Lattice QCD Simulations towards Strong and Weak Coupling Limits

Jiqun Tu

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY
2020

Abstract

Lattice QCD Simulations towards Strong and Weak Coupling Limits

Jiqun Tu

Lattice gauge theory is a special regularization of continuum gauge theories and the numerical simulation of lattice quantum chromodynamics (QCD) remains as the only first principle method to study non-perturbative QCD at low energy. The lattice spacing a, which serves as the ultraviolet cut off, plays a significant role in determining error on any lattice simulation results. Physical results come from extrapolating a series of simulations with different values for a to $a = 0$. Reducing the size of these errors for non-zero a improves the extrapolation and minimizes the error.

In the strong coupling limit the coarse lattice spacing pushes the analysis of the finite lattice spacing error to its limit. Section 4 measures two renormalized physical observables, the neutral kaon mixing parameter B_K and the $\Delta I = 3/2$ $K \to \pi\pi$ decay amplitude A_2 on a lattice with coarse lattice spacing of $a \sim 1$ GeV and explores the a^2 scaling properties at this scale.

In the weak coupling limit the lattice simulations suffer from critical slowing down where for the Monte Carlo Markov evolution the cost of generating decorrelated samples increases significantly as the lattice spacing decreases, which makes reliable error analysis on the results expensive. Among the observables the topological charge of the configurations appears to have the longest integrated autocorrelation time. Based on a previous work where a diffusion model is proposed to describe the evolution of the topological charge, section 2 extends this model to lattices with dynamical fermions using a new numerical method that captures the behavior for different Fourier modes.

Section 3 describes our effort to find a practical renormalization group transformation to transform lattice QCD between two different scales, whose knowledge could ultimately leads to a multi-scale evolution algorithm that solves the problem of critical slowing down. For a particular choice of action, we have found that doubling the lattice spacing of a fine lattice yields observables that agree at the few precent level with direct simulations on the coarser lattice.

Section 5 aims at speeding up the lattice simulations in the weak coupling limit from the numerical method and hardware perspective. It proposes a preconditioner for solving the Dirac equation targeting the ensemble generation phase and details its implementation on currently the

fastest supercomputer in the world.

Contents

List of Figures

List of Tables

Acknowledgments

I want to express my deepest gratitude to my advisor, Robert Mawhinney, for hundreds of hours of insightful discussions and suggestions leading up to the completion of this thesis.

I would like to thank other senior mentors in our RBC/UKQCD callaborations, including Norman Christ, Peter Boyle, Chulwoo Jung, Christ Kelly, Christopher Lehner, and Taku Izubuchi. I am also deeply indebted to my fellow graduate students, including Luchang Jin, David Murphy, Duo Guo, Tianle Wang, Bigeng Wang, and Yidi Zhao.

I would like to thank our collaborators at NVIDIA Corporation, especially Kate Clark, for patiently answering all my relentless questions and providing extremely helpful hardware support.

I would aslo like to thank my Dota2 squad members, including, but not limited to, Gao Siling, Jin Jiangjun, Diaogen, Yutang Ge, Xiao Tianshi, and Xiao Wangzi, for their companionship on my way to study physics.

Lastly, and most importantly, I would like to thank my parents, for many years of love and unwavering support as I study physics tens of thousands of miles away from them.

1 Introduction

In this section a brief introduction to lattice QCD will be given. It will be more as a display of what it is in practice rather than an explanation of the motivations why things are performed in a certain way.

1.1 Quantum Chromodynamics (QCD)

Quantum Chromodynamics (QCD) is an Yang-Mills theory with SU(3) as the gauge group. Under the path integral representation the correlation functions in QCD are defined as

$$\langle \mathcal{O} \rangle = \frac{1}{Z} \int [d\bar{\psi}][d\psi][dA_\mu] \mathcal{O} \exp\left[iS[A_\mu, \bar{\psi}, \psi]\right], \tag{1}$$

where Z is the partition function

$$Z = \int [d\bar{\psi}][d\psi][dA_\mu] \exp\left[iS[A_\mu, \bar{\psi}, \psi]\right]. \tag{2}$$

The QCD action $S[A_\mu, \bar{\psi}, \psi]$ is the space-time integration of the QCD Lagrangian $\mathcal{L}(x)$

$$S = \int d^4x \; \mathcal{L}(x), \tag{3}$$

$$\mathcal{L}(x) = -\frac{1}{2} \operatorname{tr} G_{\mu\nu} G^{\mu\nu} + \bar{\psi} i \gamma^\mu D_\mu \psi + m \bar{\psi} \psi. \tag{4}$$

The various elements in this equation will be introduced below.

1.1.1 The Fermionic Fields

Technically speaking the spin-$\frac{1}{2}$ fermionic fields ψ, whose quanta are called *quarks*, are anti-commuting (Grassmann) fields that has a commuting and an anti-commuting part:

$$\psi(x) = \sum_i \theta_i \phi_i(x). \tag{5}$$

The θ_i's are anti-commute and form a basis of the anti-commuting algebra,

$$\theta_i \theta_j = -\theta_j \theta_i. \tag{6}$$

The complex valued commuting part $\phi_i(x)$ lives in the fundamental representation of the SU(3) gauge group. This index of ϕ_i, which runs from 1 to 3, is called *color* and it gives rise the word *chromo* in QCD. ϕ_i also carries a *spin* index that runs from 0 to 3 and acts as a vector in the corresponding Dirac algebra. The γ^μ's are the γ-matrices and they form a matrix representation of the Dirac algebra. Details about the lattice (Euclidean) version and Minkowski version of the γ-matrices and the Dirac algebra will be given in the appendix.

In a canonical quantization framework $\bar{\psi}$ indicates a combination of transpose, complex conjugate and a right multiplication of γ^0:

$$\bar{\psi} = \psi^\dagger \gamma^0. \tag{7}$$

As a path integral field $\bar{\psi}$ is an independent Grassmann field.

Currently there are 6 *flavors* of quarks in the theory: the up, down, charm, strange, top, and bottom quark.

1.1.2 The Bosonic Fields

The spin-1 bosonic gauge field A_μ, whose quanta are called *gluons*, lives in the adjoint representation of the gauge group and is $\mathfrak{su}(3)$-valued. $G_{\mu\nu}$ is the field strength tensor of A_μ, given by

$$G_{\mu\nu} = \partial_\mu A_\nu - \partial_\nu A_\mu - ig_s[A_\mu, A_\nu], \tag{8}$$

where g_s is the strong coupling constant. With the definition of the gauge field the covariant derivative D_μ is defined,

$$D_\mu = \partial_\mu - ig_s A_\mu. \tag{9}$$

Through out this document we will be using the lower case "tr •" to indicate taking the trace over the three $\mathfrak{su}(3)$ or SU(3) color index.

1.1.3 Gauge Invariance

Under a local SU(3) gauge transformation, denoted by $V(x)$, A_μ and ψ transform as

$$A_\mu(x) \to V(x)\left[A_\mu(x) + \frac{i}{g_s}\partial_\mu\right]V^\dagger(x), \tag{10}$$

$$\psi(x) \to V(x)\psi(x). \tag{11}$$

The Lagrangian, and thus the action is gauge invariant since they are unchanged under gauge transformations.

Local gauge transformations assign *phases* to the fermion and gauge fields locally and the idea of gauge invariance comes from the assumption that the unphysical local phase changes should not affect physical interpretation of the theory. Geometrically it shares some similarity with the idea of parallel transport on Riemannian manifolds.

1.1.4 The Index Theorem and Topological Charge

One important concept realized through the use of the γ-matrices is the idea of chiral projectors,

$$\gamma_5 = i\gamma_0\gamma_1\gamma_2\gamma_3,\ P_\pm = \frac{1}{2}(1 \pm \gamma_5). \tag{12}$$

Under the chiral representation of γ-matrices the chiral Dirac operator $\mathcal{D}$ is defined as

$$\begin{pmatrix} 0 & \mathcal{D}^\dagger \\ \mathcal{D} & 0 \end{pmatrix} \equiv i\gamma^\mu D_\mu \tag{13}$$

the famous index theorem states that

$$\mathcal{I}(\mathcal{D}) \equiv \dim\ker(\mathcal{D}) - \dim\ker(\mathcal{D}^\dagger) = \frac{g_s^2}{16\pi^2}\int d^4x\ \epsilon^{\mu\nu\alpha\beta}\,\mathrm{tr}\,G_{\mu\nu}G_{\alpha\beta}. \tag{14}$$

Here $\dim\ker(\mathcal{D})$ means the dimension of the space of all non-zero χ such that

$$\mathcal{D}\chi = 0. \tag{15}$$

$\epsilon^{\mu\nu\alpha\beta}$ is the total anti-symmetric tensor. The right hand side of the equation is known as the *topological charge* of a gauge configuration, which, and thus the index, is integer-valued and is invariant under continuous defamation of the gauge field. It states that the difference between number of the zero modes of the Dirac operator with plus chirality and those with minus chirality equals to the topological charge. The index theorem can be viewed as an application of the more general Atiyah-Singer index theorem [1] in local gauge theory. Detailed discussion and the proof of the theorem could be found in [2], as well as a variety of other literatures.

