1.62\times10^{-11}$
2.0	$7.99\times10^{-10}\pm1.07\times10^{-11}$	$8.04\times10^{-10}\pm1.53\times10^{-11}$
2.5	$8.34\times10^{-10}\pm1.06\times10^{-11}$	$8.42\times10^{-10}\pm1.52\times10^{-11}$

Table 2.4: The average error on the disconnected vector correlator, calculated on 30 gauge configurations taken from the E5 data set with distillation low-mode averaging on 20×4 modes, for different values of the distillation multiplier α. The first row shows the error calculated without distillation low-mode averaging.

give high-lying modes time to decay, there is a clear upwards trend in the size of the error as α increases. This indicates that just as with the connected contributions, the disconnected vector-vector correlator includes far less exploitable correlation than the disconnected pseudoscalar correlator, with the ideal value likely somewhere in the range $1.0 \leq \alpha < 2.0$. Given the necessary computational investment, this is very insufficient to make the technique practically useful.

2.6 Conclusion

We conclude that distillation-low-mode-averaging does not seem computationally efficient for the calculation of the vector and pseudoscalar correlators. Contrary to our hopes, the similarity between the low-lying spectra of the spatial Laplacian and the Dirac operator was not strong enough to allow efficiently substituting the eigenmodes of the former for the latter. Since the eigenmodes of the spatial Laplacian become denser at larger lattice volumes, we would expect the efficiency of the technique to fall off further if the physical lattice volume were increased.

However, a modification of the operator by a scalar prefactor optimized for the observable proved enough to significantly increase the spectrum overlap for connected and disconnected correlators. While still falling quite short of the bar for practical usefulness, this suggests that the operator overlap is both observable dependent and sensitive to simple transformations. Low-mode averaging with an operator related to the distillation operator through a different

simple transformation may thus still prove computationally efficient if the right transformation were found. Future work could attempt to find such transformations numerically. A possible starting point may be to use different scalar prefactors for different distillation modes or to equip the distillation operator with a non-trivial structure in spin space, perhaps matching that of the Dirac operator. This would not increase the cost of solving the Dirac equation for the operator eigenvectors but could bring it much closer to the Dirac operator in character. The distillation operator is also not necessarily the only starting point to consider in this context. Any operator or combination of operators with a cheaply evaluated eigenspectrum that is suspected to overlap with the physical states of interest in the correlator is a potential candidate.

Chapter 3

Machine Learning for Gauge Configuration Generation

3.1 Introduction

Machine learning has seen success in solving longstanding problems across a very wide range of domains in recent years. Examples include image recognition, an old problem in artificial intelligence that saw great progress starting with the AlexNet network, introduced in [49], protein folding, a bottleneck in biology alleviated substantially by machine learning, starting with the AlphaFold network detailed in [50], and autonomous driving. Across many fields, a small set of general techniques based around deep neural networks (NNs) trained with simple local numeric optimization techniques, implemented through the backpropagation algorithm, has seen massive success in solving problems that were previously computationally difficult or outright impossible to handle. Due in part to massive investment in this space, the scope, and power of these general techniques has also currently kept improving every year, as the power of hardware increases and better neural network architectures become available. In the wake of this, it could be taken as a good general policy for every field of science to have a look over its catalog of techniques and problems and see if any of them prove amenable to improvement through machine learning techniques.

In lattice QCD, one potential area of application is gauge configuration generation. Instead of, or supplementary to, classic Monte Carlo proposal techniques like HMC [4], machine learning algorithms are "trained" to generate gauge configurations for a particular action or class of actions. Unlike HMC techniques, the bulk of the computational effort for machine

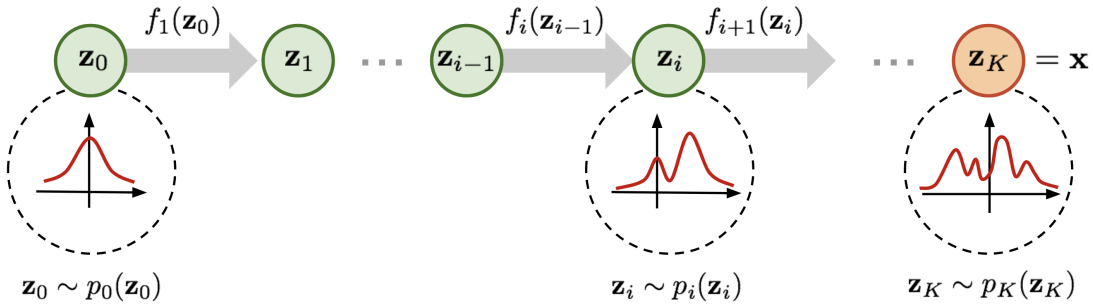


Figure 3.1: Normalizing flow model illustration, taken from [52].

learning models usually lies in carrying out this training. After training, configurations can be "sampled" from the model at comparatively low computational effort. To guarantee that configurations sampled in this manner follow the distribution of the action, they are then used as proposals for a Markov chain, with acceptance probability

$$A(z \to z') = \min\left(1, \frac{p(z')}{p(z)} e^{S(z')-S(z)}\right) . \tag{3.1}$$

Thus, it is necessary to use an algorithm for which we can compute $p(\varphi)$, the proposal probability. Investigation into ML configuration generation is in an early phase. As yet, no such algorithm has been successfully used to generate configurations used in a full, four-dimensional SU(3) QCD calculation. This is in large part due to current, early implementations of this class of algorithms having hardware requirements that appear to scale prohibitively with the number of lattice points used, as shown, e.g., in [51]. Nevertheless, progress has been made toward reaching this eventual goal.

The general class of machine learning algorithms that have seen the most attention to date are normalizing flow models. Normalizing flow models are invertible functions, optimized to implement a mapping from a trivial distribution over a number of variables, like a Gaussian or uniform distribution, to a more complicated distribution that is difficult to sample directly. They were originally developed for applications in image generation, though it should be noted that current image generation algorithms no longer use them. In our case, the variables in question are the values of the fields at the points and edges of the lattice, and the more complicated distribution is the action. So, the machine learning model is a function $f(z_0, \Theta)$, depending on a number of parameters Θ, that takes in field configuration z_0 drawn from some trivial distribution $p_0(z_0)$ and maps them onto other field configurations z that follow

the distribution given by the action, $p(z) \propto e^{-S(z)}$. The function begins initialized in a configuration of parameters Θ_0 drawn from an "initialization distribution". Variants on Gaussian distributions are common choices for this.

"Training" the neural network means running numerical optimization algorithms to optimize the values of Θ. Currently, this is usually done with variants of the stochastic gradient descent algorithm. Each iteration of the optimization algorithm can adjust every Θ. One iteration is called a training step. For normalizing flow models, in each training step, the inverse Jacobian of $f(z_0, \Theta)$ is calculated, which can be used to determine the distribution of the function's image:

$$z \sim p(z) = p_0\left(f^{-1}(z,\Theta)\right)\left|\det\left(\frac{df^{-1}(z,\Theta)}{dz}\right)\right| = p_0\left(z_0\right)\left|\det\left(\frac{df(z_0,\Theta)}{dz_0}\right)\right|^{-1}. \tag{3.2}$$

Then, the logarithm of this distribution is compared to the action. The more similar the two are, the closer f is toward performing as intended. To quantify the disparity between the distributions, the inverse Kullback–Leibler divergence is used. It is defined as

$$D_{KL}(p||e^{-S}) = \sum_{z\in T} e^{-S(z)}\left(S(z) - \log(p(z))\right), \tag{3.3}$$

with T being a set of samples drawn from p_0. The size of this set is a hyperparameter of training. Increasing the set size reduces the stochastic noise of the gradient descent parameter updates.

In many applications of ML, this stochastic noise actually seems to serve a useful purpose in preventing overfitting to the training data set, as can, e.g., be seen in [53]. But in our case, overfitting is not a concern, since we are not working with a training data set distinct from data that may be encountered in deployment. Thus, we can make effective use of larger batch sizes than those commonly used for other ML models. $D_{KL}(p||e^{-S})$ is our *loss function*. The training process aims to minimize this loss function over Θ. The dependence of f on Θ must be determined before the start of training and is called the *architecture* of the network. In our case of normalizing flow models, f is divided into K "layers". Each layer is a function that takes the output of the previous layer as its input. That is, each layer takes in a field configuration, modifies it, and then passes it to the next layer.

$$f(z_0) = f_K(f_{K-1}(...f_1(z_0)...)) \tag{3.4}$$

This structure enables efficient calculation of $\log(p(z))$

$$\log(p(z)) = \log p_0\left(z_0\right) - \sum_{i=1}^{K} \log \left| \det \left(\frac{df_i(z_{i-1}, \Theta)}{dz_{i-1}} \right) \right| , \tag{3.5}$$

and with it, $D_{KL}(p||e^{-S})$.

To keep the Jacobian determinants of individual layers easy to calculate, while simultaneously making the layers able to express highly complex mappings, an architectural trick is employed. The affine coupling layers split incoming variables z_i into two sets, *active* and *frozen*, of size d and $D - d$, respectively, where D is the dimension of z_i

$$\begin{aligned} z_{i,1:d} &= z_{i-1,1:d} \\ z_{i,d+1:D} &= z_{i-1,d+1:D} e^{s_i(z_{i-1,1:d})} + t_i(z_{i-1,1:d}) \ . \end{aligned} \tag{3.6}$$

This ensures the $\frac{dz_{i,1:d}}{dz_{i-1,d+1:D}}$ off-diagonal block of the Jacobian determinant for each layer is zero, meaning the log determinant is given purely by

$$\log \frac{dz_{i,d+1:D}}{dz_{i-1,d+1:D}} = s(z_{i-1,1:d}) \ . \tag{3.7}$$

And $\log(p(z))$ as a whole is just

$$\log(p(z)) = \log p_0\left(z_0\right) - \sum_{i=1}^{K} s_i(z_{i-1,1:d}) \tag{3.8}$$

This is the only term depending on the network output that appears in the loss function, and since it does not involve any derivatives of the s_i, and does not feature the t_i at all, the functional dependence of both on the input can be complicated without making the loss function too expensive to compute.Thus, both $s_i(z_{i-1,1:d})$ and $t_i(z_{i-1,1:d})$ can be neural networks of their own, which map $z_{i-1,1:d}$ to output couplings z_i. These networks contain the parameters Θ that are adjusted during training.

Theoretically, any kind of neural network could be used for $s_i(z_{i-1,1:d})$ and $t_i(z_{i-1,1:d})$. In this work, convolutional neural networks (CNNs) were employed, as they automatically respect translation invariance, have a proven track record in image recognition [49], and were used with success in flow-based models for lattice field theory before by other groups [54]. A

fully connected, or "MLP" Neural Network layer, is defined as a function

$$f_\alpha^{l+1} = \text{act}\left(W_{\alpha,\beta}^l f_\beta^l + b_\alpha^l\right) , \tag{3.9}$$

where f^l is the output of the previous layer, W^l and b^l are a matrix and a vector with freely adjustable parameters, and act is a non-linear *activation function*. Common choices for act include Leaky ReLU

$$R(x^+) := x \qquad R(x^{0/-}) = 0.01x\,, \tag{3.10}$$

ReLU

$$R(x^+) := x \qquad R(x^{0/-}) = 0\,, \tag{3.11}$$

ELU

$$R(x^+) := x \qquad R(x^{0/-}) = \alpha(e^x - 1)\,, \tag{3.12}$$

where α is a hyperparameter set by hand before training and tanh

$$\tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1}\,, \tag{3.13}$$

the hyperbolic tangent function. A convolutional layer may be defined as a fully connected layer with enforced translation invariance. For a single "input channel", it is the convolution of an input f^l of dimension $n^l \times n^l$ with a filter function g^l of dimension $k \times k$:

$$f_{i,j}^{l+1} = \sum_{m,n=0}^{k} f_{i+m,j+n}^l \, g_{m,n}^l \,. \tag{3.14}$$

If $n^{l+1} = n^l$, this definition requires "padding" the input f^l with $k-1$ additional rows and columns, $\frac{k-1}{2}$ at index $i = 0$, and $\frac{k-1}{2}$ at index $i = n^l - 1$. In general, there are different schemes for choosing these padded values. In this work, periodic padding was used, to match the boundary conditions of the lattice.

A layer with multiple input channels f^l uses separate g^l with trained parameters for each channel. Thus, a convolutional layer with i input channels usually has $i \times k \times k$ trainable parameters. Thus, the greater the "window" of local information a single convolutional layer takes into account, the greater its computational cost, scaling quadratically. Convolutional networks thus often learn long-range correlations in their inputs through stacking of many convolutional layers, in a hierarchy where early layers implement correlations on short scales,

and later layers implement successively larger scales. Convolutional layers can also be set to use "dilated" kernels g. A dilated kernel with dilation size $d > 1$ has "holes" inserted into it. Between adjustable parameters $g_{i,j}$, d entries of g are hard-coded to be 0. In other words, the convolution "skips over" inputs, in our case the field values at lattice sites. This allows layers to create longer-range correlations at a reduced cost, with the drawback of reducing resolution. Dilated convolutions were used for the Schwinger model with success in past works, for example, to sample it at criticality in [55].

CNNs have been replaced by transformer architectures in state-of-the-art image recognition algorithms, such as Coco, introduced in [56], and text-to-image generation algorithms, such as DALL-E 2. Transformer architectures combine MLP layers with multi-head self-attention layers, which were designed with the intent of enabling networks to focus their "attention" on the most relevant information in their input. They were introduced in [57], and have since become the architecture of choice for many ML applications. This marks a departure from the previous ML paradigm, in which architectures were less unified, and varied more between different applications. It may be prudent to explore the application of transformer-based models for lattice QCD in future work.

We also want the network architecture we choose to inherently respect the gauge invariance of our field theory. Theoretically, this invariance should also be learnable by the network during training. However, in current studies, this usually increases the training cost significantly. This was, e.g., observed in [37], which first proposed an architecture tailored to a gauge symmetry. However, it should be noted that in other parts of ML, the advantages of such symmetry-respecting architectures have more recently been found to shrink as the size (parameter count) of models increases. For example, newer image recognition networks use transformer architectures, which do not inherently respect translation symmetry. Given sufficient training data, they nevertheless match or outperform convolutional networks. Even newer, larger models based entirely on classic MLP layers, such as MLP-Mixer, introduced in [58], have become competitive with convolutional networks again. The cause of this is unknown, but speculatively, "preprogramming" the model with knowledge of the symmetries averts a fixed cost for learning them. Thus, very large models, that already learn a lot, gain relatively less from this preprogramming. The models used in this work, and the models currently used in other research in machine learning lattice QCD, may still be too small for this diminishing advantage to be of relevance, however. Future works using models with much

higher numbers of parameters might benefit from taking this effect into consideration.

To form a gauge invariant model architecture, gauge invariant or equivariant degrees of freedom are constructed out of the fields, and the coupling layers are exclusively used to transform those. The results are then transformed back into fields. This guarantees that the application of the layers commutes with gauge transformations.

3.2 Model

The test environment for implementing the technique described in Chapter 4 was the Schwinger model in 2D, with gauge fields coupling to flavors of spin-1/2 fermions

$$S(U,\psi) = -\beta \sum_n \sum_{\mu<\nu} \mathrm{Re}\, P_{\mu,\nu}(n) + \sum_{n,m} \sum_{f=1}^{n_f} \overline{\psi}^{(f)}(n) D[U](n|m) \psi^{(f)}(m)\,. \tag{3.15}$$

The plaquettes P are defined as

$$P_{\mu,\nu}(n) = U_\mu(n) U_\nu(n+\hat{\mu}) U_\mu^*(n+\hat{\nu}) U_\nu^*(n)\,. \tag{3.16}$$

After formally solving the Grassmann integrals, then rewriting the resulting Dirac operator determinant as an integral over an action with pseudofermion fields, this becomes

$$S(U,\phi) = -\beta \sum_n \sum_{\mu<\nu} \mathrm{Re}\, P_{\mu,\nu}(n) + \sum_{n,m} \sum_{f=1}^{\frac{n_f}{2}} {\phi^{(f)}}^\dagger(n) (D[U]D^\dagger[U])^{-1}(n|m) \phi^{(f)}(m) \tag{3.17}$$

Following the terminology introduced in [59], we chose a fully joint sampler, that is, the ML model proposes configurations for both the gauge bosons and pseudofermions to the Monte-Carlo chain. Much of the basic network architecture and code for network training and supervision were taken from, or heavily inspired by, the didactic introduction to flow-based models presented in [60].

To capture the dynamics of the gauge fields in the pure gauge part of the action, the gauge-invariant quantity chosen to be directly transformed by the neural network layers was the plaquette, copying the formulation and the code used in [60]. A coupling layer for the gauge fields then transforms the plaquette as

$$P_a'(n) = e^{s[P_f](n)}\, P_a(n) + t[P_f](n)\,, \tag{3.18}$$

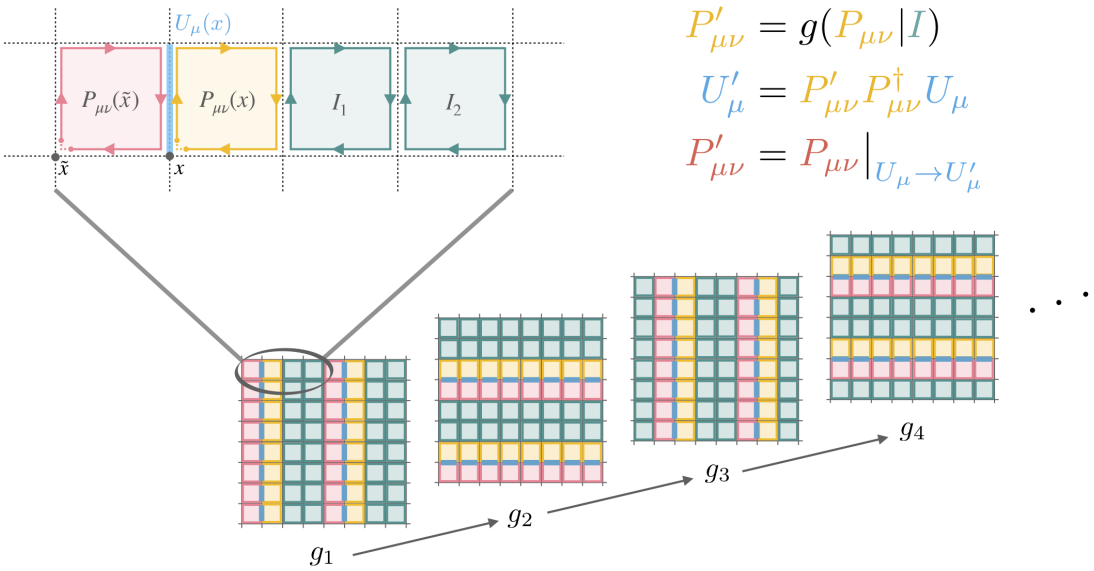


Figure 3.2: Freezing pattern for the plaquettes, taken from [60]

where s and t are neural networks, and the subscripts P_a and P_f indicate active and frozen plaquettes, respectively. After every update to the plaquettes, the update needs to be "pushed onto" the field configurations themselves. Due to the gauge invariance of the theory, these updates are not uniquely defined. For the gauge links, the updating rule also used in [60] was employed:

$$U'_\mu(n) = P(n)'_{\mu,\mu'} P(n)^{-1}_{\mu',\nu} U_\nu(n)\,. \tag{3.19}$$

This imposes a restriction on the possible freezing patterns. Since a single gauge link will be part of two plaquettes, the other plaquette including $U'_\mu(x)$ will also be "passively" updated by the layer, and can thus not be used as a frozen variable to pass to s or t. Following [60], a freezing pattern of alternating vertical and horizontal stripes over the lattice was thus chosen as the freezing pattern for the plaquette coupling layers. This pattern is also illustrated in figure 3.2. The Jacobian of the transformation induced by the affine coupling layer is then

$$\log J = \sum_x \mathbb{P}_a\, s[P_f](n)\,, \tag{3.20}$$

where $\mathbb{P}_a$ is the projector onto active plaquettes. For the pseudofermions and their interaction with the gauge fields, gauge equivariant variables were built by parallel transporting pseudofermions to the same lattice point

$$\phi^\mu(n) = U_\mu(n)\phi(n+\hat{\mu}), \tag{3.21}$$

following the formulation in [54], as first proposed in [61]. In total, this results in ten complex gauge equivariant variables,

$$\mathbb{L}[U,\phi](x) = \{\phi(x), \phi^{x_1}(x), \ldots, \phi^{x_{2\times d}}(x)\}\,, \tag{3.22}$$

where x_i labels the $2 \times d = 4$ neighbors of x. These variables are then transformed using a *parallel transport convolution* [54]

$$\mathrm{PTC}[U,\phi]_\alpha = \sum_\beta M_{\alpha,\beta}[P]\mathbb{L}_\beta\,, \tag{3.23}$$

with $\beta \in [1, ..., 10]$. The entries of M are the outputs of convolutional networks, with the plaquettes P as their inputs. As the freezing pattern for the pseudofermions ϕ, alternation in spin and a checkerboard pattern on the lattice positions was chosen. The update to the fermionic fields is then given by

$$\begin{aligned}
\phi_a(n) &= \mathbb{P}_a\phi(n) \\
\phi_f(n) &= (1-\mathbb{P}_a)\phi(n) \\
\phi'_a &= a[P](n)\,\phi^a + \mathrm{PTC}[U,\phi_f]_\alpha\,,
\end{aligned} \tag{3.24}$$

with

$$\begin{aligned}
\mathbb{P}_a(n_a) &= 0.5(1 \pm \rho_z) \\
\mathbb{P}_a(n_f) &= 0\,.
\end{aligned} \tag{3.25}$$

$a(P)$ is, again, given by the output of a convolutional neural network. The Jacobian for the pseudofermion layers is then

$$\log J = 2\sum_{n_a} \log|a[P](n)|\,. \tag{3.26}$$

It should be noted that this gauge equivariant does not grant the CNN networks access to the fermionic variables. The only inputs to the CNNs forming M and a are the plaquettes. The fermionic degrees of freedom are only taken into account indirectly, through the linear PTC transformation. Despite the restricted information to the CNNs, these fermionic layers have been successful in practice.

As the initial distribution $p_0(U,\phi)$ to sample from, a uniform distribution over $[-\pi,\pi]$ was chosen for the phases of the gauge links, and a Gaussian distribution centered on zero

for the real and imaginary parts of the pseudofermion components $\mathrm{Re}\,\phi$, $\mathrm{Im}\,\phi$. Periodic boundary conditions, and anti-periodic boundary conditions for ϕ in the time direction, need to be taken into account by defining the Dirac operator in the action accordingly. The CNN networks only directly access the plaquettes, so periodic padding in the kernels suffices to enforce the boundary conditions on the model. The complete model $f(U, \phi)$ then consists of m affine coupling layers for the plaquettes as defined in equations 3.18, 3.19, followed by $n_f \times n$ pseudofermion layers as defined in equation 3.24. After training, configurations produced by the model can be used as proposals for a Metropolis-Hastings algorithm

$$A(U, \phi^0 ..., \phi^{n_f} \to U', \phi'^0 ..., \phi'^{n_f}) = \min\left(1, \frac{p(U', \phi'^0 ..., \phi'^{n_f})}{p(U, \phi^0 ..., \phi^{n_f})} \frac{e^{S(U', \phi'^0 ..., \phi'^{n_f})}}{e^{S(U, \phi^0 ..., \phi^{n_f})}})\right) . \tag{3.27}$$

Chapter 4

Sampling fermionic lattice theories locally using flow-based models

4.1 Introduction

Multilevel Monte Carlo (MMC) techniques have seen success in reducing the computational costs needed to generate gauge field configurations. For example, [62] used MMC to calculate the hadronic vacuum polarization. The core idea of MMC techniques is to divide the lattice into multiple domains and generate configurations independently on each. In pure gauge theory, where degrees of freedom are independent, if one creates n configurations on the right half of a lattice and n on the left half, they can be combined through simple concatenation to form n^2 configurations on the whole lattice.

In a theory with fermions, the fermion determinant, which is non-local, hinders this procedure. However, the long-range interaction terms in the determinant that cross the domain boundary between lattice halves can still be approximated through a polynomial, allowing a division of the lattice into two subdomains with Dirichlet boundary conditions [63, 64]. For a lattice decomposition into three domains Λ_0, Λ_2 and a thinner region Λ_1 to separate the former two, the determinant of the Hermitian Wilson-Dirac operator $Q = \gamma_5 D$ is rewritten as [62]

$$\det Q = \frac{\det (1 - w)}{\det Q_{\Lambda_1} \det Q^{-1}_{\Lambda_0+\Lambda_1} \det Q^{-1}_{\Lambda_2+\Lambda_1}} , \tag{4.1}$$

with Q_{Λ_1}, $Q_{\Lambda_0+\Lambda_1}$, and $Q_{\Lambda_2+\Lambda_1}$ restricted to the domains specified by the subscript. w is

$$w = \mathbb{P}_{\partial\Lambda_0} Q^{-1}_{\Lambda_0+\Lambda_1} Q_{\Lambda_{1,2}} Q^{-1}_{\Lambda_2+\Lambda_1} Q_{\Lambda_{1,0}} , \tag{4.2}$$

where $Q_{\Lambda_{1,0}}$ and $Q_{\Lambda_{1,2}}$ are the hopping terms of Q across the boundaries separating Λ_1 from Λ_0, and Λ_2, respectively. $\mathbb{P}_{\partial\Lambda_0}$ is the projector onto the boundary of Λ_0 [64]. w can be understood as the component of interactions between the bulk lattice domains Λ_0 and Λ_2 in $Q = \gamma_5 D$. The key step now is to carry out a polynomial expansion of $\frac{1}{1-w}$,

$$\det\,(1-w) \propto \prod_{k=1}^{N/2} \frac{1}{\det\left[(u_k - w)^\dagger (u_k - w)\right]} + \mathcal{O}(N+1)\,, \tag{4.3}$$

yielding an expression for $Q = \gamma_5 D$ with the dependence on the gauge fields factorized into Ω_0, Ω_2 and Λ_1. In this way, the computational advantage gained through the concatenation in the trivial pure gauge case can still be somewhat exploited, through a division of configurations into a hierarchy. "First level" configurations on the whole lattice are generated the usual way and used to set the boundary conditions for two (or more) subdivided lattices. "Second level" configurations generated on these subdivided lattices can then be concatenated.