The term on the right hand side of equation (14) is also related to the strong CP problem. Gauge invariance does not prevent a term like

$$\Delta\mathcal{L}_\theta = \theta\epsilon^{\mu\nu\alpha\beta}\,\mathrm{tr}\,G_{\mu\nu}G_{\alpha\beta} \tag{16}$$

from entering the QCD Lagrangian. This term does not enter perturbation theory at any order since it is a total derivative yet a non-zero θ predicts a non-zero neutron electric dipole moment [3], which has not been observed in experiments. Why this term is strictly zero remains a theoretical problem to this day.

In lattice QCD calculations we are able to measure the topological charge distributed over the lattice explicitly and this will be the focus of section 2.

1.2 Lattice QCD

Perturbative calculations in QCD show that the strong coupling constant, $\alpha_s(\mu) = g_s^2(\mu)/4\pi$, becomes greater than 1 at Λ_{QCD}, where $\Lambda_{\mathrm{QCD}} \sim 258$ MeV. Perturbative calculations at these low energies are clearly unreliable and even at 1 GeV substantial errors are expected in QCD perturbative theory due to the size of α_s. The techniques that lead to the great theoretical success of QED fail in explaining the inner structure and interactions of light mesons and baryons that QCD should describe in detail. A non-perturbative approach is needed.

The lattice method enters as an ultraviolet regularizer of the continuum theory. The fermion and gauge field are discretized on a *lattice* and the correlation functions in equation (20) are calculated by evaluating the path integral explicitly. The lattice spacing a acts as a natural cut off and all the singularities the continuum theory has are removed.

Physical predictions of lattice QCD, however, can only derived after great numerical and theoretical challenges are overcome. These issues are discussed below.

1.2.1 Discretization of the Continuous Theory

Because of the non-perturbative nature of lattice QCD we first promote all $\mathfrak{su}(3)$-valued gauge fields $A_\mu(x)$ to SU(3) elements $U_\mu(x)$

$$U_\mu(x) = \exp[iag_sA_\mu(x)] \tag{17}$$

and for the rest of this document we will be dealing with $U_\mu(x)$.

We perform a Wick rotation such that

$$x_0 \to -ix_4. \tag{18}$$

The metric now becomes Euclidian

$$x^2 = x_\mu x_\mu = x_1^2 + x_2^2 + x_3^2 + x_4^2, \tag{19}$$

and the *weight* factor in the path integral becomes real as well

$$\langle 0|T\{\psi(x_1)\bar{\psi}(x_2)\}|0\rangle = \frac{1}{Z}\int [d\bar{\psi}][d\psi][dU_\mu]\psi(x_1)\bar{\psi}(x_2)\exp[-S[U_\mu,\bar{\psi},\psi]]. \tag{20}$$

Here the symbol S is still used to denote the Euclidian action. Given that $S[U_\mu,\bar{\psi},\psi]$ is a real valued number the path integral formulation defines a probability that can be used in a Monte Carlo algorithm. Samples of U_μ, $\bar{\psi}$ and ψ are drawn according to the probability $\exp[-S[U_\mu,\bar{\psi},\psi]]$ and the desired quantity $\psi(x_1)\bar{\psi}(x_2)$ is measured on these samples. By taking the average of the values we acquire an approximation of the true correlation function

$$\langle \mathcal{O}\rangle \simeq \frac{1}{n}\sum_j [\mathcal{O}]_j, \tag{21}$$

where $[\,\bullet\,]_j$ indicates the quantity measured on sample j. By the strong law of large numbers the right hand side is an unbiased estimator of the true expectation and its variance falls as $\mathcal{O}(1/n)$ as n becomes large.

Discretization of the gauge part of the action goes straight forwardly. We require the gauge action, to leading order in the same small a expansion, to have the same action as the continuum theory. A general form of the gauge action is

$$S_g = -\frac{\beta}{3}\Big[(1-8c_1)\sum_{x,\mu<\nu} P_{\mu\nu}(x) + c_1 \sum_{x,\mu\neq\nu} R_{\mu\nu}(x)\Big], \tag{22}$$

where P and R are the ordered plaquette and rectangular link products,

$$P_{\mu\nu}(x) = \operatorname{tr} U_\mu(x) U_\nu(x+\hat{\mu}) U_\mu^\dagger(x+\hat{\nu}) U_\nu^\dagger(x), \tag{23}$$

$$R_{\mu\nu}(x) = \operatorname{tr} U_\mu(x) U_\mu(x+\hat{\mu}) U_\nu(x+2\hat{\mu}) U_\mu^\dagger(x+\hat{\mu}+\hat{\nu}) U_\mu^\dagger(x+\hat{\nu}) U_\nu^\dagger(x). \tag{24}$$

When $c_1 = 0$, S_g only contains the plaquette part and is called the Wilson gauge action; S_g is called the Iwasaki gauge action (DBW2 gauge action) when $c_1 = -0.331$ ($c_1 = -1.4088$).

1.2.2 Discretization of the Fermion Fields

The discretization of the fermion fields is much more complicated. A straightforward finite difference version of the differential operator in (9) gives us 16 excitation modes when the lattice spacing $a \to 0$ limit is taken: 15 redundant modes are added as lattice artifacts. A theorem proved by Nielsen and Ninomiya [4] states that it is not possible to have a fermion formulation in a four dimensional space that is simultaneously hermitian, local, and translationally invariant without explicitly breaking chiral symmetry while free of these lattice artifacts.

The lattice fermion formulation that will be used throughout this document is (Möbius) domain wall fermions [5]. It adds a fifth dimension s with size L_s to each 4-dimensional site on the lattice. With good approximation the low energy fermion modes with plus chirality (left handed) are placed on the $s = 0$ slice and those with minus chirality (right handed) are put on the $s = L_s - 1$ slice. Chiral symmetry is approximately preserved and the breaking is of order $\mathcal{O}(e^{-L_s})$. An illustration can be found in figure 1. Note that there is no fifth component of the gauge fields.

Here we introduce the notation of the Möbius domain wall fermion that will be reviewed further

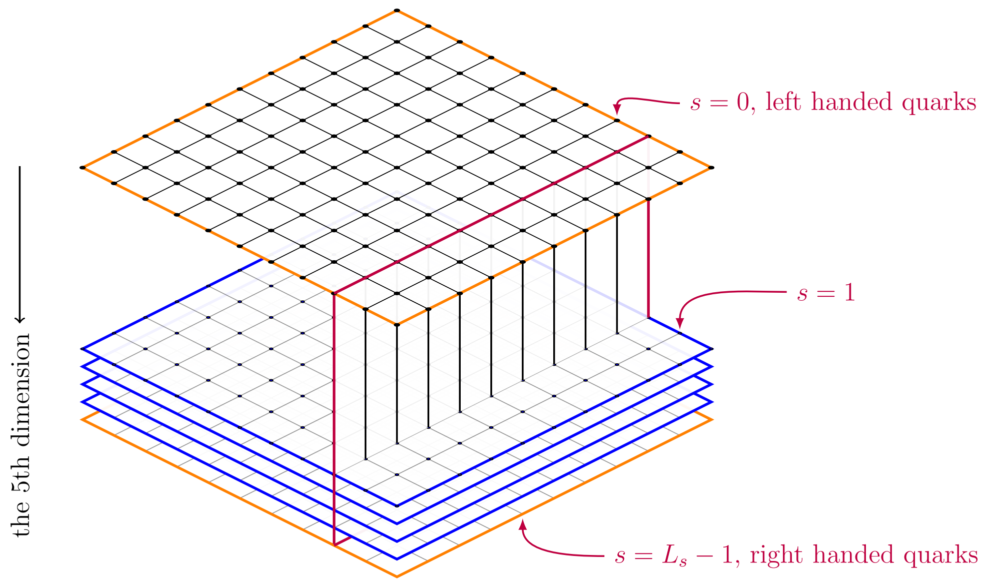
$s = 0$, left handed quarks
$s = 1$
$s = L_s - 1$, right handed quarks
the 5th dimension

Figure 1: An illustration of adding a fictitious fifth dimension to the original four dimensional lattice.

in section 5. The set of parameters of the formulation includes real and constant *Möbius parameters* b and c, which are related by

$$b - c = 1, \tag{25}$$

the fermion input mass m_f, the height of the domain wall m_5 and lastly the size of the fifth dimension L_s. Sometimes the term *Möbius scale* refers to the combination

$$\alpha = b + c. \tag{26}$$

Upon discretization the previous continuous Dirac operator turns into a matrix acting on the fermion vectors. First we introduce

$$\kappa_b^{-1} = 2[b(4 - m_5) + 1], \tag{27}$$

$$\kappa_c^{-1} = 2[c(4 - m_5) - 1]. \tag{28}$$

We label the four dimensional points as even or odd, depending on whether $x_1+x_2+x_3+x_4$ is even or odd. In an even(e)-odd(o) basis the Möbius domain wall fermion operator acts as

$$D\psi = \begin{pmatrix} M_{ee} & M_{eo} \\ M_{oe} & M_{oo} \end{pmatrix} \begin{pmatrix} \psi_e \\ \psi_o \end{pmatrix} = \begin{pmatrix} \phi_e \\ \phi_o \end{pmatrix}. \tag{29}$$