Faced with the success of the MMC fermionic polynomial expansion procedure, one may wonder at the feasibility of carrying it further. What if we approximate the inverse Dirac operator as a polynomial everywhere, effectively localizing the entire pseudofermion action up to nearest neighbor interactions? Operating in such a theory could be conjectured to allow for the subdivision of problems on very large lattice volumes into many problems on small lattice volumes. If successful, this could perhaps take MMC techniques much further, and improve the scaling of machine learning techniques at large lattice volumes. This idea was first proposed in 1994, though in a completely different context unrelated to MMC and machine learning, as the "Lüscher technique" [39]. The core idea of the technique is to use a polynomial expansion of the Dirac operator to trade a non-local fermionic action for a local action with a greatly increased number of pseudofermion fields. This was originally intended for use with classic, non-machine-learning-based samplers, enabling Markov proposals through purely local field updates on configurations. But the action proved to be too strongly peaked for this application, with small changes to a single gauge link causing large changes in the integrand.

Here, we instead seek to sample from the Lüscher action with machine learning techniques. Since the trivializing maps of flow-based ML models follow entirely different sampling principles, they may prove successful where classic methods based on updating field values in previously accepted configurations failed. We tested this notion on the Wilson Schwinger

action with $2D$ $L = 8$ and $L = 16$ lattices, using flow-based models as detailed in the previous chapter. This approach could be considered related to [65], which attempted to use machine learning models to sample the Schwinger model on a large lattice by factorizing it into smaller ones. However, in that paper, the fermion determinant was dealt with through LU decomposition.

4.2 Lüscher action

The following briefly outlines the derivation of the Lüscher action in the original [39] paper. Starting from a conventional action

$$S(U,\psi) = S_G(U) + \sum_f \sum_{n,m} \overline{\psi}^{(f)}(n) D[U](n|m) \psi^{(f)}(m)\,, \tag{4.4}$$

for a positive hopping parameter $\kappa = \frac{1}{2d+2m}$, where d is the lattice dimension, one can show that [39]

$$\|\gamma_5 D\| \leq 2d + m_f\,. \tag{4.5}$$

So by normalizing $\gamma_5 D$, which has real eigenvalues, we can obtain

$$Q' := \frac{\gamma_5 D}{2d + m_f}\,. \tag{4.6}$$

This has a real eigenspectrum localized between $+1$ and -1, enabling a polynomial expansion of $\frac{1}{Q'^2}$. Ideally, the expansion used is chosen to optimize convergence speed, but for this work, the naive expansion

$$P_{n_L}(Q'^2) = \sum_k^{n_L} (1 - Q'^2)^k \tag{4.7}$$

is used as a proof of concept.

Applying this expansion to the fermion determinant of two identical fermion flavors, one obtains

$$\det[Q'^2] = \lim_{n_L \to \infty} \det[P_{n_L}(Q'^2)]^{-1}\,. \tag{4.8}$$

The polynomial can be written as a product of its roots. In the case of 4.7, this yields

$$\begin{aligned} z_k &= 1 - e^{\frac{2\pi k}{n_L+1}} \\ \sqrt{z_k} &=: \mu_k + i\nu_k \ , \nu_k > 0 \\ P(Q'^2) &= \prod_{k=1}^{n_L} \left((Q' - \mu_k)^2 + \nu_k^2 \right) \ , \end{aligned} \tag{4.9}$$

where every value of ν_k occurs twice with opposite sign for the corresponding μ_k. The n_L roots z_k of the polynomial P then form the terms of a new action, with n_L different multi-boson fields that couple to the gauge field, but not to each other:

$$S_L(U,\phi) = S_G(U) + \sum_{n,m} \sum_{k=1}^{n_L} \phi_k^\dagger(n) \big(Q' - \mu_k\big)^2 (n|m) \phi_k(m) + \nu_k^2 \phi_k^\dagger(n) \phi_k(m) \, . \tag{4.10}$$

To compute observables O in this new theory, note first that any fermionic observables may be written as

$$O = \sum_{n,m,\alpha,\beta,A,B} a(n,m,\alpha,\beta,A,B) \big(\overline{\psi}_{\alpha,A}(n) \psi_{\beta,B}(m)\big)^{b(n,m,\alpha,\beta,A,B)} \, , \tag{4.11}$$

$a \in \mathbb{C}$, $b \in \mathbb{Z}$, without loss of generality.

If a source term

$$O = \sum_{n,m,\alpha,\beta,A,B} J_{\alpha,A,\beta,B}(n,m) \overline{\psi}_{\alpha,A}(n) \psi_{\beta,B}(m) \tag{4.12}$$

is added in equation 4.4, any observable can therefore be expressed through derivatives with respect to the source fields J. After carrying out the transformation into the Lüscher theory as before, this addition results in the operator Q' in equation 4.10 being shifted by J. Thus, the derivatives with respect to the source terms become polynomials in ϕ, $\phi^\dagger$, $Q'\phi$, $\phi^\dagger Q'$. Thus, calculating observables with the Lüscher action does not require costly inversions of the Dirac operator. The predominant contribution to the cost of computing a lattice observable is solely the sampling from the action 4.10.

Originally, the Lüscher technique failed to become competitive, because sampling from the Lüscher action proved intractable with classic approaches. Specifically, the distribution $p(U|\phi_0, .., \phi_{n_L})$ of the gauge links conditional on the pseudofermions is very sharply peaked. Here, the action is instead sampled with a flow-based model for the joined action, with an

architecture as described in the previous chapter. The only substantive change to the sampling setup for the Schwinger model detailed there is the presence of many multi-boson fields. In realistic sampling, the number of multi-boson fields n_L will usually go into the hundreds, or higher, to make the approximation of the Dirac operator precise enough. Training separate neural networks for the coupling layers of each Lüscher layer would thus add great computational costs to the technique at scale.

Instead, a single neural network is used for all multi-boson fields. Instead of trying to approximate the distribution of ϕ_k with particular choices of μ_k, ν_k, the network is instead trained to sample the action 4.10 for whatever values $\mu_k, \nu_k, k = 1...n_L$ it is given. This is achieved by creating two additional input channels for the convolutional neural networks forming the coupling layers of the multi-boson fields. The input to these channels is simply the value of μ_k and ν_k for the field being transformed, expanded to the size of a $L \times L$ tensor object since this is the input format the standard CNN architecture expects. Despite the crudeness of this implementation, in the experiments detailed in this chapter, it proved sufficient to provide the neural network with the information needed to learn to sample the action for any values of μ_k and ν_k.

4.3 Implementation and results

A normalizing flow model with gauge invariant layers as described in the previous chapter was used to sample from the joint distribution $q(U, \phi_k) = e^{-S_L(U,\phi_k)}$ given by the action 4.10. The model samples $U = e^{i\varphi_\mu}$ from a uniform distribution, $\varphi_\mu \in [-\pi, \pi]$, and ϕ_i from a normal distribution

$$p_0(\phi) = \frac{1}{0.7\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\mathrm{Re}(\phi)}{0.7}\right)^2 - \frac{1}{2}\left(\frac{\mathrm{Im}(\phi)}{0.7}\right)^2}. \tag{4.13}$$

The standard deviation $\sigma = 0.7$ was chosen through rough trial and error to reduce the networks' training time. The sample is then passed through a total of 24 affine coupling layers for the gauge fields as described in equation 3.18, followed by 16 affine coupling layers for the multi-boson fields as described in equation 3.24. The CNNs inside the gauge coupling layers use two input channels for the real and imaginary part of the plaquette, $\cos(P)$ and $\sin(P)$, followed by two hidden layers with 32 channels each, followed by three output channels

for s_1, s_2 and t, where the average of applying s_1 and s_2 is used to define s

$$e^{s[P_f](x)} P_a(x) = \frac{1}{2}\left(e^{s_1[P_f](x)} P_a(x) + e^{s_2[P_f](x)} P_a(x)\right) . \tag{4.14}$$

This follows the implementation by [60]. Averaging over multiple s can increase performance by heightening the range of expression for individual layers. The convolutional kernels chosen were 3×3, with a dilation of 1,2, and 3 for the first, second, and final convolution, respectively. The affine coupling layers for the multi-boson fields use two CNNs, one for $a[P]$ and one for $M[P]$. They both have four input channels, for $\cos(P)$, $\sin(P)$, and the Lüscher action parameters μ_k, ν_k^2. Likewise, they both use two hidden layers with 32 channels each. The CNNs for $a[P]$ have two output channels for their real and imaginary components. The CNNs for $M[P]$ have eighteen output channels for the real and imaginary components of the nine entries of $M[P]$. Both types of CNNs use 3×3 convolutional kernels without dilation. These architectures were chosen ad hoc, through limited trial and error, rather than any application of fundamental principles. The point of these results is to serve as a first test for whether NN-based techniques can sample the Lüscher action at all, not to do so efficiently. A more dedicated study of different network hyperparameter choices and additional possible layer inputs, like the $2x1$ Wilson loops used in [55], might find many ways to increase performance.

Networks were implemented with Pytorch, and trained with the AdamW optimizer [66] on a Google Colab GPU. The learning rate was set to $\eta = 10^{-3}$ at the start of training and decreased to $\eta = 10^{-4}$ at the halfway point. Figure 4.1 shows the performance of one of the networks over the course of training. It depicts the loss D_{KL} and the effective sample size (ESS)

$$\rho_{\text{ESS}} = \frac{\left(\frac{1}{N}\sum_j p(U_j, \phi_{j,1}...\phi_{j,n_L})/e^{(-S(U_j,\phi_{j,1}...\phi_{j,n_L}))}\right)^2}{\frac{1}{N}\sum_j \left(p(U_i, \phi_{j,1}...\phi_{j,n_L})/e^{-S(U_j,\phi_{j,1}...\phi_{j,n_L})}\right)^2} , \tag{4.15}$$

which is defined on the N samples included in a single update step, averaged over the last hundred gradient updates. The ESS is a normalized quantity taking values in the range $[0, 1]$, with higher values indicating a closer match between p and e^{-S}. Since the loss function is not normalized, the ESS can make for a more informative quantity to monitor in training. However, its variance is too high to make for a suitable loss function itself [60]. It should also be noted that the ESS is biased upwards for small N. The trained networks were used

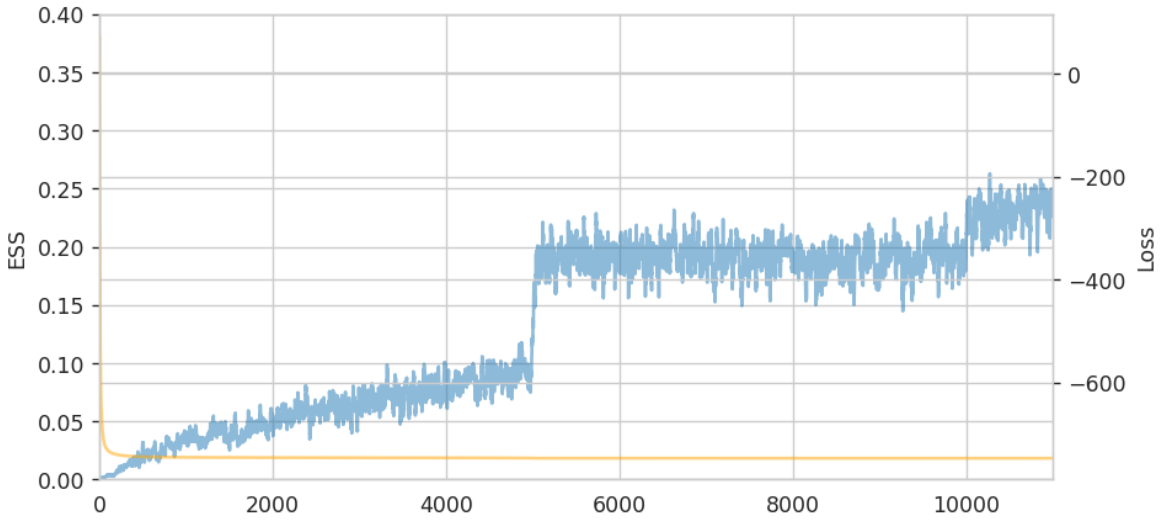


Figure 4.1: The Loss in yellow and Effective Sample Size in blue, both averaged over the last hundred training steps, for a network trained on a $L/a = 8$ $2D$ lattice with $n_L = 4$ multi-boson fields. Further information on the network can be found in the second row of table 4.1. The jumps in the curve are a result of learning rate scheduling.

to make proposals in a Metropolis sampler. Following [60], we define a topological charge

$$Q := \frac{1}{2\pi} \sum_n \arg(P_{0,1}(n)) \,, \tag{4.16}$$

on the lattice, which, in the 2D Schwinger model, takes integer values $Q \in \mathbb{Z}$ [67]. The topological susceptibility Q^2 and the average plaquette

$$P := \frac{1}{L^2 \beta} \sum_n \mathrm{Re}\, P_{0,1}(n) \,. \tag{4.17}$$

were then calculated with bootstrapping. The first six rows in table 4.1 show quantities calculated on a $(L/a)^2 = 8^2$ lattice, along with the Markov proposal acceptance rates, for a theory with $\beta = 2.0$, $\kappa = 0.265$, corresponding to a bare mass of $ma = -0.11$, and $n_L = 2$ to $n_L = 12$ multi-boson fields. The Markov chains all used $3,600,000$ proposals, discarding the first $600,000$ steps for thermalization. Observables were averaged together into bins of s_{bin} Markov time steps for the bootstrap to remove autocorrelations. For more detail on the error analysis, see section B.2 in the appendix. The acceptance rate drops quickly with the number of multi-boson fields. The average plaquette grows with the boson count, presumably approaching the asymptotic value of the conventional Dirac action for $n_L \to \infty$. This is also depicted for the $L/a = 8$ networks in figure 4.2. In contrast to HMC samplers [4], which

L/a	κ	n_L	Accept. rate	$E(P)$	$E(Q^2)$	Batch size	Training steps	s_{bin}
8	0.265	2	0.453	0.69750 ± 0.00007	1.242 ± 0.002	1024	11000	600
8	0.265	4	0.376	0.69717 ± 0.00009	1.246 ± 0.003	1024	11000	600
8	0.265	6	0.276	0.69818 ± 0.00013	1.237 ± 0.004	1024	11000	1500
8	0.265	8	0.226	0.69974 ± 0.00017	1.233 ± 0.005	1024	11000	2000
8	0.265	10	0.128	0.70085 ± 0.00034	1.227 ± 0.010	900	12520	3000
8	0.265	12	0.116	0.70270 ± 0.00036	1.218 ± 0.009	650	17340	3000
8	0.125	8	0.311	0.69867 ± 0.00011	1.233 ± 0.003	1024	11000	1000
16	0.265	2	0.172	0.69739 ± 0.00011	4.984 ± 0.030	1024	11000	2000
16	0.265	4	0.124	0.69708 ± 0.00017	5.049 ± 0.042	900	12520	3000

Table 4.1: Results from flow-based samplers. Larger configurations require smaller sample batches per training step to reduce memory consumption. All results were calculated on $10,000$ bootstrap samples. See section B.2 in the appendix for more detail on the bootstrap.

often exhibit topological freezing, the flow-based samplers move quickly between topological sectors over Markov time, as was noted and examined in detail in [37]. Figures 4.3 and 4.4 show the topological charge over Markov time for the $n_L = 4$ networks. All networks used the same number of parameters. Plots of the topological charge over Markov time for the other networks listed in table 4.1 can be found in appendix A. Networks with more degrees of freedom start exhibiting visible plateaus in the topological charge again, due to the lower acceptance rate. Computing the gradient for 10+ multi-boson fields requires training with fewer samples per training step to fit into GPU memory. The number of training steps was then increased in proportion to keep the total number of data points used constant. The scaling with the field count would likely prove prohibitively bad for real applications. Practical networks would require more training time, more parameters, or a more efficient architecture, to better learn the general algorithm for sampling with any μ_k, ν_k^2. Replacing the convolutional channels for the action parameters with fully connected MLP layers, and the affine coupling with a more general spline [51], could allow the network to more easily express the dependence of the needed transformations on the multi-boson index. Raising the mass to $ma = 2.0$, corresponding to $\kappa = 0.125$, lowers the value of the average plaquette. It also raises the acceptance rate, as the fermionic degrees of freedom decline in relevance. $m \to \infty$ recovers the simple heat bath sampling of the pure gauge theory. Finally, the last two rows show results for networks trained on a $(L/a)^2 = 16^2$ lattice, with $n_L = 2, 4$. As is observed with other samplers, the bigger lattice volume reduces the acceptance rate. Sampling more multi-boson fields or large lattice volumes efficiently would require reducing the programs' memory consumption to allow for large enough sample sizes per gradient update.

We conclude that flow-based models can indeed sample from the Lüscher action for the $2D$ Schwinger model, even with the primitive architecture and very small-scale computational

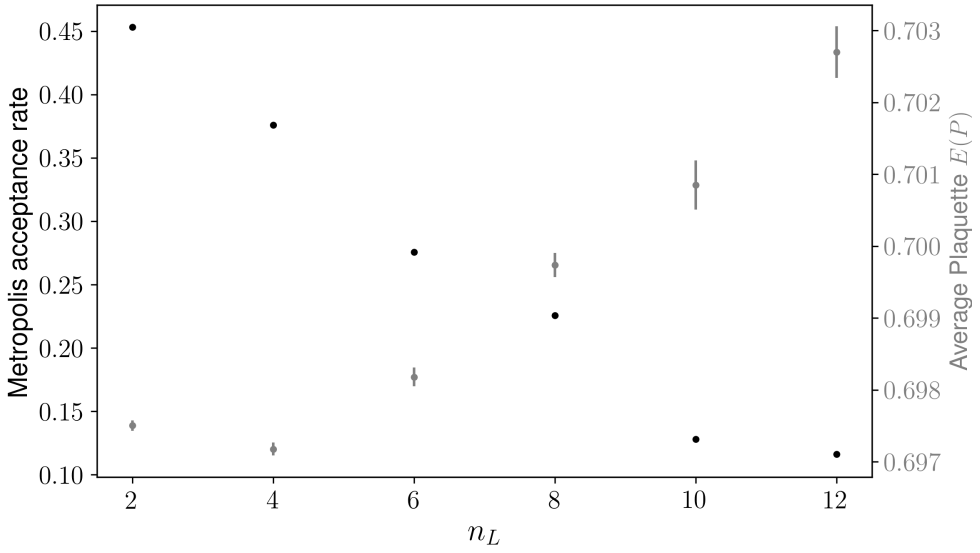


Figure 4.2: The Metropolis acceptance rate and average plaquette for networks with different numbers of multi-boson fields n_L on a $L/a = 8$ lattice. All networks use the same number of parameters and were trained on roughly the same number of data points. The average plaquette grows with n_L, presumably converging on the value for the conventional Schwinger model for $n_L \to \infty$. The acceptance rate falls roughly linearly until $n_L = 10$. The $n_L = 12$ network needed to be trained on significantly smaller data batches for more training steps due to hardware constraints, which could be confounding the comparison. So the plateau in the acceptance rate might be spurious. See the first six entries of table 4.1 for more information on the networks.

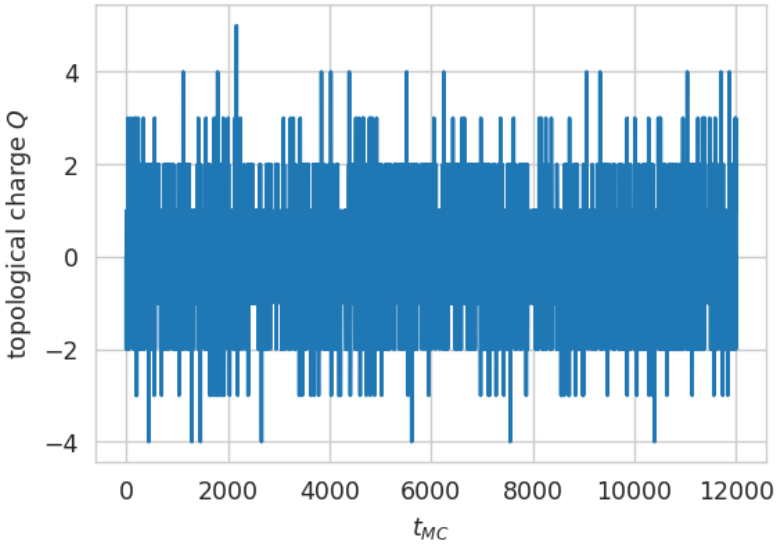


Figure 4.3: The topological charge Q over 12,000 Markov steps, after 600,000 steps of thermalization, for a network trained on a $L/a = 8$ $2D$ lattice with $n_L = 4$ multi-boson fields. Further information on the network can be found in the second row of table 4.1. The Markov chain quickly switches between topological sectors.

investment used for these exploratory models. The scaling of the acceptance rate with the number of multi-boson fields would need to be improved for the technique to become practical. However, one should be cautious not to extrapolate these results too far. Compared to classical samplers, flow-based models and machine learning-based samplers more generally may be better thought of as a very large family of algorithms than a single algorithm [68]. Even two networks trained on the same architecture, with the same type of coupling layers, but differing numbers of parameters or training time, could theoretically become quite different algorithms. A straightforward extension of the results seen here to different actions and networks may be inadvisable. To find out how well a given network can solve a given problem, there seem to be, at present, few good alternatives to actually training the network and seeing what happens.

4.4 Next steps

4.4.1 Memory efficient multi-boson fields

While the current implementation does not suffer from a fundamental requirement of scaling the number of model parameters with the number of multi-boson fields n_L, it nevertheless requires the field configurations of all n_L multi-boson fields to be stored in GPU memory to

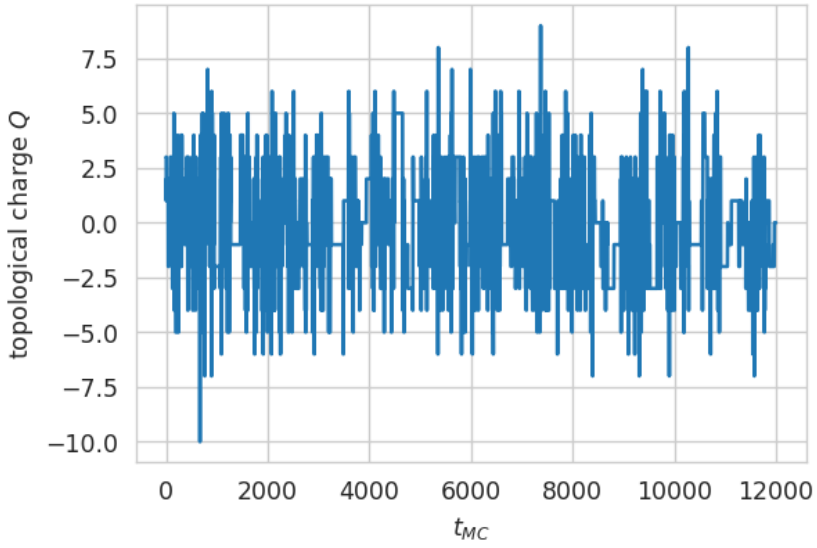


Figure 4.4: The topological charge Q over 12,000 Markov steps, after 600,000 steps of thermalization, for a network trained on a $L/a = 16$ $2D$ lattice with $n_L = 4$ multi-boson fields. Further information on the network can be found in the last row of table 4.1. Compared to figure 4.3, Q has some noticeable plateaus.

calculate the loss function, and its first moment in the model parameters, which are required by the optimization algorithm

$$\begin{aligned} D_{KL}(p||e^{-S}) &= e^{-S(U,\phi)}\left(S(U,\phi) - \log(p(U,\phi))\right) \\ \partial_p D_{KL}(p||e^{-S}) &= e^{-S(U,\phi)}\partial_p S(U,\phi)\left(1 + (\log(p(U,\phi)) - S(U,\phi))\right) \\ &- e^{-S(U,\phi)} - \sum_{i=1}^{n_L} \partial_p \log\left|\det\left(\frac{df_i(z_{i-1},\Theta)}{dz_{i-1}}\right)\right| . \end{aligned} \tag{4.18}$$

Since accurate approximations of Q^{-1} usually require $n_L > 10^2$, this scaling requirement could be prohibitive in realistic applications. A model implementation with better GPU memory scaling in n_L is needed. Fortunately, since the Lüscher action 4.10 includes no joint terms between different multi-boson fields, S and $\partial_\Theta S$ can be calculated as

$$S(U,\phi) = S_G(U) + \sum_{k=1}^{n_L} S_k\,, \quad \partial_\Theta S(U,\phi) = \partial_\Theta S_G(U) + \sum_{k=1}^{n} \partial_\Theta S_k\,. \tag{4.19}$$

By calculating each term in the sums over k separately, the need to store more than one multi-boson field at a time can be avoided. Likewise, since the model used here does not condition the updates to one multi-boson field on another, effectively sampling every field

independently, the terms

$$\partial_\Theta \log \left| \det \left(\frac{df_i(z_{i-1}, \Theta)}{dz_{i-1}} \right) \right| \tag{4.20}$$

from calculating the gradient of the distribution p can also be evaluated separately for each multi-boson field.

When calculating fermionic observables in the Lüscher theory, one deals exclusively with source term derivatives of the form

$$\partial_J^a S = \sum_{k=1}^{n_L} \partial_J^a \phi_k^\dagger \left(Q' + J - \mu_k\right)^2 \phi_k + \nu_k^2 \phi_k^\dagger \phi_k \,, \tag{4.21}$$

so terms for different multi-boson fields can again be calculated in isolation. Thus, the model's memory problems could be alleviated, but it would require implementing a custom version of the PyTorch methods for training routines that use equations 4.19 and 4.20. There was no time left to do this in the scope of this thesis.

4.4.2 Dirichlet boundary conditions

Any exploitation of the locality of the Lüscher action for multilevel configuration generation schemes would likely require the model to operate with Dirichlet boundary conditions. Thus, testing the ability of flow-based models in sampling the Lüscher action with Dirichlet, rather than periodic boundary conditions would seem like the logical next step in developing the Lüscher technique to improve the lattice volume scaling problem for machine learning models. In [65] flow-based samplers of pure gauge links with fixed boundary values already proved to be effective in the conventional $2D$ Schwinger model, which seems encouraging. Building such a model would require defining a new training loop that includes drawing the Dirichlet boundary conditions to be trained on from some distribution. If the ultimate aim is to build flow-based models that sample the lattice in small patches, the boundary conditions the model trains on would likely need to be drawn from a distribution given by the action S. So, to train a model with Dirichlet boundary conditions on a lattice patch of size L/a, it would be necessary to first create a sampler for the Dirichlet boundaries. The obvious candidate would be a flow-based model trained on periodic boundary conditions, which draws configurations, and discards all field values except those at a box-shaped boundary on the lattice. This was what was done in [65]. The networks used in this chapter, potentially with further refinements, could serve this purpose. A link to a code repository with the trained networks

and the code to train more can be found in appendix A.