M_{oo} and M_{ee} are nontrivial in the fifth dimension, even though they are diagonal in the even-odd basis:

$$M_{oo} = M_{ee} \equiv M_5 = 1 + \frac{\kappa_b}{\kappa_c} D_5, \tag{30}$$

$$D_{5[s,t]} = P_+(\delta_{s-1,t} - m_f \delta_{s,0}\delta_{t,L_s-1}) + P_-(\delta_{s+1,t} - m_f \delta_{s,L_s-1}\delta_{t,0}). \tag{31}$$

The fifth dimension index s, t runs from 0 to $L_s - 1$. On the other hand,

$$M_{oe} = M_{eo} = -\kappa_b D_w (b + cD_5) \equiv -\kappa_b D_w M_\phi. \tag{32}$$

D_w is the usual Wilson fermion matrix,

$$D_{w[x,y]} = \sum_\mu (1+\gamma_\mu) U^\dagger_{x-\hat{\mu},\mu} \delta_{x-\hat{\mu},y} + (1-\gamma_\mu) U_{x,\mu} \delta_{x+\hat{\mu},y}. \tag{33}$$

Note that the convention used here is equivalent to that in [5] (Brower) when the b_s and c_s are real and constant, up to a numerical factor

$$\Big[D\Big]_{\text{here}} = 2\kappa_b \cdot \Big[D\Big]_{\text{Brower}} \tag{34}$$

The D_- operator, which is diagonal in the fifth dimensional, is defined as

$$D_- = -c(4 - m_5) + \frac{1}{2} D_w. \tag{35}$$

The Möbius domain wall fermion enters the lattice action as

$$S_f = \bar{\psi} D(m_f) \psi + \phi^\dagger D(1) \phi. \tag{36}$$

The second (Pauli-Villar) term, $\phi^\dagger D(1)\phi$, enters to cancel the divergence that arise when the $L_s \rightarrow$

∞ is taken in the first term, $\bar{\psi}D(m_f)\psi$ [6]. The *pseudo-fermion* field ϕ is a bosonic (commuting) field with color and spinor indices.

1.3 The Evolution

A complete lattice QCD simulation consists of two phases: the Monte Carlo phase in which samples of the gauge configurations are drawn according to the probability e^{-S}, often called the *evolution phase*, and the phase in which lattice correlation functions are measured and renormalization procedures are performed, often called the *measurement phase*. One of the most important feature of lattice QCD results, if not *the* most important one, is the statistical and systematic error that come along with the results. Lattice results are meaningless without reliable error analysis. Both phases contribute to both types of error of the final results. They are both important, if not equally important. When measurements are performed the statistical error scales as $\mathcal{O}(1/\sqrt{n})$ but the number of independent samples n one can possibly measure on during the measurement phase is limited by the number of independent samples that are generated during the evolution phase. Due to limited computational resources the number of samples is limited and samples can be correlated. These limitations play a central role in the error analysis.

An introduction of the evolution phase will be given in this subsection.

1.3.1 The Markov Chain and the Metropolis Steps

For reasons that will be clear later the samples we are generating are only the gauge configurations, i.e. the set of SU(3) gauge matrices on the lattice sites. All possible gauge configurations are identified as the random variable of a Markov chain, whose transition matrix, which we need to construct, should lead to a stationary distribution that is identical to our target probability $e^{-S[U]}$.

Next we construct the transition matrix. Suppose we start with a configuration $\{U_0\}$. Once a new configuration $\{U_1\}$ is proposed we accept it with the *detailed balance* probability

$$P = \begin{cases} 1, & S[U_1] \leq S[U_0] \\ \exp[S[U_0] - S[U_1]], & S[U_1] > S[U_0] \end{cases} \tag{37}$$

otherwise the $\{U_0\}$ is kept unchanged.

Note that this is the approach that will be used in this document but it is not the only possible way to achieve the desired stationary distribution.

1.3.2 The Hybrid Monte Carlo Algorithm

We still need a way to propose random but reasonably probable new configuration $\{U_1\}$'s such that $S[U_1]$ is equal or close to $S[U_0]$ to maintain a decent acceptance. The hybrid Monte Carlo (HMC) algorithm [7] achieves this by constructing a Hamiltonian system combining the gauge field $U_\mu(x)$ and their hypothetical conjugate momenta $\pi_\mu(x)$, together with a fictitious molecular dynamics (MD) time τ. Specifically $\pi_\mu(x)$ lives in the Lie-algebra of SU(3),

$$\pi_\mu(x) = \omega_\mu^a(x) t^a, \tag{38}$$

where t^a are the eight generators of SU(3) and $\omega_\mu^a(x)$ are eight real numbers for each lattice site. Here we assume the generators are normalized as

$$\operatorname{tr} t^a t^b = \frac{1}{2}\delta^{ab}. \tag{39}$$

The Hamiltonian of the system is

$$\mathcal{H}[\pi_\mu(x), U_\mu(x)] = \sum_{x,\mu} \frac{1}{2}\omega_\mu^a(x)\omega_\mu^a(x) + S[U] \tag{40}$$

With a set of Hamiltonian equations the system undergoes a fictitious *evolution* in τ, which is the reason why the evolution phase gets its name. The set of equations are

$$\frac{d}{d\tau} U_\mu(x) = i\pi_\mu(x) U_\mu(x), \tag{41}$$

$$\frac{d}{d\tau} \pi_\mu(x) = -\partial_{x,\mu}^a S[U] t^a, \tag{42}$$

where the Lie derivative $\partial^a_{x,\mu}$ is defined as

$$\partial^a_{x,\mu} S[U] = \frac{d}{ds} S[U^{s,a}]\bigg|_{s=0}, \ U^{s,a}_\nu(y) = \begin{cases} e^{ist^a} U_\mu(x) & \mu = \nu, x = y \\ U_\nu(y) & \text{else} \end{cases} \tag{43}$$

The conjugate momenta $\pi_\mu(x)$ introduce randomness into the system and drive the gauge field U to move in a certain direction with the Hamiltonian *trajectory*. The path integral is extended to include these momenta and the joint partition function is now

$$Z = \int [d\pi][dU] \exp\left[-\mathcal{H}[\pi, U]\right]. \tag{44}$$

At the beginning of every trajectory the momenta are refreshed without much technical difficulties due to the fact that this part of the joint probability $e^{-\mathcal{H}}$ is gaussian. The Hamiltonian system is then evolved with a certain trajectory length. If the molecular dynamics evolution is performed exactly without any finite step-size error and machine rounding error, the Hamiltonian is conserved exactly and the change is always accepted in the metropolis step at the end of the trajectory. With the finite step-size errors present in a practical trajectory an accept-reject step at the end of the trajectory is still needed for the joint system $\mathcal{H}(\pi, U)$.

The above procedure is repeated for every trajectory and a series of gauge configurations are generated.

1.3.3 Incorporating the fermion fields into the HMC

So far we have ignored the presence of the fermion fields in the action S and their treatment in the evolution phase. The fermion fields are constructed from a basis of anti-commuting grassmann variables, which makes them difficult to compute on conventional modern computers, at least before the existence of large-scale general purpose *quantum computers*. In this subsubsection we will discuss the current method to incorporate the fermion fields into the HMC.

It turns out that the fermion fields can be integrated out analytically

$$\int [d\bar{\psi}][d\psi] \exp[-\bar{\psi} D \psi] = \det D. \tag{45}$$

All the information about the fermion fields are embedded in the determinant of the Dirac matrix $D[U]$, which could be calculated in principle. In reality, however, modern lattice QCD simulations usually deal with a Dirac matrix of size $\sim 10^{10}$ by $\sim 10^{10}$. Calculating the determinant of an $N \times N$ matrix requires $\mathcal{O}(N^3)$ floating point operations, which in our case scales to $\mathcal{O}(10^{30})$. Even with an exascale machine (a machine with a computational capability of more than 10^{18} floating point operations per second (flops)) a single evaluation would take around $\sim 10^{12}$ seconds, or $\sim 10^4$ years.