4.4.3 Lüscher action in multilevel schemes

Sampling the lattice in small patches may lead to a slow variation of the long-distance physics with Markov time. Since such fluctuations make up the bulk of the noise on many correlation functions, this may be acceptable in many applications. However, one could hope to go further and sample fluctuations on different physical scales using multilevel integration schemes with multiple flow-based models. As an illustrative example, one could start with a "level-0" model trained to sample a lattice of volume L_0^4 with lattice spacing $a = a_0$, with periodic boundary conditions. Then, a "level-1" model is trained. Its task is to take in small blocks of the configurations generated by the level-0 model and apply an "inverse" renormalization transformation to them to generate a proposal for a L_1^4 lattice with $a = a_1 = \frac{a_0}{8}$ and Dirichlet boundary conditions. This is then repeated for every block of the L_0^4 configuration. Machine learning models have demonstrated the ability to reverse renormalization transformations in the past. In [69], a model successfully learned to take in 2D configurations for a ϕ^4 scalar action generated on a $L/a = 32$ lattice that had been transformed through renormalization into $L/a = 16$ configurations, and upscale them into $L/a = 32$ configurations again. This mirrors the success of ML techniques like image enhancement, where seemingly missing information in low-resolution images is filled in by ML models, creating higher-resolution images. In this way, it might be possible to generate configurations on very large lattice volumes encompassing interactions across multiple energy scales, with the level-0 model providing the physics closest to the infrared, and level-1 or possibly level-2, level-3, or higher models providing the physics closest to the ultraviolet. To train a level-1 or higher model, one would first train an auxiliary model to provide configurations that can be transformed to a lower renormalization scale. The loss function used in training would then be the D_{KL} divergence between the original configuration, and the reconstruction provided by the level-1 model.

4.4.4 Full QCD

In theory, flow-based network architectures using plaquette and fermionic PTC affine coupling layers straightforwardly generalize to non-Abelian gauge groups and $4D$ lattices. This was practically demonstrated in [70], which implemented sampling for the four-dimensional $n_f = 2$ QCD action on a $L/a = 4$ lattice. The $U(1)$ plaquettes in equation 3.16 are exchanged for

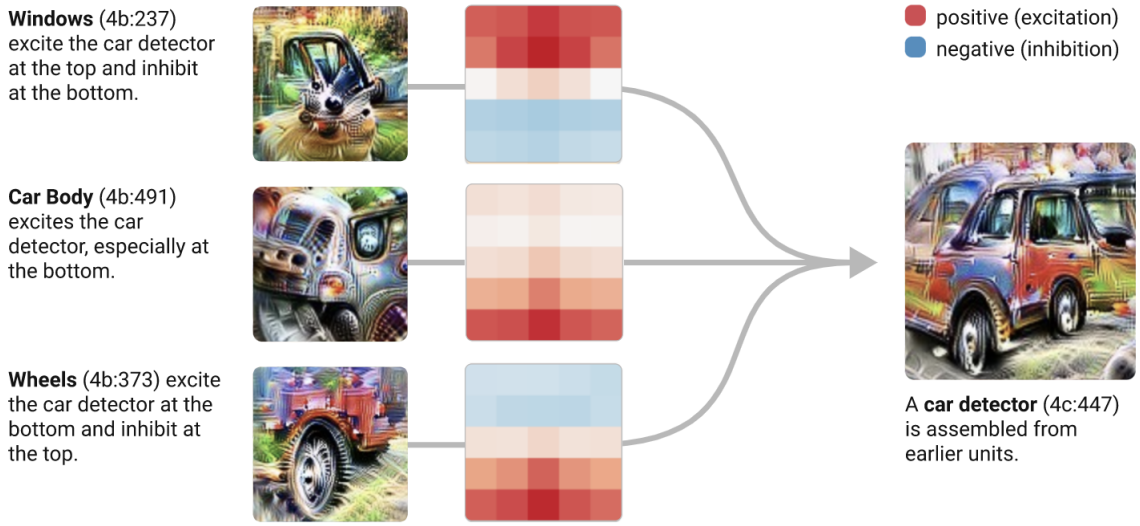


Figure 4.5: Features in one layer forming a car detector in the next layer of the InceptionV1 vision network, credit [71]

their $SU(3)$ counterparts. The number of parallel transported variables included in $\mathbb{L}$ from equation 3.22 grows to $s \times (2D + 1) = 36$. Masking patterns need to be adjusted to the new geometry. Likewise, the Lüscher action was originally defined for use in full QCD. Thus, should the volume scaling problem be solved for $2D$ lattices in the Schwinger model through any of the avenues proposed here, or another, one might be optimistic about the prospects of extending the solution to full QCD.

4.4.5 Alternative route: Model interpretability

An alternative route towards creating trivializing mappings of lattice QCD at large lattice volumes might be to look inside neural networks that successfully sample from S_{QCD} at small L/a, and reverse-engineer and understand the trivializing mappings these networks have learned. Neural networks are mostly still black boxes to us. While their parameter configurations entirely determine their behavior, the algorithms implemented by these parameters are usually not understood by their designers or anyone else. However, over the past years, there has been an increase in attempts to interpret the hidden layers of neural networks and form mechanistic models of their function. These attempts are starting to meet some success. As an illustrative example, the early hidden layers of image labeling networks have been found to include functions that implement Gabor filters, separating high and low frequencies in the input image. Deeper layers in the network then use these filters to imple-

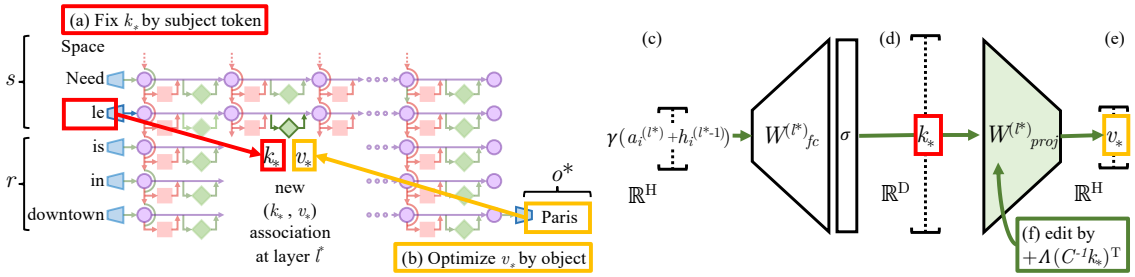


Figure 4.6: By modifying select weights in a multilayer perceptron, the GPT-3 language model can be made to "believe" that the Space Needle is in Paris. The weights were identified through causal tracing techniques. Source: [73]

ment algorithms for edge detection, which yet deeper layers use to implement curve detectors. Even deeper layers then combine these results to implement algorithms for the detection of specific concepts and objects, such as a function that determines whether the image contains the face of a dog.

As another example, [72] reverse-engineered a small network trained to perform modular addition, and was able to extract and understand the complete end-to-end algorithm the network learned to carry out this task. This algorithm turned out to be based on the periodicity of trigonometric functions, first implementing a Fourier transformation of the input numbers, then performing the modular addition through the application of trigonometric identities, then finally applying an inverse Fourier transform to create the output number. At bigger scales, the 2022 paper introducing Rank-One Model Editing (ROME) [73], demonstrated a systematic technique to localize and change the knowledge contained in GPT-3, a transformer-based neural network trained to complete text prompts with 175 billion parameters. E.g., using ROME, one can find the exact weights in the network that encode the fact that the Eiffel Tower is located in Paris, and then edit them to say that the Eiffel Tower lies in Philadelphia instead. In future queries involving the location of the Eiffel Tower, the model will then assert that it is located in Philadelphia. This growing field of neural network interpretability is creating a toolbox of mathematical techniques and code bases for understanding and reverse engineering the functionality of neural networks. One could attempt to use this toolbox of techniques to peer into the NN weights used in the affine coupling layers of QCD normalizing flow models, and reverse engineer an analytical, general algorithm to map trivial distributions to the distribution of the QCD action. Such a project could represent a significant advance in the fields of both QCD and Machine Learning. An extracted analytical formula may be far less costly to implement, and far easier to generalize to greater precision

and larger volume, than a trained neural network. Thus, it could enable progress in precision lattice QCD calculations. The mere existence and shape of such an algorithm might also lead to an improved understanding of QCD as a theory.

On the Machine Learning side, while interpretability techniques have been successfully used to reverse-engineer some NN algorithms, to date, there is not yet a published case of successfully extracting an algorithm that solves a problem for which humans did not already have an analytical solution. Success in this endeavor would thus mark a significant milestone in the advance of NN interpretability as a field, and new interpretability techniques invented to accomplish this goal might quickly find application across many fields of science.

A generally successful strategy in interpretability projects to date has been to look at the outputs of models at intermediary hidden layers, rather than just their final outputs. In the case of flow-based models, one form this could take could be investigating $p_l(z_0)$, the distribution of field configurations after applying only the model layers up to layer l. Characterizing this distribution and comparing it to distributions found at bigger and smaller l may reveal some internal hierarchy the flow-based model uses to implement the interactions of the action layer by layer. Concretely, this could, e.g., take the form of fitting parameterized actions $p_l = e^{-S_l(U_l,\phi_l)}$ to the distributions, then investigating the terms that appear in these S_l. Such fits may be performed using classic optimization techniques, or other neural networks. Once formed, hypotheses about the functionality of parts of the algorithm implemented by the flow model may benefit from validation using frameworks like Causal Scrubbing [74].

In this approach, there does not seem to be an a priori reason to investigate the internals of models sampling the Lüscher action in particular. Investigation of models using a Wilson action, as in [54], or any other lattice formulation would likely prove just as fruitful in the beginning stages of such a project.

Chapter 5

Concluding remarks

We have investigated the use of the distillation operator for cheaper low-mode averaging. Low-mode averaging requires solving for the low-lying eigenmodes of the Dirac operator, which is expensive. The spatial Laplacian operator is smaller and its low-lying eigenmodes thus cheaper to calculate, while their successful use in smearing suggests that they may have high overlap with the low energy modes of correlators. We tested the noise reduction capability of distillation low-mode averaging for the connected and disconnect contributions to the pseudoscalar and vector correlators on two CLS data sets with two flavors of $\mathcal{O}(a)$ improved dynamical Wilson fermions. The Laplacian eigenmodes exhibited insufficient overlap with the correlators to make the technique cost-effective, requiring too many modes to be calculated. However, a trivial change to the spatial Laplace operator, multiplying it with a real scalar prefactor, greatly increased the overlap of its eigenspectrum with the correlators. While still falling short of practical relevance, this suggests that the overlap is sensitive to small changes in the structure of the operator. Thus, future work may be able to find other small modifications to the distillation operator that greatly enhance its overlap with observables.

We also tested the ability of generative flow-based models based on neural networks to sample from the Lüscher action of the 2D U(1) Schwinger model. Flow-based machine learning models using gauge equivariant layers have recently become able to sample from the four-dimensional QCD action at very small lattice volumes, replacing HMC algorithms in the proposal step of MMC sampling. To enable the use of flow-based samplers on very large lattices of $L/a = 128$ or more, it may be convenient to decompose the lattice into many subdomains, enabling independent sampling of smaller lattice sections to improve the scaling of computational requirements with the volume. The Lüscher action technique potentially

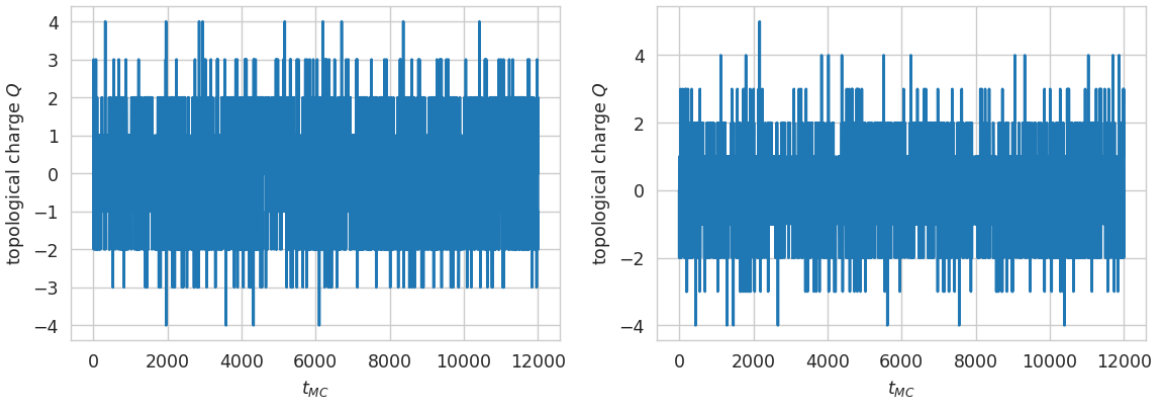
enables this. It approximates the inverse Dirac operator through a polynomial, recovering a local action at the cost of introducing many additional flavors of pseudofermions. When originally developed in the 1990s, the Lüscher action proved intractable to sample from with classical methods. In contrast, flow-based models we trained for $< 20,000$ update steps on a single Google Colab GPU proved able to sample from the Lüscher action for the Schwinger model on a $L/a = 8$ lattice with $n_L = 12$ Lüscher flavors, and on a $L/a = 16$ lattice with $n_L = 4$ Lüscher flavors. The models use the same fermionic layers for all the Lüscher fields, by treating their action parameters as additional inputs. Thus, their total parameter count is not extensive in n_L. The displayed scaling of the configuration acceptance rate with n_L likely falls short of what would be required for immediate practical use since a good approximation of the Dirac operator can require hundreds of Lüscher flavors. However, there seem to be many avenues present for improving on the proof-of-principles results found here, through improvements to the model architecture, better GPU memory allocation in training, and additional computational investment into training.