This computation difficulty can be overcome by evaluating the determinant stochastically rather than analytically. By introducing a *pseudo* fermion field ϕ the determinant is stochastically estimated by

$$\int [d\bar{\psi}][d\psi] \exp[-\bar{\psi} D \psi] = \det D = \int [d\phi^\dagger][d\phi] \exp[-\phi^\dagger D^{-1} \phi]. \tag{46}$$

In addition although $\det D$ is guaranteed be be real, the matrix D itself has complex spectrum. For a specific ϕ, $\phi^\dagger D^{-1} \phi$ will likely become complex and jeopardize the probability interpretation of the exponential. A more sophisticated approach is to hermitianize the form

$$\det D = \det(D^\dagger D)^{1/4} (D^\dagger D)^{1/4} = \int [d\phi^\dagger][d\phi] \exp[-\phi^\dagger (D^\dagger D)^{-1/4} (D^\dagger D)^{-1/4} \phi] \tag{47}$$

$$= \int [d\phi^\dagger][d\phi] \exp[-\|(D^\dagger D)^{-1/4} \phi\|^2]. \tag{48}$$

The right most side of (46) can now be calculated on a computer but the price we pay is that we need to invert a fractional power p (in this case $p = 1/4$) of the (usually gigantic) matrix D. The traditional way to calculate this is to use the Chebyshev rational approximation method [8]. A rational polynomial is constructed to approximate the inverse fractional power

$$x^{-p} \simeq r_p(x) = \alpha_0 + \sum_i \alpha_i (x + \theta_i)^{-1}, \; 0 < \lambda_{\min} < x < \lambda_{\max}, \tag{49}$$

where $\lambda_{\min}$ and $\lambda_{\max}$ are the minimum and maximum of the eigenvalue of the matrix $D^\dagger D$. The inverse fractional power is then calculated with

$$(D^\dagger D)^{-p} = r_p(D^\dagger D) = \alpha_0 + \sum_i \alpha_i (D^\dagger D + \theta_i)^{-1}, \tag{50}$$

where a series of shifted inversions of the normal operator $D^\dagger D$ needs to be performed on the pseudo-fermion vector ϕ. Here the term *inversion* refers to solving the following *Dirac equations*, i.e. linear equations with the Dirac matrix D,

$$D^\dagger D x = y \ \rightarrow \ x = (D^\dagger D)^{-1} y. \tag{51}$$

With current computation power of the state-of-art supercomputers and fairly sophisticated solving techniques these inversions can be done within a reasonable amount of time. This will be the focus of section 5.

More recently the exact one flavor algorithm (EOFA) has been proposed [9] to calculate $\det D$ without the need to perform a series of shifted inversions for every evolution step.

For a 2+1 flavor lattice QCD simulation only the up, down and strange quarks are dynamically included in the evolution. Current Particle Data Group (PDG) values of the $\overline{\mathrm{MS}}$ mass of the up and down quark are 2.2 and 4.7 MeV. Both these values, as well as their difference, are much smaller than Λ_{QCD}. Thus the usual practice is to treat the up and down as quarks (the *light* quarks) with the same mass. This gives a technological advantage in that now the combination of the two fermion determinants is

$$\det D \det D = \det(D^\dagger D) = \int [d\phi^\dagger][d\phi] \exp[-\phi^\dagger (D^\dagger D)^{-1} \phi], \tag{52}$$

thus evaluation of the part of the action that arises from degenerate light quarks does not need the rational approximation and it involves only a single inversion $(D^\dagger D)^{-1}\phi$. The strange quark is sometimes called the *heavy* quark because of this reason[1].

The pseudo-fermion field ϕ needs to be refreshed before each HMC trajectory, which involves an additional Dirac matrix inversion, just as the conjugate momenta do. The Lie-derivatives of the single and degenerate flavor part of the fermion action are easily constructed with the introduction of the ϕ.

[1] Note that in most of the literatures the term *heavy quark* refers to the charm, bottom and top quark.

1.4 The Measurements

1.4.1 The Propagators

The measurement phase of lattice QCD simulation centers around the measurement of the propagotors of the form

$$\langle \psi(x_1,s)\bar{\psi}(x_2,t)\rangle = \frac{1}{Z}\int [d\bar{\psi}][d\psi][dU]\psi(x_1,s)\bar{\psi}(x_2,t)\exp[-\bar{\psi}D\psi+\cdots] \tag{53}$$

$$= \frac{1}{Z}\int [d\bar{\psi}][d\psi][dU]D^{-1}(x_1,s;x_2,t)\exp[-\bar{\psi}D\psi+\cdots] \tag{54}$$

$$= \langle D^{-1}(x_1,s;x_2,t)\rangle, \tag{55}$$

where the second equal can be shown by generating functional method. Note that most of the time we are only interested in the correlation function of the fermion field ψ and the pseudo-fermion field ϕ corresponds to the Pauli-Villar term in the action does not enter here.

For the measurement phase we typically want to calculate the correlation functions in four dimensional space. Based on the five dimensional current analysis [5], up to the $\mathcal{O}(e^{-L_s})$ chiral symmetry breaking, the effective four dimensional fermion field could be defined as

$$q(x) = P_+\psi(x,s=1) + P_-\psi(x,s=L_s). \tag{56}$$

$$\bar{q}(x) = \bar{\psi}(y,s=1)D_-(y,x)P_- + \bar{\psi}(y,s=L_s)D_-(y,x)P_+. \tag{57}$$

The four dimensional propagator can thus be calculated as

$$\begin{aligned}\langle q(x_1)\bar{q}(x_2)\rangle =&\langle P_+[D^{-1}D_-](x_1,1;x_2,L_s)P_+ + P_+[D^{-1}D_-](x_1,1;x_2,1)P_-\\ &+ P_-[D^{-1}D_-](x_1,L_s;x_2,L_s)P_+ + P_-[D^{-1}D_-](x_1,L_s;x_2,1)P_-\rangle \end{aligned}\tag{58}$$

$$\equiv\langle D^{-1}(x_1,x_2)\rangle \equiv \langle S_q(x_1\leftarrow x_2)\rangle. \tag{59}$$

$\langle D^{-1}(x_1,x_2)\rangle$, or $\langle S_q(x_1 \leftarrow x_2)\rangle$, as well as $\langle q(x_1)\bar{q}(x_2)\rangle$, are often referred as the (four dimensional) lattice quark propagator. Evaluation of the full inverted matrix is not realistic with current computation power but linear combinations of the form are possible to be computed,

$$\sum_{x_2} S_q(x_1 \leftarrow x_2)\eta(x_2) = \xi(x_1), \tag{60}$$

which is equivalent to solving for ξ in the linear equation

$$\sum_{x_2} D(x_1, x_2)\xi(x_2) = \eta(x_1). \tag{61}$$

The field is η is referred to as the *source* of the propagator. It should only be understood as a collection of complex numbers: all the particle nature of the original fermion fields ψ has been removed. More measurement techniques will be introduced in section 4.

1.4.2 The Correlation Functions

With propagators correlation functions of quark bound states can be constructed. Single pion correlation function, for example, can be expressed as (u and d are the up and down quark, t_1 and t_2 are time coordinates while $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ and $\boldsymbol{y}$ are space coordinates.)

$$\left\langle \sum_{\boldsymbol{x}_1,\boldsymbol{y}_1} \bar{d}(\boldsymbol{x}_1, t_1)\gamma_5 u(\boldsymbol{y}_1, t_1) \sum_{\boldsymbol{x}_2,\boldsymbol{y}_2} \bar{u}(\boldsymbol{x}_2, t_2)\gamma_5 d(\boldsymbol{y}_2, t_2) \right\rangle = \left\langle \gamma_5 S_u(t_1 \leftarrow t_2)\gamma_5 S_d(t_2 \leftarrow t_1) \right\rangle, \tag{62}$$

where the *wall source wall sink* propagator is defined on two time slices as

$$S_q(t_1 \leftarrow t_2) = \sum_{\boldsymbol{x},\boldsymbol{y}} S_q(\boldsymbol{x}, t_1 \leftarrow \boldsymbol{y}, t_2). \tag{63}$$

Another common propagator is the *wall source point sink* propagator

$$S_q(\boldsymbol{x}, t_1 \leftarrow t_2) = \sum_{\boldsymbol{y}} S_q(\boldsymbol{x}, t_1 \leftarrow \boldsymbol{y}, t_2). \tag{64}$$

For Möbius domain wall fermion the propagator has γ_5-hermiticity so we can easily revert the direction of the arrow:

$$S_q(\boldsymbol{x}, t_1 \leftarrow \boldsymbol{y}, t_2) = \gamma_5 S_q^\dagger(\boldsymbol{y}, t_2 \leftarrow \boldsymbol{x}, t_1)\gamma_5, \tag{65}$$

$$S_q(t_1 \leftarrow \boldsymbol{x}, t_2) = \gamma_5 S_q^\dagger(\boldsymbol{x}, t_2 \leftarrow t_1)\gamma_5, \tag{66}$$

$$S_q(t_1 \leftarrow t_2) = \gamma_5 S_q^\dagger(t_2 \leftarrow t_1)\gamma_5. \tag{67}$$

1.4.3 Fitting Strategy

According to Proposition 2 in [10] a generic correlation function in lattice gauge theory can be related to a statistical theory with a corresponding Euclidean Hamiltonian:

$$\langle \phi_1(t_1)\cdots\phi_k(t_k)\rangle_g = \frac{1}{Z}\mathrm{Tr}\left[e^{-\hat{H}(T+t_1)}N[\hat{\phi}_1]e^{-\hat{H}(t_2-t_1)}N[\hat{\phi}_2]\cdots N[\hat{\phi}_k]e^{-\hat{H}(-t_k)}\right],\ t_1 < t_2\cdots < t_k, \tag{68}$$

where $\langle\bullet\rangle_g$ means correlation function with the gauge configuration fixed to a certain gauge g, $\hat{H}$ is the Euclidean Hamiltonian associated with the gauge theory action, $N[\bullet]$ stands for normal ordering defined also in [10].