Appendix A

Topological charge over Markov time

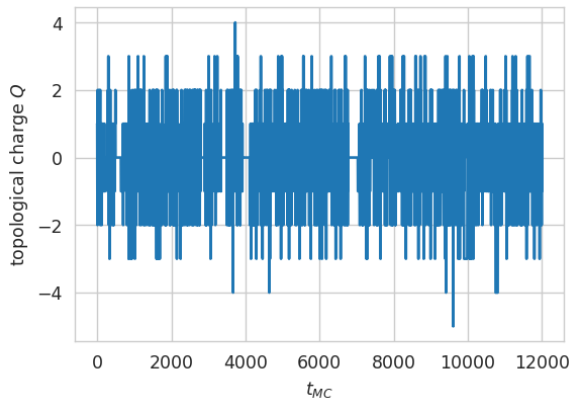
The following figures show the topological charge over 12,000 Markov steps after 600,000 steps of thermalization, for the flow-based samplers of the 2D U(1) Schwinger action with n_L multi-boson fields, described in table 4.1. As the degrees of freedom in the action increase, either through sampling more multi-boson fields n_L or large lattice volumes $(L/a)^2$, the topological charge freezes more frequently. This is due to the acceptance rate dropping, as depicted in figure 4.2, due to the sampler getting "stuck" on some configurations for longer stretches of Markov time.

The trained networks, as well as the code to create and evaluate them, can be found at https://github.com/LuciusBushnaq/PhDQCD under the GNU General Public License v3.0.

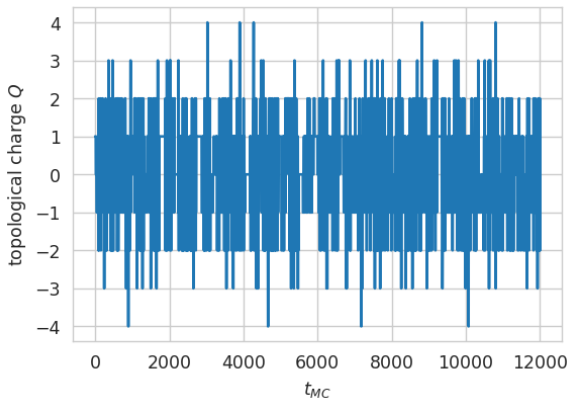


(a) $n_L = 2$ multi-boson fields, $L/a = 8$ lattice, $\kappa = 0.265$.

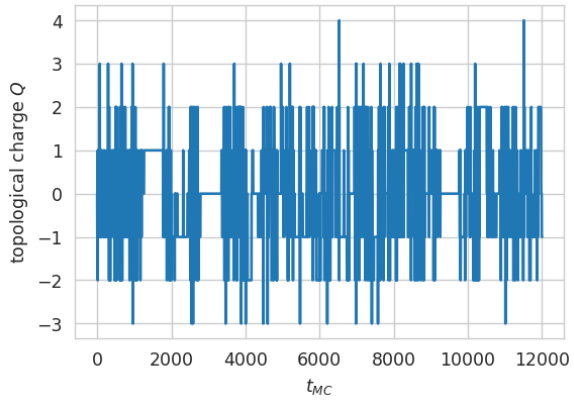
(b) $n_L = 4$ multi-boson fields, $L/a = 8$ lattice, $\kappa = 0.265$.



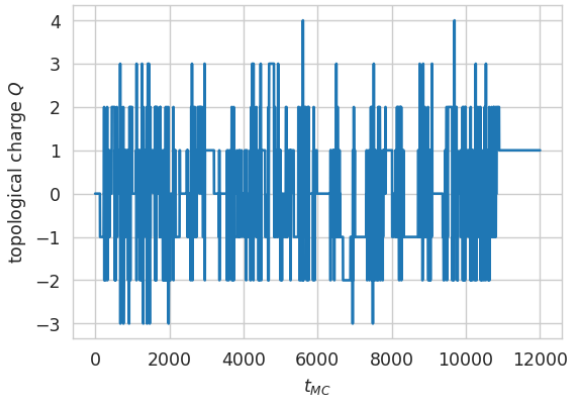
(a) $n_L = 6$ multi-boson fields, $L/a = 8$ lattice, $\kappa = 0.265$.



(b) $n_L = 8$ multi-boson fields, $L/a = 8$ lattice, $\kappa = 0.265$.



(a) $n_L = 10$ multi-boson fields, $L/a = 8$ lattice, $\kappa = 0.265$.



(b) $n_L = 12$ multi-boson fields, $L/a = 8$ lattice, $\kappa = 0.265$.

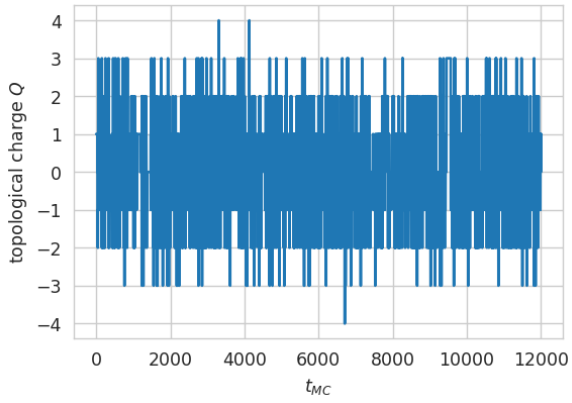
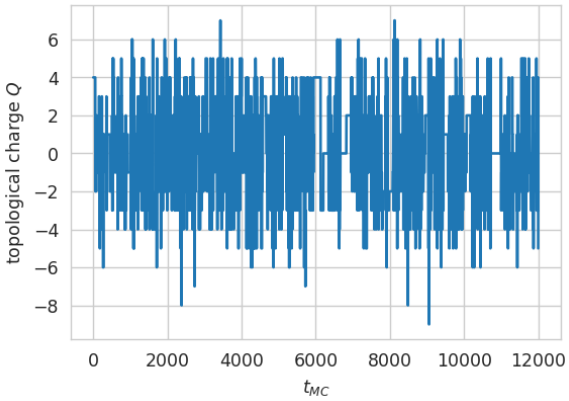
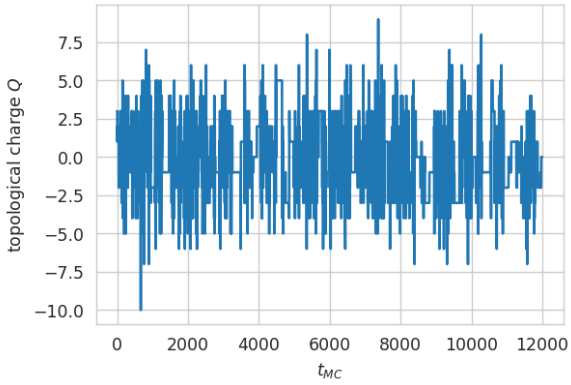


Figure A.4: $n_L = 8$ multi-boson fields, $L/a = 8$ lattice, $\kappa = 0.125$.



(a) $n_L = 2$ multi-boson fields, $L/a = 16$ lattice, $\kappa = 0.265$.



(b) $n_L = 4$ multi-boson fields, $L/a = 16$ lattice, $\kappa = 0.265$.

Appendix B

Error analysis

B.1 Chapter 2

For both of the CLS ensembles described in table 2.1 and used throughout this chapter, gauge configurations were chosen spaced 40 Molecular Dynamics Units [MDU] apart. For the A5 ensemble, the exponential autocorrelation time τ_{exp} associated with the slowest decaying mode with the smallest eigenvalue of the transition matrix λ_1 satisfies $\tau_{\text{exp}} = -\frac{1}{\log(\lambda_1)} = 25\,\text{MDU}$ [75, 76]. For the E5 ensemble, $\tau_{\text{exp}} = -\frac{1}{\log(\lambda_1)} = 50\,\text{MDU}$, and the integrated autocorrelation for the pion mass is $\tau_{\text{int}} = 9\,\text{MDU}$ [75]. The results in this chapter only involve error estimates of the pseudoscalar correlator, vector correlator, fractions of either projected onto subspaces of the distillation operator spectrum, and, for cross-validation purposes of the program setup, effective masses of the pseudoscalar, vector, and axial vector correlator in section 2.3. Since none of these quantities seemed likely to couple strongly to the slowest-decaying modes of the transition matrix, unlike for example, the topological charge, it was assumed that autocorrelations were small, and would not have a relevant effect on the conclusions drawn. The gauge configurations were thus treated as independent measurements in the error analysis.

B.1.1 Correlators

In all plots of correlation functions $C(t)$, $C_{dist}(t)$, $C_{rest}(t)$, etc., and correlation function ratios $\frac{C_{dist}(t)}{C(t)}$ in this chapter, such as those depicted in figures 2.15, 2.18 or 2.27, the quantities and their error ratios were estimated through jackknife resampling over the configurations in the

gauge ensembles. That is, for an observable x, jackknife replicates

$$x_i^{\text{jack}} = \frac{1}{n-1} \sum_{g \neq i} x_g \,, \tag{B.1}$$

over the $n = 30$ gauge configurations $g = 0, ..., n-1$, the average of these replicates

$$\overline{x}^{\text{jack}} = \frac{1}{n} \sum_i x_i^{\text{jack}} \,, \tag{B.2}$$

and their variance

$$\sigma^2 \left(x^{\text{jack}} \right) = \frac{1}{n} \sum_i (x_i^{\text{jack}} - \overline{x}^{\text{jack}})^2 \tag{B.3}$$

were calculated. For correlator ratios, the bias-corrected estimator

$$\hat{x} = n\overline{x} - (n-1)\overline{x}^{\text{jack}} \,, \tag{B.4}$$

was used as the estimate of correlator ratios $\frac{C_{dist}(t)}{C(t)}$, where $\overline{x}$ is the sample mean

$$\overline{x} = \frac{1}{n} \sum_g x_g \,. \tag{B.5}$$

For correlators $C(t)$, $\overline{x} = \overline{x}^{\text{jack}}$, so the estimator equals the sample mean.

The errors on correlators $C(t)$ and correlator ratios $\frac{C_{dist}(t)}{C(t)}$ were then both estimated using the variance of the jackknife replicates

$$\sigma(\hat{x}) = \sqrt{(n-1)}\sigma \left(x^{\text{jack}} \right) \,. \tag{B.6}$$

Correlations between the errors at different time slices were not accounted for.

B.1.2 Effective masses

The error analyses for the effective masses M_{eff} and the fits to their plateau regions depicted in figures 2.1, 2.2, 2.5 and 2.6 were likewise conducted through jackknife resampling.

For the pseudoscalar and vector correlator, functions

$$A \cosh \left(\text{M}_{\text{eff}} (t/a - L_0/(2a)) \right) \tag{B.7}$$

were first fit to each pair of neighbouring points $C(t)$, $C(t+a)$ using non-linear least squares

optimisation, provided by scipy.optimize.curve_fit [77]. The error estimate on C needed for the least squares fit was provided by the analysis detailed in the previous paragraph. The fit was carried out for all $n = 30$ Jackknife replicas of all neighboring point pairs, and $\mathrm{M}_{\mathrm{eff}}(t)$ and $\sigma(\mathrm{M}_{\mathrm{eff}}(t))$ were then estimated using B.2 and B.3.

Using the resulting effective mass curves, suitable plateau regions for fitting the mass were identified by eye, as depicted in the plots. In these regions, non-linear least squares fits to 2.15 using all points in the plateau regions were performed, again using Jackknife replicas and equations B.2, B.3.

The analysis of the axial vector correlator proceeded identically, substituting sinh for cosh in the fit.

B.2 Chapter 4

The observables calculated in this chapter, that is, the average plaquette $E(P)$ and topological susceptibility $E(Q^2)$ for the generated ensembles as depicted in table 4.1 and figure 4.2, were estimated analogously to [60], through bootstrapping with binning.

That is, the the Markov chain of length n was separated into $\frac{n}{s_{\mathrm{bin}}} = N_{\mathrm{bin}}$ bins of size s_{bin}, and the observables in each bin were averaged

$$x_i^b = \frac{1}{s_{\mathrm{bin}}} \sum_{j=0}^{s_{\mathrm{bin}}-1} x_{i \times s_{\mathrm{bin}}+j} \,. \tag{B.8}$$

Here, x_i denotes the observable at Markov time i. Then, N_{boot} bootstrap replicas of length N_{bin} were created by re-sampling $N_{\mathrm{boot}} \times N_{\mathrm{bin}}$ times in total from the set of x_i^b. Each bootstrap replica is averaged, yielding N_{boot} averages x_r^B. The observable is then estimated as

$$\overline{x}^B = \frac{1}{N_{\mathrm{boot}}} \sum_{r=0}^{N_{\mathrm{boot}}-1} x_r^B \,. \tag{B.9}$$

and the variance on it as

$$\sigma^2(x^B) = \frac{1}{N_{\mathrm{boot}}} \sum_{r=0}^{N_{\mathrm{boot}}-1} \left(x_r^B - \overline{x}^B\right)^2 . \tag{B.10}$$

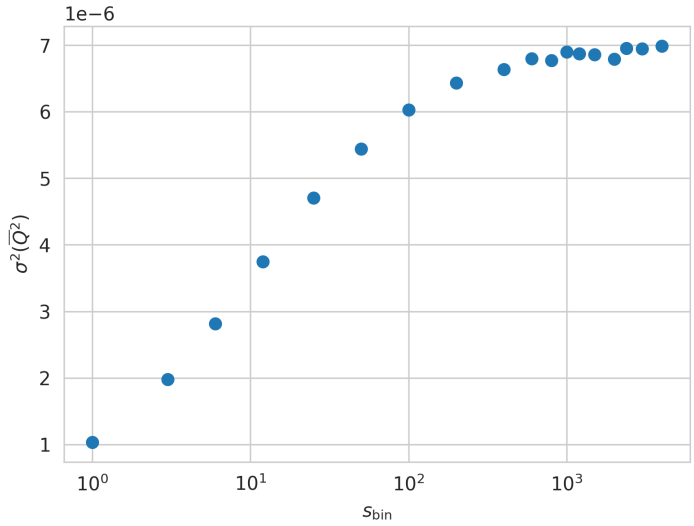


Figure B.1: The variance of the mean of the topological susceptibility $\sigma^2(\overline{Q}^2)$, calculated with increasing bin sizes s_{bin} on a Markov chain with $3,600,000$ points generated for a $L/a = 8, \kappa = 0.265$ Lüscher action with $n_L = 4$ multi-boson fields. The first $n_{\text{therm}} = 600,000$ points were discarded for thermalization. The error plateaus around $s_{\text{bin}} = 600$.