For a meson corrector $k = 2$, e.g. in (62). According to (68)

$$\langle \phi_1(t_1)\phi_2(t_2)\rangle = \frac{1}{Z}\mathrm{Tr}\left[e^{-\hat{H}(T-t)}\hat{\phi}_1 e^{-\hat{H}t}\hat{\phi}_2\right]. \tag{69}$$

Expand the trace in the basis consists of the eigenstates of $\hat{H}$:

$$\hat{H}|i\rangle = E_i|i\rangle,\ i = 0,1,2,\cdots,\ E_0 = 0, \tag{70}$$

$$\langle \phi_1(t_1)\phi_2(t_2)\rangle = \frac{1}{Z}\sum_{i,j} e^{-E_j(T-t)}e^{-E_i t}\langle j|\hat{\phi}_1|i\rangle\langle i|\hat{\phi}_2|j\rangle \tag{71}$$

$$= \frac{1}{Z}\left[e^{-E_1 t}\langle 0|\hat{\phi}_1|1\rangle\langle 1|\hat{\phi}_2|0\rangle + e^{-E_1(T-t)}\langle 1|\hat{\phi}_1|0\rangle\langle 0|\hat{\phi}_2|1\rangle + \mathcal{O}\left(e^{-E_1 t} + e^{-E_1(T-t)}\right)\right] \tag{72}$$

$$\overset{*}{=} A\left[e^{-E_1 t} \pm e^{-E_1(T-t)}\right], \tag{73}$$

where $t = t_2 - t_1$ and the amplitude A has absorbed the various constants. The condition $*$ indicates that with a given range of t all other higher order terms in (72) are suppressed. The sign choice in the fit form depends on the relative sign difference between $\langle 0|\hat{\phi}_1|1\rangle\langle 1|\hat{\phi}_2|0\rangle$ and $\langle 1|\hat{\phi}_1|0\rangle\langle 0|\hat{\phi}_2|1\rangle$. Fitting to forms similar to (73) yields physical energy levels of the quark bound states as well as the amplitude information contained in A.

1.4.4 Fitting Forms for Meson Correlators

The extraction of the matrix elements frequently involves meson (and hadron) states such as the pion and kaon states. While there is no known easy way to construct these particle states exactly[2], *interpolating operators* are used to create these particles states by paying the price of an overlapping factor in the fitting form. Specifically we choose the ϕ_1 and ϕ_2 in (71) to be the correlator that has the same quantum numbers as the particle to be extracted. For example in order to create a π^+ state and calculate the pion correlator we use $\bar{u}\gamma_5 d$ as the interpolating operator

$$C_\pi(t_1, t_2) = \langle [\bar{u}\gamma_5 d]^\dagger(t_1)[\bar{u}\gamma_5 d](t_2)\rangle. \tag{74}$$

The correlator C_π is then fitted as

$$C_\pi(t_1, t_2) \to |Z_\pi|^2 \Big[e^{-m_\pi t} + e^{-m_\pi(T-t)}\Big],\ Z_\pi \propto \langle \pi|\bar{u}\gamma_5 d|0\rangle \tag{75}$$

since the lowest energy state that has the same quantum number and net quark content is indeed a pion state.

1.5 Lattices Setup

In table 1 the lattices involved in this work are listed. See the caption for a description of the all the labels.

[2] In fact measurement of the parton distribution functions (PDF) that describes the inner structure of the particles is a hot topic in lattice QCD.

Ensemble	β	$L^3 \times T \times L_s$	α	am_l	am_h	am_π
24**I**	2.13	$24^3 \times 64 \times 16$	1.0	0.005	0.04	0.19032(67)
		$24^3 \times 64 \times 16$	1.0	0.01	0.04	0.24221(78)
32**I**	2.25	$32^3 \times 64 \times 16$	1.0	0.004	0.03	0.16924(62)
		$32^3 \times 64 \times 16$	1.0	0.006	0.03	0.15101(50)
		$32^3 \times 64 \times 16$	1.0	0.008	0.03	0.17246(63)
32**I**-fine	2.37	$32^3 \times 64 \times 12$	1.0	0.0047	0.0186	0.11778(23)
48**I**	2.13	$48^3 \times 96 \times 24$	2.0	0.00078	0.0362	0.08050(14)
64**I**	2.25	$64^3 \times 128 \times 12$	1.0	0.000678	0.02661	0.05905(11)
96**I**	2.31	$96^3 \times 192 \times 12$	2.0	0.00054	0.02132	0.0483*
32**ID**	1.75	$32^3 \times 64 \times 32$	1.0	0.001	0.046	0.1250(2)
		$32^3 \times 64 \times 32$	1.0	0.0042	0.046	0.1810(2)
		$32^3 \times 64 \times 12$	32/12	0.0001	0.045	0.10468(32)
32**ID**-M1	1.633	$32^3 \times 64 \times 24$	4.0	0.00022	0.0596	0.11806(25)
24**ID**	1.633	$24^3 \times 64 \times 24$	4.0	0.00107	0.085	0.13974(16)
32**ID**	1.633	$32^3 \times 64 \times 24$	4.0	0.00107	0.085	0.13945(15)
32**ID**-kaon		$32^3 \times 64 \times 24$	4.0	0.00107	0.0305	0.13412(27)
32**ID**-M2	1.943	$32^3 \times 64 \times 12$	4.0	0.00478	0.03297	0.19483(64)

Table 1: Summary of ensembles included in this work and their input parameters. Here β is the gauge coupling, $L^3 \times T \times L_s$ is the lattice volume decomposed into the length of the spatial (L), temporal (T), and fifth (L_s) dimensions, and am_l and am_h are the bare, input light and heavy quark masses. $\alpha = b + c$ is the Möbius parameter. The value of am_π quoted is the unitary pion mass. Label **I** in ensemble names indicate Iwasaki gauge action and **ID** indicates Iwasaki gauge action plus the dislocation suppressing determinant ratio (DSDR) term. The pion mass for 96**I**, which is measured from one single wall source on one single configuration, is very preliminary.

1.6 Global Chiral Fit

The particle mass and matrix elements measured on the lattices contain finite lattice spacing errors and finite volume errors, in addition to other lattice artifacts, and need to be extrapolated to the physical quark mass point in order to give physical predicitons. The strategy to remove these errors is to fit the measured low energy observables to a combined theory that includes quark mass dependence through chiral perturbation theory (ChPT) and a linear analytic anzatz for the a^2

dependence,

$$X(m, a^2) \simeq X_0(1 + f(m) + c_X^{\mathbf{A}} a^2), \tag{76}$$

where $f(m)$ is the ChPT form, which depends on the input quark masses, and $c_X^{\mathbf{A}} a^2$ is the linear anzatz. The a^2 coefficient is different between the Iwasaki ($c_X^{\mathbf{I}}$) and **ID** ($c_X^{\mathbf{ID}}$) lattices. If SU(2) ChPT is used the theory also includes a linear analytic anzatz in the heavy quark mass m_h. The fitting strategy is detailed extensively in [11].

1.7 Wilson Flow

The Wilson flow evolves (flows) the lattice gauge fields according to a certain differential equation shown in (1.4) of [12]. In this work it mainly serves two purposes:

1. On the flowed gauge field, compared to the unflowed one, localized fluctuations that reflect detailed ultraviolet lattice artifacts are smeared away. This allows topological charge to be measured on lattices with much less noise.

2. The Wilson flow time t, together with the Wilson flow energy, which is an gauge invariant quantities defined also in [12], can be used to set the lattice scale. Specifically two quantities, $\sqrt{t_0}$ and w_0, have been defined in [12] and [13], respectively. By measuring them in lattice unit and matching them to the physical value, the lattice spacing of the underlying lattice is determined.

2 Evolution of Topological Charge on Lattices: the Diffusion Model

2.1 Abstract

We apply the diffusion model, which successfully describes how topological charge density moves across Euclidean space-time lattices and tunnels across molecular dynamics time during hybrid Monte Carlo(HMC) evolution of a series of pure DBW2 gauge ensembles, is applied to ensembles with dynamical fermions. A new Fourier transform approach is proposed and used in the fitting. Apart from some possible lattice artifacts the diffusion model fits well to the dynamical ensembles. The diffusion constant D, which is shown to scale as a^2, serves as a key performance parameter to tell the success of new algorithms beyond the traditional HMC.

2.2 Introduction

The only currently viable algorithm for the numerical evaluation of the Feynman path integral for Quantum Chromodynamics (QCD), including fermions, is the hybrid Monte Carlo (HMC) algorithm. As the lattice spacing a decreases, the production of independent lattice configurations suffers from critical slowing down; the integrated autocorrelation time τ_{int} for an observable, such as the topological charge or energy density of the gauge configurations, increases dramatically. For quenched QCD, the scaling behavior of the integrated autocorrelation time for currently accessible values of a can be fit to the form

$$\tau_{\text{int}} \propto a^{-z}, \tag{77}$$

with z reported to be as large as 5 [14]. The presence of long correlations between the configurations jeopardizes the quality of the Monte Carlo samples and makes it harder to estimate the statistical error on the physical quantities measured on them.

The topological charge, Q_{top}, or topological charge density, $\rho(\boldsymbol{x}, t)$, are generally found to be observables with the largest integrated autocorrelation times. While only parity even combinations of these variables, such as Q_{top}^2 are relevant to physical observables, their rate of evolution during HMC simulations provides a stringent test of the quality of the evolution algorithm in avoiding critical slowing down. Given that continuum fields of different topologies are not connected by continuous deformations, it is not surprising that evolution algorithms are slow to change the topology of lattice fields. Improved algorithms should evolve these variables more rapidly. To provide a more quantitative method to describe the evolution of topology, a diffusion model is proposed in [15] to describe the correlations in Euclidean space (on the lattice) and between configurations (in molecular dynamics (MD) time) of the topological charge on a Euclidean time-slice t at MD time τ, denoted by $Q(t, \tau)$.

The diffusion model in [15] was found to well-describe the evolution of quenched QCD with the HMC algorithm. With the wide use of full QCD gauge configurations, which include physical dynamical light and strange quarks, it is natural to ask whether this model equally well describes the evolution of topology on lattice configurations with fermions. In this report, we detail results from studies of topological evolution in full QCD and observe that, as an empirical model, the diffusion model works well on dynamical lattices with fine enough lattice spacing. On coarse lattices with dynamical fermions, we find that the long distance behavior of the topological charge correlations in Euclidean space deviates from the diffusion model prediction.

By fitting data to the diffusion model, we gain a quantitative measure of the rate at which topological charge is moving around the lattice, given by the diffusion constant D, and also the rate at which it is created and destroyed, given by τ_{tunn}. The diffusion constant D is observed to be proportional to a^2 for the dynamical lattices used in this paper; a similar result was found in the quenched case. We see that D serves as a key performance parameter, in addition to the usual integrated autocorrelation time, to quantify the success of new algorithms aimed at resolving the critical slowing down of lattice gauge theory simulations.

2.3 The Diffusion Model

In a continuum non-abelian gauge theory topological charge is an integer valued index of the gauge field that stays constant during continuous deformation of the field:

$$Q = \frac{g^2}{32\pi^2} \int d^4x F^a_{\mu\nu} \tilde{F}^{a\mu\nu}, \tag{78}$$

where F is the usual field strength tensor, $\tilde{F}$ being its dual. In lattice QCD a finite cut off of lattice spacing a is introduced. In this work we use the “5Li” [16] method to discretize the above continuous definition of the topological charge. Specifically we Wilson flow [12] the gauge field configuration to a reference flow time defined in equation (3.3) of [12] and use the following

$$Q = \sum_{\boldsymbol{x},t} \frac{1}{32\pi^2} \epsilon_{\mu\nu\rho\lambda} \mathrm{Tr}[\hat{F}_{\mu\nu}(\boldsymbol{x},t)\hat{F}_{\rho\lambda}(\boldsymbol{x},t)] = \sum_{\boldsymbol{x},t} \rho(\boldsymbol{x},t), \tag{79}$$

to find the topological charge density $\rho(\boldsymbol{x},t)$ on each lattice site. The Wilson flow smoothes the gauge fields and reduces localized fluctuations which reflect detailed ultraviolet lattice physics. The total topological charge is the sum of the density over the whole lattice. The detailed form of $\hat{F}_{\mu\nu}$ can be found in [16].

A diffusion model is proposed and the necessary quantities are constructed in [15]. Here we give a summary of the procedures. The topological charge observable $Q(t,\tau)$ is measured as

$$Q(t,\tau) = \sum_{\boldsymbol{x}} \rho(\boldsymbol{x},t;\tau), \tag{80}$$

where τ is the molecular dynamics(MD) time. The topological charge density is summed on each time slice t and each MD time τ. Then correlation functions $C(t,\tau)$ are constructed[3]:

$$C(t,\tau) = \langle Q(t_0+t,\tau_0+\tau)Q(t_0,\tau_0)\rangle_{t_0,\tau_0}. \tag{81}$$

Here $\langle\bullet\rangle_{t_0,\tau_0}$ means averaging over all possible t_0's and τ_0's. We will focus only on lattices with periodic boundary condition, thus this definition is different from that defined for lattice with open boundary condition in [15].

The diffusion model states the correlation function evolves according to the famous diffusion equation:

$$\frac{\partial}{\partial\tau}C(t,\tau) = D\frac{\partial^2}{\partial t^2}C(t,\tau) - \frac{1}{\tau_{\mathrm{tunn}}}C(t,\tau). \tag{82}$$

There are two parameters in the diffusion model: D and τ_{tunn}. D is the *diffusion constant* and intuitively measures the diffusion or the Euclidean space-time movement of topological charge over the lattice. τ_{tunn} is the *tunneling time scale* and measures the rate at which topological charge is

[3] In technical statistics language this is the natural *estimator* of the true correlation function $\hat{C}(t,\tau) = \langle Q(t_0 + t, \tau_0+\tau)Q(t_0,\tau_0)\rangle$.

created or annihilated.

2.4 Simulation Method and Parameters

2.4.1 Ensembles

In this section five ensembles with dynamical fermions are analyzed: three of them are generated with fermions in the Möbius domain wall formulation and the other two are generated with highly-improved staggered-quarks. In addition, in [15] five pure gauge ensembles generated with DBW2 gauge action were analyzed.

2.4.2 Ensembles with Möbius domain wall fermion

We analyze three of the RBC/UKQCD Collaborations' ensembles with Iwasaki gauge action, Möbius domain wall fermions (MDWF) for degenerate light (up and down) and strange quarks, and the dislocation suppressed determinant ratio (DSDR). Details of these ensembles are summarized in table 2. Discussion of the Möbius parameters $\alpha = b + c$ and the DSDR term are presented in [17]. The Hasenbusch mass splitting setups for the light quark determinant are also listed as $(m_1, m_2, \cdots, m_n)$. Instead of calculating the ratio $\det[D(m_l)/D(1)]$ directly, we rewrite that into the product of several ratios to reduce the scale of the fermion forces:

$$\det\left[\frac{D(m_l)}{D(1)}\right] = \det\left[\frac{D(m_l)}{D(m_1)}\right] \cdot \det\left[\frac{D(m_1)}{D(m_2)}\right] \cdots \det\left[\frac{D(m_n)}{D(1)}\right]. \tag{83}$$

Here D is the Dirac operator. For the strange quark determinant we use the rational approximation to calculate the fractional power.

For all three ensembles we use a nested Sexton-Weingarten integration scheme [18] for the molecular dynamics in HMC. A complete description can be found in Appendix A in [19] and we give a summary here. Specifically the full lattice Hamiltonian is

$$H = T + \sum S_R + \sum S_Q + S_{DSDR} + S_G, \tag{84}$$

where $\sum S_R$ is the rational quotient action[4] for strange quark; $\sum S_Q$ means the quotient action for

[4] Here we use the rational approximation to evaluate the fractional power of the fermion matrix, which is needed

	MDWF1	MDWF2	MDWF3
size $L^3 \times T \times L_s$	$12^3 \times 32 \times 12$	$24^3 \times 64 \times 24$	$24^3 \times 64 \times 12$
β	1.633	1.633	1.943
am_l	0.008521	0.00107	0.000787
$(m_1, m_2, \cdots, m_n)$	(0.0291,0.095,0.3,0.548)	(0.0291,0.095,0.3,0.548)	(0.00789,0.00291,0.095,0.3)
am_h	0.065073	0.085	0.019896
$\alpha = b + c$	4.0	4.0	4.0
$\Delta\tau$	1/12	1/8	1/6
a^{-1} [GeV]	1.01	1.01	1.98
m_π [MeV]	308	140	297

Table 2: Lattice simulation parameters for the MDWF ensembles. $(m_1, m_2, \cdots, m_n)$ shows the Hasenbusch mass splitting used for the light quark part of the action. $\alpha = b + c$ shows the Möbius scale. m_π is the unitary pion mass. $\Delta\tau$ refers to the step size for the top level integrator.

the light quark; S_G is the gauge action and S_{DSDR} is the DSDR action. T is the kinetic part of the Hamiltonian. Three levels of QPQPQ force gradient integrators(FGI) defined in equation (A6) in [19] are used. Denote the FGI integrator with generic *kinetic* part $\hat{T}$ and *potential* part $\hat{S}$ and trajectory length τ as

$$\begin{aligned} \hat{U}_{\mathrm{FGI}}\left(\tau; \hat{T}, \hat{S}\right) = &\exp\left(\alpha\tau\hat{T}\right) \exp\left(\frac{1}{2}\tau\hat{S} - \beta\tau^3\{\hat{S}, \{\hat{S}, \hat{T}\}\}\right) \\ &\cdot \exp\left(\gamma\tau\hat{T}\right) \exp\left(\frac{1}{2}\tau\hat{S} - \beta\tau^3\{\hat{S}, \{\hat{S}, \hat{T}\}\}\right) \exp\left(\alpha\tau\hat{T}\right), \end{aligned} \tag{85}$$

where $\alpha = (3 - \sqrt{3})/6$, $\beta = (2 - \sqrt{3})/48$ and $\gamma = \sqrt{3}/3$. The original Hamiltonian is decomposed as

$$H = T_0' + \sum S_R, \tag{86}$$

$$T_0' = T_1' + \sum S_Q, \tag{87}$$

$$T_1' = T + (S_G + S_{DSDR}). \tag{88}$$

for the strange quark. See equation (A16) in of [19].