To choose the bin size, the variance of the mean of binned Markov samples

$$\sigma^2(\overline{x}^b) = \frac{1}{N_{\text{bin}}(N_{\text{bin}}-1)} \sum_{i=0}^{N_{\text{bin}}} \left(x_i^b - \overline{x}^b\right)^2, \text{with } \overline{x}^b = \frac{1}{N_{\text{bin}}} \sum_{i=0}^{N_{\text{bin}}} x_i^b \tag{B.11}$$

was calculated for increasing bin sizes s_{bin} while holding the number of Markov samples n constant. As s_{bin} increases, the systematic underestimate of the variance provided by $\sigma^2(\overline{x}^b)$ decreases, plateauing once $s_{\text{bin}} >> \tau_{\text{int}}$. The ratio between $\sigma^2(\overline{x}^b)$ calculated at the plateau and $\sigma^2(\overline{x}^b)$ calculated at $s_{\text{bin}} = 1$ is then an estimator of $2\tau_{int}$ for this observable, because the autocorrelations increase the variance on the observable over the estimate provided by the variance of the sample mean by a factor of $2\tau_{int}$ [78]. s_{bin} was then chosen large enough to lie in the plateau region for both $E(P)$ and $E(Q^2)$. Figure B.1 depicts this for the topological susceptibility in an ensemble generated by the $L/a = 8, n_L = 4$ network.

The first $n_{\text{therm}} = 600,000$ steps in the Markov chain were discarded for thermalization, ensuring $n_{\text{therm}} >> \tau_{int}$ for all observables. The number of boots n_{boot} was chosen to be $n_{\text{boot}} = 10,000$ for all measurements, and the distribution of bootstrap samples looked to be close to Gaussian at that number. Figure B.2 depicts the distribution for the topological susceptibility in an ensemble generated by the $L/a = 8, n_L = 4$ network.

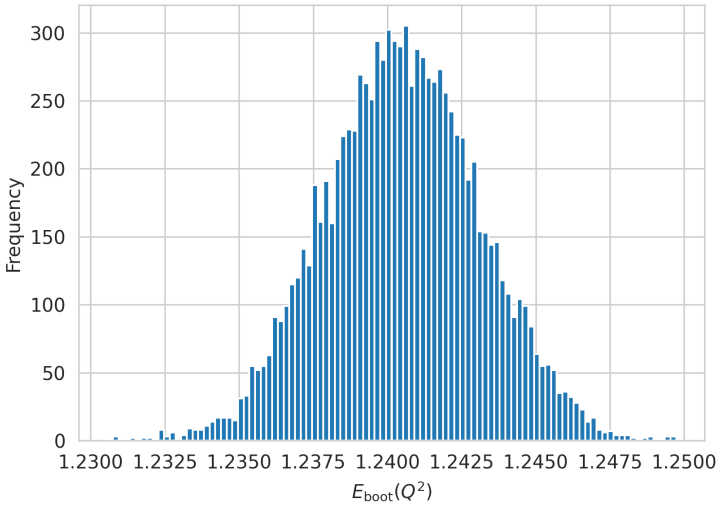


Figure B.2: Histogram of the means $E_{\text{boot}}(Q^2)$ for $n_{\text{boot}} = 10,000$ boots drawn from an ensemble of $3,600,000$ MCMC samples, generated for a $L/a = 8, \kappa = 0.265$ Lüscher action with $n_L = 4$ multi-boson fields. The first $n_{\text{therm}} = 600,000$ samples were discarded for thermalization, and the remainder binned into $5,000$ block samples, from which the boots were drawn. The sample looks Gaussian to the bare eye, suggesting that $n_{\text{boot}} = 10,000$ is sufficient for bootstrap estimates of $E(Q^2)$ in this ensemble for our purposes.

Bibliography

[1] Christof Gattringer and Christian B. Lang. *Quantum chromodynamics on the lattice*. Vol. 788. Berlin: Springer, 2010. ISBN: 978-3-642-01849-7. DOI: 10.1007/978-3-642-01850-3.

[2] Kenneth G. Wilson. "Confinement of quarks". In: *Phys. Rev. D* 10 (8 Oct. 1974), pp. 2445–2459. DOI: 10.1103/PhysRevD.10.2445. URL: https://link.aps.org/doi/10.1103/PhysRevD.10.2445.

[3] T. Blum et al. *An update of Euclidean windows of the hadronic vacuum polarization*. 2023. DOI: 10.48550/ARXIV.2301.08696. URL: https://arxiv.org/abs/2301.08696.

[4] Simon Duane, A.D. Kennedy, Brian J. Pendleton, and Duncan Roweth. "Hybrid Monte Carlo". In: *Physics Letters B* 195.2 (1987), pp. 216–222. ISSN: 0370-2693. DOI: https://doi.org/10.1016/0370-2693(87)91197-X. URL: https://www.sciencedirect.com/science/article/pii/037026938791197X.

[5] Richard Brower et al. "Lattice QCD Application Development within the US DOE Exascale Computing Project". In: *EPJ Web of Conferences* 175 (2018). Ed. by M. Della Morte, P. Fritzsch, E. Gá miz Sánchez, and C. Pena Ruano, p. 09010. DOI: 10.1051/epjconf/201817509010. URL: https://doi.org/10.1051%2Fepjconf%2F201817509010.

[6] Justin Foley et al. "Practical all-to-all propagators for lattice QCD". In: *Computer Physics Communications* 172.3 (Nov. 2005), pp. 145–162. ISSN: 0010-4655. DOI: 10.1016/j.cpc.2005.06.008. URL: http://dx.doi.org/10.1016/j.cpc.2005.06.008.

[7] Alexei Bazavov et al. *Light-quark connected intermediate-window contributions to the muon $g-2$ hadronic vacuum polarization from lattice QCD*. 2023. arXiv: 2301.08274 [hep-lat].

[8] A.S. Kronfeld and U.-J. Wiese. "SU(N) gauge theories with C-periodic boundary conditions (I). Topological structure". In: *Nuclear Physics B* 357.2 (1991), pp. 521–533. ISSN: 0550-3213. DOI: https://doi.org/10.1016/0550-3213(91)90479-H. URL: https://www.sciencedirect.com/science/article/pii/055032139190479H.

[9] Anian Altherr et al. *$N_f = 1+2+1$ QCD+QED simulations with $C^\star$ boundary conditions*. 2022. DOI: 10.48550/ARXIV.2212.09578. URL: https://arxiv.org/abs/2212.09578.

[10] G. W. Bennett et al. "Final report of the E821 muon anomalous magnetic moment measurement at BNL". In: *Physical Review D* 73.7 (Apr. 2006). ISSN: 1550-2368. DOI: 10.1103/physrevd.73.072003. URL: http://dx.doi.org/10.1103/PhysRevD.73.072003.

[11] T. Albahri et al. "Magnetic-field measurement and analysis for the Muon $g-2$ Experiment at Fermilab". In: *Phys. Rev. A* 103 (4 Apr. 2021), p. 042208. DOI: 10.1103/PhysRevA.103.042208. URL: https://link.aps.org/doi/10.1103/PhysRevA.103.042208.

[12] T. Aoyama et al. *The anomalous magnetic moment of the muon in the Standard Model*. 2020. arXiv: 2006.04822 [hep-ph].

[13] T. Aoyama et al. "The anomalous magnetic moment of the muon in the Standard Model". In: *Physics Reports* 887 (Dec. 2020), pp. 1–166. ISSN: 0370-1573. DOI: 10.1016/j.physrep.2020.07.006. URL: http://dx.doi.org/10.1016/j.physrep.2020.07.006.

[14] Sz. Borsanyi et al. "Leading hadronic contribution to the muon magnetic moment from lattice QCD". In: *Nature* 593.7857 (Apr. 2021), pp. 51–55. DOI: 10.1038/s41586-021-03418-1. URL: https://doi.org/10.1038%2Fs41586-021-03418-1.

[15] Antoine Gé rardin et al. "The leading hadronic contribution to $(g-2)_\mu$ from lattice QCD with $N_f = 2+1$ flavors of O(a) improved Wilson quarks". In: *Physical Review D* 100.1 (July 2019). DOI: 10.1103/physrevd.100.014510. URL: https://doi.org/10.1103%2Fphysrevd.100.014510.

[16] C. T. H. Davies et al. "Hadronic-vacuum-polarization contribution to the muon's anomalous magnetic moment from four-flavor lattice QCD". In: *Physical Review D* 101.3 (Feb. 2020). DOI: 10.1103/physrevd.101.034512. URL: https://doi.org/10.1103%2Fphysrevd.101.034512.

[17] D. Giusti et al. "Electromagnetic and strong isospin-breaking corrections to the muon $g-2$ from lattice QCD+QED". In: *Physical Review D* 99.11 (June 2019). DOI: 10.1103/physrevd.99.114502. URL: https://doi.org/10.1103%2Fphysrevd.99.114502.

[18] T. Blum et al. "Calculation of the Hadronic Vacuum Polarization Contribution to the Muon Anomalous Magnetic Moment". In: *Physical Review Letters* 121.2 (July 2018). DOI: 10.1103/physrevlett.121.022003. URL: https://doi.org/10.1103%2Fphysrevlett.121.022003.

[19] Sz. Borsanyi et al. "Hadronic Vacuum Polarization Contribution to the Anomalous Magnetic Moments of Leptons from First Principles". In: *Physical Review Letters* 121.2 (July 2018). DOI: 10.1103/physrevlett.121.022002. URL: https://doi.org/10.1103%2Fphysrevlett.121.022002.

[20] M. Davier, A. Hoecker, B. Malaescu, and Z. Zhang. "A new evaluation of the hadronic vacuum polarisation contributions to the muon anomalous magnetic moment and to $\alpha(m_Z^2)$". In: *The European Physical Journal C* 80.3 (Mar. 2020). DOI: 10.1140/epjc/s10052-020-7792-2. URL: https://doi.org/10.1140%2Fepjc%2Fs10052-020-7792-2.

[21] Alexander Keshavarzi, Daisuke Nomura, and Thomas Teubner. "The $g-2$ of charged leptons, $\alpha(M_Z^2)$, and the hyperfine splitting of muonium". In: *Physical Review D* 101.1 (Jan. 2020). DOI: 10.1103/physrevd.101.014029. URL: https://doi.org/10.1103%2Fphysrevd.101.014029.

[22] Gilberto Colangelo, Martin Hoferichter, and Peter Stoffer. "Two-pion contribution to hadronic vacuum polarization". In: *Journal of High Energy Physics* 2019.2 (Feb. 2019). DOI: 10.1007/jhep02(2019)006. URL: https://doi.org/10.1007%2Fjhep02%282019%29006.

[23] Martin Hoferichter, Bai-Long Hoid, and Bastian Kubis. "Three-pion contribution to hadronic vacuum polarization". In: *Journal of High Energy Physics* 2019.8 (Aug. 2019). DOI: 10.1007/jhep08(2019)137. URL: https://doi.org/10.1007%2Fjhep08%282019%29137.

[24] C. Alexandrou et al. *Lattice calculation of the short and intermediate time-distance hadronic vacuum polarization contributions to the muon magnetic moment using twisted-*

mass fermions. 2022. DOI: 10.48550/ARXIV.2206.15084. URL: https://arxiv.org/abs/2206.15084.

[25] Davide Giusti and Silvano Simula. *Window contributions to the muon hadronic vacuum polarization with twisted-mass fermions*. 2021. arXiv: 2111.15329 [hep-lat].

[26] M. Cè et al. "Window observable for the hadronic vacuum polarization contribution to the muon $g-2$ from lattice QCD". In: *Physical Review D* 106.11 (Dec. 2022). DOI: 10.1103/physrevd.106.114502. URL: https://doi.org/10.1103%2Fphysrevd.106.114502.

[27] Christopher Aubin et al. "Light quark vacuum polarization at the physical point and contribution to the muon $g-2$." In: *Physical Review D* 101.1 (Jan. 2020). DOI: 10.1103/physrevd.101.014503. URL: https://doi.org/10.1103%2Fphysrevd.101.014503.

[28] G. Colangelo et al. "Data-driven evaluations of Euclidean windows to scrutinize hadronic vacuum polarization". In: *Physics Letters B* 833 (2022), p. 137313. ISSN: 0370-2693. DOI: https://doi.org/10.1016/j.physletb.2022.137313. URL: https://www.sciencedirect.com/science/article/pii/S0370269322004476.