The integrator we finally use is

$$\begin{aligned}\exp\Big(\Delta\tau H\Big) &= \hat{U}_{\mathrm{FGI}}\Big(\Delta\tau; T_0', \sum S_R\Big) && (89)\\ &= \hat{U}_{\mathrm{FGI}}\Big(\Delta\tau; \Big[\hat{U}_{\mathrm{FGI}}\Big(1/n_1; T_1', \sum S_Q\Big)\Big]^{n_1}, \sum S_R\Big) && (90)\\ &= \hat{U}_{\mathrm{FGI}}\Big(\Delta\tau; \Big[\hat{U}_{\mathrm{FGI}}\Big(1/n_1; \Big[\hat{U}_{\mathrm{FGI}}\Big(1/n_2; T, S_G + S_{DSDR}\Big)\Big]^{n_2}, \sum S_Q\Big)\Big]^{n_1}, \sum S_R\Big). && (91)\end{aligned}$$

In the first level we treat the composite operator T_0' as the kinetic part in the FGI integrator, where whenever we need to integrate $\exp(\tau T_0')$ for some trajectory length τ we deploy a second level of FGI integrator with T_1' as the kinetic part. For this second level whenever we encounter T_1' a third level of FGI integrator with T as the kinetic part is used. The n_1 and n_2 could be any positive integers and for this document we always use $n_1 = n_2 = 1$. The $\Delta\tau$ for the top level integrator can be found in table 2.

2.4.3 Ensembles with Highly-improved Staggered-quark Fermion

We also analyze two of the MILC Collaboration's ensembles of QCD gauge-field configurations with four flavors of highly-improved staggered-quark(HISQ) fermions. Labelled in this work with `MILC1` and `MILC2`, these two ensembles are listed in line 4(Key $m_s/5$ with $a \sim 0.12$ fm) and line 16(Key $m_s/5$ with $a \sim 0.09$ fm) in table I of [20], respectively. A summary of these two ensembles are presented in table 3 but readers should refer to [20] for further details.

	`MILC1`	`MILC2`
size $L^3 \times T$	$24^3 \times 64$	$32^3 \times 96$
β	6.00	6.30
am_l	0.0102	0.0509
am_h	0.0074	0.037
am_c	0.635	0.440
a^{-1} [GeV]	1.64	2.19
m_π [MeV]	299	301

Table 3: Lattice simulation parameters for the MILC ensembles.

2.5 Approaches to Fit to the Diffusion Model

2.5.1 Partial Differential Equation (PDE) Approach

In [15] the simulation data is fitted to the partial differential equation of diffusion model directly. Consider the discretized version of equation (82):

$$C(t,\tau+1) - C(t,\tau) = D[C(t+1,\tau) + C(t-1,\tau) - 2C(t,\tau)] - \frac{1}{\tau_{\text{tunn}}} C(t,\tau). \tag{92}$$

With the correlation function $C(t,\tau)$ constructed, starting from the *initial condition* $C(t,\tau=0)$ this equation is integrated numerically to obtain the model prediction for the correlation functions for $\tau > 0$, symbolically denoted as $C_{\text{model}}(t,\tau)$. The two free parameters, diffusion constant D and tunneling scale τ_{tunn}, are then varied to minimize the χ^2 defined as

$$\chi^2 = \sum_t \sum_{\tau>0}^{\tau_{\text{cut}}} \left[\frac{C_{\text{model}}(t,\tau) - C(t,\tau)}{\text{Err}[C(t,\tau)]} \right]^2, \tag{93}$$

where a cufoff τ_{cut} is introduced to truncate the long summation over τ since the correlation function is expected to vanish for large enough τ.

The error of the constructed correlation function $C(t,\tau)$(donated as $\text{Err}[C(t,\tau)]$) is estimated using the following formula [21][5]

$$\text{Err}[C(t,\tau)]^2 = \left\langle C(t_0,\tau_0)^2 + C(t_0+t,\tau_0+\tau)C(t_0-t,\tau_0-\tau) \right\rangle_{\tau_0,t_0}. \tag{94}$$

To prevent data from being completely polluted with noise this error function is only calculated for $\tau \leq \tau_{\text{err}}$, where τ_{err} is some other cutoff which is chosen to be less than τ_{cut}. For $\tau_{\text{err}} < \tau < \tau_{\text{cut}}$ the limit value is used, i.e.

$$\text{Err}[C(t,\tau_{\text{err}} < \tau < \tau_{\text{cut}})] = \text{Err}[C(t,\tau_{\text{err}})]. \tag{95}$$

[5] Again technically speaking this is actually *an* estimator for the square root of the variance of the correlation function, which is itself an estimator.

2.5.2 Fourier Transform Approach

Here we propose a new approach. Equation (82) is organized into the following form

$$\frac{\partial}{\partial \tau} C(t,\tau) = D[C(t+1,\tau) + C(t-1,\tau) - 2C(t,\tau)] - \frac{1}{\tau_{\text{tunn}}} C(t,\tau), \tag{96}$$

i.e. the τ derivative is kept continuous and the double derivative with respect to t is turned into its finite difference form. Performing discrete Fourier transform on $C(t,\tau)$ with respect to t gives the following

$$\frac{\partial}{\partial \tau} c(k,\tau) = \left[D(2\cos k - 2) - \frac{1}{\tau_{\text{tunn}}} \right] c(k,\tau), \tag{97}$$

where the Fourier convention is

$$c(k,\tau) = \sum_{t=0}^{T-1} C(t,\tau) e^{ikt},\ k = \frac{2\pi}{T}(-T/2), \frac{2\pi}{T}(-T/2+1), \cdots, \frac{2\pi}{T}(+T/2-1). \tag{98}$$

Equation (97) is trivially solved using an exponential ansätz

$$c(k,\tau) = c_0 \exp\left[-\left(D(2 - 2\cos k) + \frac{1}{\tau_{\text{tunn}}} \right) \tau \right]. \tag{99}$$

Numerically we form the following ratio

$$B(k) = \log\left[\frac{c(k,\tau)}{c(k,\tau+1)} \right] = D(2 - 2\cos k) + \frac{1}{\tau_{\text{tunn}}}. \tag{100}$$

Fitting the ratio $B(k)$ to the above equation gives us D and τ_{tunn}.

On the five ensembles with dynamical fermions we adopt this Fourier transform approach instead of the PDE one. The necessity of this choice will be clear later in the results section.

2.6 Results

2.6.1 Ensembles with Dynamical Fermions

The fitting results on the five ensembles with dynamical fermions could be found in table 4 and the fitting shapes are presented in figure 2.

The diffusion model, and consequently equation (100), indicates a linear relationship between

$B(k)$(defined in equation (100)) and $(2-2\cos k)$ with D as the slope and $1/\tau_{\text{tunn}}$ as the intercept. This is not, however, entirely the case as shown in figure 2. We find a range of k values where we can reliably conclude a linear relationship and extract D and $1/\tau_{\text{tunn}}$. Correlations deviate from linear relationship when k is close to zero on ensembles with coarse lattice spacing(MDWF1, MDWF2 and MILC1) and spikes are present for these k values. On ensembles with fine lattice spacing(MDWF3 and MILC2) the correlations fit to the diffusion model over full range of k values.

2.6.2 Pure DBW2 Gauge Ensembles

As a comparison we also apply the new Fourier transform approach to the five pure gauge ensembles previously analyzed in [15]. The fitting shapes are presented in figure 3.

The deviations from linearity that appear on ensembles with dynamical fermions at small k values are absent. Correlations at large k values for the ensembles with fine lattice spacing deviate from a linear behavior. This appears to be the consequence from how the correlation data is measured on the pure DBW2 gauge ensembles. While we measure correlation functions for every 1 MD time unit on the dynamical ensembles, the same data are measured every 28 and 40 MD time units on the DBW2-14 and DBW2-16 ensemble, respectively. For large k values the magnitude of the correlation signal over such a long MD time period diminishes and is likely polluted by noise. As a result for the Fourier transform approach we only extract D and $1/\tau_{\text{tunn}}$ from the first few k values(indicated in the figure 3), while effectively the PDE approach always fits over the full range of k values. The diffusion constant D extracted from the two approaches, listed in table 5, are consistent despite different fitting range.

2.6.3 Scaling of Diffusion Constant

We plot the diffusion constant D for all ten pure DBW2 gauge and dynamical ensembles versus their respective lattice spacing a in figure 4. Apart from a small a^4 term, D scales as a^2.