[29] T. Blum. "Lattice Calculation of the Lowest-Order Hadronic Contribution to the Muon Ano malous Magnetic Moment". In: *Physical Review Letters* 91.5 (July 2003). ISSN: 1079-7114. DOI: 10.1103/physrevlett.91.052001. URL: http://dx.doi.org/10.1103/PhysRevLett.91.052001.

[30] Eduardo de Rafael. "Hadronic contributions to the muon g-2 and low-energy QCD". In: *Physics Letters B* 322.3 (Feb. 1994), pp. 239–246. ISSN: 0370-2693. DOI: 10.1016/0370-2693(94)91114-2. URL: http://dx.doi.org/10.1016/0370-2693(94)91114-2.

[31] David Bernecker and Harvey B. Meyer. "Vector correlators in lattice QCD: Methods and applications". In: *The European Physical Journal A* 47.11 (Nov. 2011). ISSN: 1434-601X. DOI: 10.1140/epja/i2011-11148-6. URL: http://dx.doi.org/10.1140/epja/i2011-11148-6.

[32] Thomas DeGrand and Stefan Schaefer. "Improving meson two-point functions in lattice QCD". In: *Computer Physics Communications* 159.3 (June 2004), pp. 185–191. ISSN: 0010-4655. DOI: 10.1016/j.cpc.2004.02.006. URL: http://dx.doi.org/10.1016/j.cpc.2004.02.006.

[33] L Giusti et al. "Low-energy couplings of QCD from current correlators near the chiral limit". In: *Journal of High Energy Physics* 2004.04 (Apr. 2004), pp. 013–013. ISSN: 1029-8479. DOI: 10.1088/1126-6708/2004/04/013. URL: http://dx.doi.org/10.1088/1126-6708/2004/04/013.

[34] Michael Peardon et al. "Novel quark-field creation operator construction for hadronic physics in lattice QCD". In: *Physical Review D* 80.5 (Sept. 2009). ISSN: 1550-2368. DOI: 10.1103/physrevd.80.054506. URL: http://dx.doi.org/10.1103/PhysRevD.80.054506.

[35] Martin Lüscher. "Trivializing Maps, the Wilson Flow and the HMC Algorithm". In: *Communications in Mathematical Physics* 293.3 (Nov. 2009), pp. 899–919. DOI: 10.1007/s00220-009-0953-7. URL: https://doi.org/10.1007%2Fs00220-009-0953-7.

[36] M. S. Albergo, G. Kanwar, and P. E. Shanahan. "Flow-based generative models for Markov chain Monte Carlo in lattice field theory". In: *Physical Review D* 100.3 (Aug. 2019). DOI: 10.1103/physrevd.100.034515. URL: https://doi.org/10.1103%2Fphysrevd.100.034515.

[37] Gurtej Kanwar et al. "Equivariant Flow-Based Sampling for Lattice Gauge Theory". In: *Physical Review Letters* 125.12 (Sept. 2020). DOI: 10.1103/physrevlett.125.121601. URL: https://doi.org/10.1103%2Fphysrevlett.125.121601.

[38] Ryan Abbott et al. *Sampling QCD field configurations with gauge-equivariant flow models*. 2022. arXiv: 2208.03832 [hep-lat].

[39] Martin Lüscher. "A new approach to the problem of dynamical quarks in numerical simulations of lattice QCD". In: *Nuclear Physics B* 418.3 (May 1994), pp. 637–648. DOI: 10.1016/0550-3213(94)90533-9. URL: https://doi.org/10.1016%2F0550-3213%2894%2990533-9.

[40] Gunnar S. Bali et al. "Masses and decay constants of the η and η' mesons from lattice QCD". In: *Journal of High Energy Physics* 2021.8 (Aug. 2021). DOI: 10.1007/jhep08(2021)137. URL: https://doi.org/10.1007%2Fjhep08%282021%29137.

[41] Constantia Alexandrou et al. *Generalized parton distributions of the proton from lattice QCD*. 2021. arXiv: 2111.03226 [hep-lat].

[42] C. McNeile and C. Michael. "Decay width of light quark hybrid meson from the lattice". In: *Physical Review D* 73.7 (Apr. 2006). ISSN: 1550-2368. DOI: 10.1103/physrevd.73.074506. URL: http://dx.doi.org/10.1103/PhysRevD.73.074506.

[43] Shao-Jing Dong and Keh-Fei Liu. "Stochastic estimation with Z2 noise". In: *Physics Letters B* 328.1-2 (May 1994), pp. 130–136. DOI: 10.1016/0370-2693(94)90440-5. URL: https://doi.org/10.1016%2F0370-2693%2894%2990440-5.

[44] RBC & UKQCD collaboration et al. "Use of stochastic sources for the lattice determination of light quark physics". In: *Journal of High Energy Physics* 2008.08 (Aug. 2008), pp. 086–086. ISSN: 1029-8479. DOI: 10.1088/1126-6708/2008/08/086. URL: http://dx.doi.org/10.1088/1126-6708/2008/08/086.

[45] Lingfei Wu, Eloy Romero, and Andreas Stathopoulos. "PRIMME_SVDS: A High-Performance Preconditioned SVD Solver for Accurate Large-Scale Computations". In: *CoRR* abs/1607.01404 (2016). arXiv: 1607.01404. URL: http://arxiv.org/abs/1607.01404.

[46] Patrick Fritzsch et al. "The strange quark mass and Lambda parameter of two flavor QCD". In: *Nuclear Physics B* 865.3 (Dec. 2012), pp. 397–429. ISSN: 0550-3213. DOI: 10.1016/j.nuclphysb.2012.07.026. URL: http://dx.doi.org/10.1016/j.nuclphysb.2012.07.026.

[47] M. Della Morte et al. "The hadronic vacuum polarization contribution to the muon g-2 from lattice QCD". In: *Journal of High Energy Physics* 2017.10 (Oct. 2017). ISSN: 1029-8479. DOI: 10.1007/jhep10(2017)020. URL: http://dx.doi.org/10.1007/JHEP10(2017)020.

[48] C. Morningstar et al. "Improved stochastic estimation of quark propagation with Laplacian Heaviside smearing in lattice QCD". In: *Physical Review D* 83.11 (June 2011). ISSN: 1550-2368. DOI: 10.1103/physrevd.83.114505. URL: http://dx.doi.org/10.1103/PhysRevD.83.114505.

[49] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger. Vol. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[50] John Jumper et al. "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596 (Aug. 2021), pp. 1–11. DOI: 10.1038/s41586-021-03819-2.

[51] Luigi Del Debbio, Joe Marsh Rossney, and Michael Wilson. "Efficient modeling of trivializing maps for lattice ϕ^4 theory using normalizing flows: A first look at scalability". In: *Physical Review D* 104.9 (Nov. 2021). DOI: 10.1103/physrevd.104.094507. URL: https://doi.org/10.1103%2Fphysrevd.104.094507.

[52] Lilian Weng. "Flow-based Deep Generative Models". In: *lilianweng.github.io* (2018). URL: https://lilianweng.github.io/posts/2018-10-13-flow-models/.

[53] Alethea Power et al. *Grokking: Generalization Beyond Overfitting on Small Algorithmic Datasets*. 2022. DOI: 10.48550/ARXIV.2201.02177. URL: https://arxiv.org/abs/2201.02177.

[54] Ryan Abbott et al. *Gauge-equivariant flow models for sampling in lattice field theories with pseudofermions*. 2022. DOI: 10.48550/ARXIV.2207.08945. URL: https://arxiv.org/abs/2207.08945.

[55] Michael S. Albergo et al. "Flow-based sampling in the lattice Schwinger model at criticality". In: *Physical Review D* 106.1 (July 2022). DOI: 10.1103/physrevd.106.014514. URL: https://doi.org/10.1103%2Fphysrevd.106.014514.

[56] Jiahui Yu et al. *CoCa: Contrastive Captioners are Image-Text Foundation Models*. 2022. DOI: 10.48550/ARXIV.2205.01917. URL: https://arxiv.org/abs/2205.01917.

[57] Ashish Vaswani et al. *Attention Is All You Need*. 2017. DOI: 10.48550/ARXIV.1706.03762. URL: https://arxiv.org/abs/1706.03762.

[58] Ilya Tolstikhin et al. *MLP-Mixer: An all-MLP Architecture for Vision*. 2021. DOI: 10.48550/ARXIV.2105.01601. URL: https://arxiv.org/abs/2105.01601.

[59] Michael S. Albergo et al. "Flow-based sampling for fermionic lattice field theories". In: *Physical Review D* 104.11 (Dec. 2021). DOI: 10.1103/physrevd.104.114507. URL: https://doi.org/10.1103%2Fphysrevd.104.114507.

[60] Michael S. Albergo et al. *Introduction to Normalizing Flows for Lattice Field Theory*. 2021. DOI: 10.48550/ARXIV.2101.08176. URL: https://arxiv.org/abs/2101.08176.

[61] Matteo Favoni, Andreas Ipp, David I. Müller, and Daniel Schuh. "Lattice Gauge Equivariant Convolutional Neural Networks". In: *Physical Review Letters* 128.3 (Jan. 2022). DOI: 10.1103/physrevlett.128.032003. URL: https://doi.org/10.1103%2Fphysrevlett.128.032003.

[62] Mattia Dalla Brida, Leonardo Giusti, Tim Harris, and Michele Pepe. "Multi-level Monte Carlo computation of the hadronic vacuum polarization contribution to (g-2)". In: *Physics Letters B* 816 (May 2021), p. 136191. DOI: 10.1016/j.physletb.2021.136191. URL: https://doi.org/10.1016%2Fj.physletb.2021.136191.

[63] Marco Cè, Leonardo Giusti, and Stefan Schaefer. "Domain decomposition, multilevel integration, and exponential noise reduction in lattice QCD". In: *Physical Review D* 93.9 (May 2016). DOI: 10.1103/physrevd.93.094507. URL: https://doi.org/10.1103%2Fphysrevd.93.094507.

[64] Marco Cè, Leonardo Giusti, and Stefan Schaefer. "Local factorization of the fermion determinant in lattice QCD". In: *Physical Review D* 95.3 (Feb. 2017). DOI: 10.1103/physrevd.95.034503. URL: https://doi.org/10.1103%2Fphysrevd.95.034503.

[65] Jacob Finkenrath. *Tackling critical slowing down using global correction steps with equivariant flows: the case of the Schwinger model*. 2022. DOI: 10.48550/ARXIV.2201.02216. URL: https://arxiv.org/abs/2201.02216.

[66] Ilya Loshchilov and Frank Hutter. *Decoupled Weight Decay Regularization*. 2017. DOI: 10.48550/ARXIV.1711.05101. URL: https://arxiv.org/abs/1711.05101.

[67] Yuya Shimizu and Yoshinobu Kuramashi. "Critical behavior of the lattice Schwinger model with a topological term at $\theta = \pi$ using the Grassmann tensor renormalization group". In: *Physical Review D* 90.7 (Oct. 2014). DOI: 10.1103/physrevd.90.074503. URL: https://doi.org/10.1103%2Fphysrevd.90.074503.

[68] Ryan Abbott et al. *Aspects of scaling and scalability for flow-based sampling of lattice QCD*. 2022. DOI: 10.48550/ARXIV.2211.07541. URL: https://arxiv.org/abs/2211.07541.

[69] Dimitrios Bachtis, Gert Aarts, Francesco Di Renzo, and Biagio Lucini. "Inverse Renormalization Group in Quantum Field Theory". In: *Physical Review Letters* 128.8 (Feb. 2022). DOI: 10.1103/physrevlett.128.081603. URL: https://doi.org/10.1103%2Fphysrevlett.128.081603.

[70] Ryan Abbott et al. *Sampling QCD field configurations with gauge-equivariant flow models*. 2022. DOI: 10.48550/ARXIV.2208.03832. URL: https://arxiv.org/abs/2208.03832.

[71] Chris Olah et al. "Zoom In: An Introduction to Circuits". In: *Distill* 5 (3 2020). DOI: 10.23915/distill.00024.001.

[72] Neel Nanda et al. *Progress measures for grokking via mechanistic interpretability*. 2023. DOI: 10.48550/ARXIV.2301.05217. URL: https://arxiv.org/abs/2301.05217.

[73] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. *Locating and Editing Factual Associations in GPT*. 2022. DOI: 10.48550/ARXIV.2202.05262. URL: https://arxiv.org/abs/2202.05262.

[74] Lawrence Chan et al. "Causal Scrubbing: a method for rigorously testing interpretability hypotheses". In: *LessWrong online forum* (Dec. 2022). URL: https://www.lesswrong.com/s/h95ayYYwMebGEYN5y/p/JvZhhzycHu2Yd57RN.

[75] Marina Marinkovic and Stefan Schaefer. *Comparison of the mass preconditioned HMC and the DD-HMC algorithm for two-flavour QCD*. 2010. arXiv: 1011.0911 [hep-lat].

[76] Stefan Schaefer, Rainer Sommer, and Francesco Virotta. "Critical slowing down and error analysis in lattice QCD simulations". In: *Nuclear Physics B* 845.1 (Apr. 2011), pp. 93–119. DOI: 10.1016/j.nuclphysb.2010.11.020. URL: https://doi.org/10.1016%2Fj.nuclphysb.2010.11.020.

[77] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[78] Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Monographs on Statistics and Applied Probability 57. Boca Raton, Florida, USA: Chapman & Hall/CRC, 1993.