2.7 Conclusions

We observe that on dynamical fermion lattices with periodic boundary conditions the evolution of the topological charge is well described by the diffusion model, if the lattice spacing is finer than $(2 \text{ GeV})^{-1}$, as our analysis shows on MDWF3 and MILC2 lattices. The same conclusion holds on

	FT approach	
	D/a^2	τ_{tunn}
`MDWF1`	0.094(11)	5.00(31)
`MDWF2`	0.103(13)	4.63(39)
`MDWF3`	0.092(11)	31.5(71)
`MILC1`	0.093(11)	19.0(32)
`MILC2`	0.0875(88)	28.3(28)

Table 4: Fitting results on the ensembles with dynamical fermions using the new Fourier transform approach.

	FT approach		PDE approach	
	D/a^2	τ_{tunn}	D/a^2	τ_{tunn}
`DBW2-08`	0.0777(72)	18(1)	0.090(12)	20(1)
`DBW2-10`	0.0918(53)	45(4)	0.1018(73)	56(3)
`DBW2-12`	0.0879(56)	104(15)	0.1085(97)	185(12)
`DBW2-14`	0.0976(64)	271(92)	0.1080(56)	561(59)
`DBW2-16`	0.0916(66)	300(106)	0.1155(29)	2350(389)

Table 5: Fitting results on the pure DBW2 gauge ensembles in [15] using the new Fourier transform approach and the PDE approach.

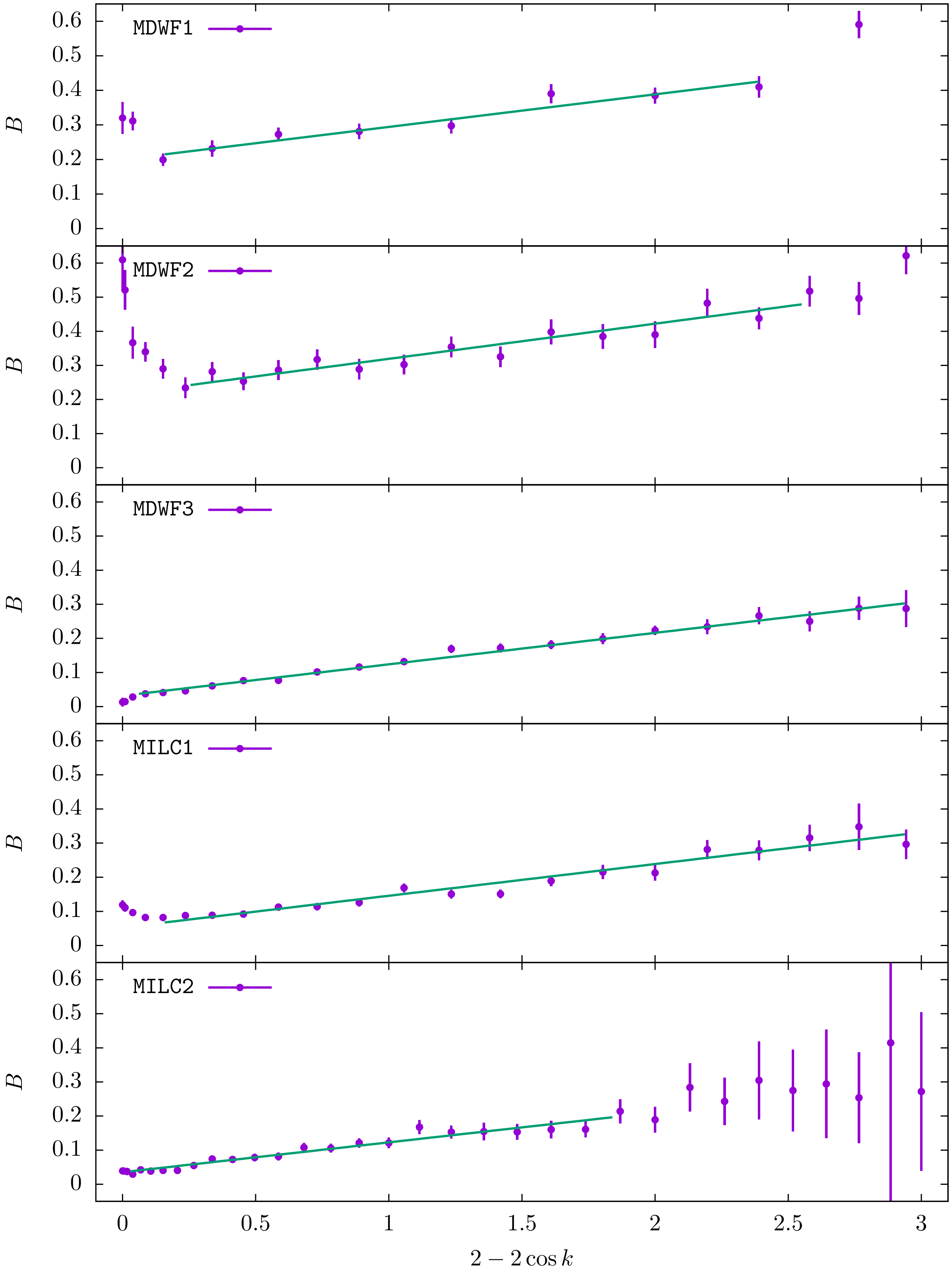

Figure 2: Results of fitting correlation functions to the diffusion model on the ensembles with dynamical fermions, using the new Fourier transform approach.

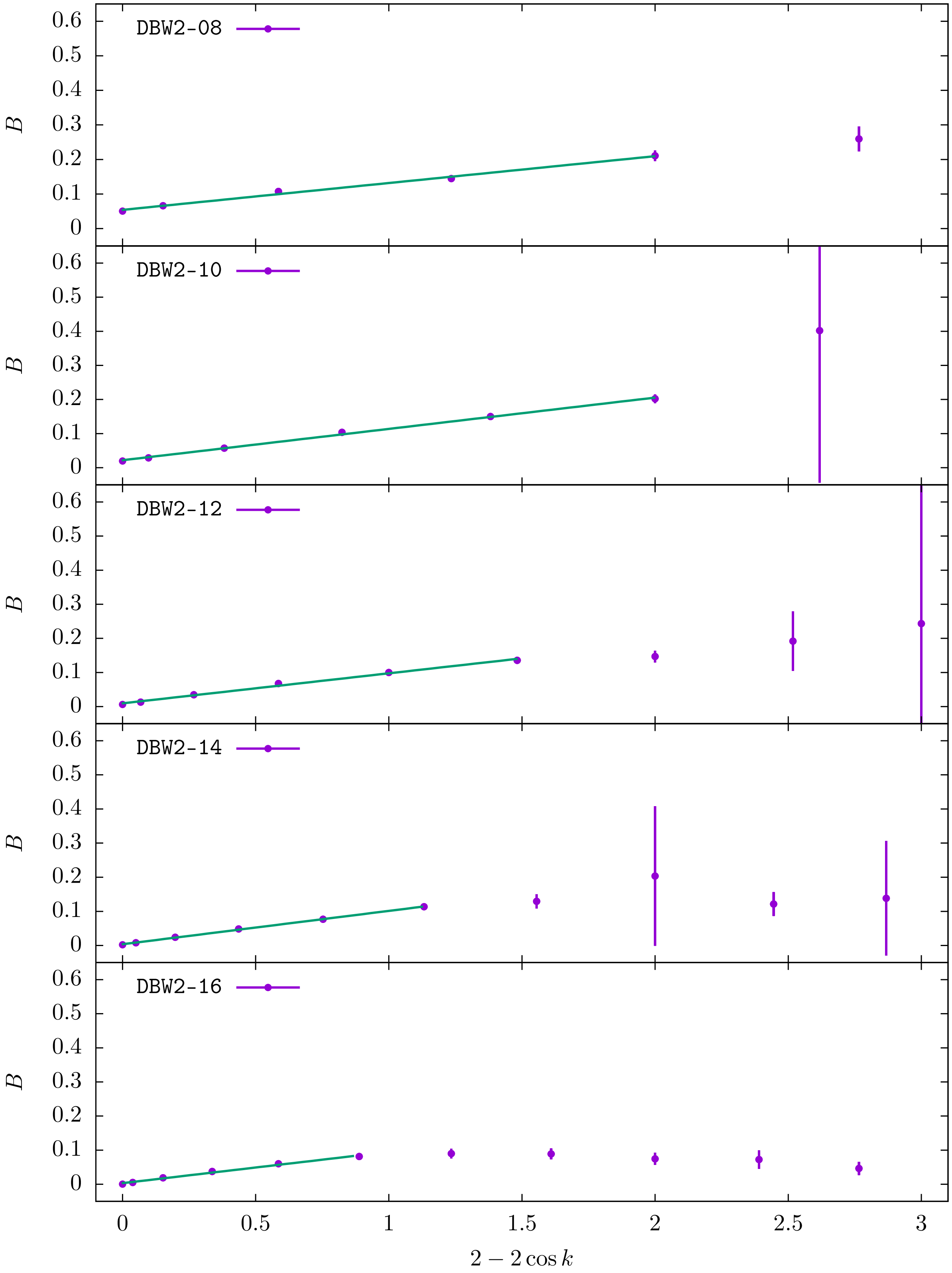

Figure 3: Results of fitting correlation functions to the diffusion model on the pure DBW2 gauge ensembles, using the new Fourier transform approach.

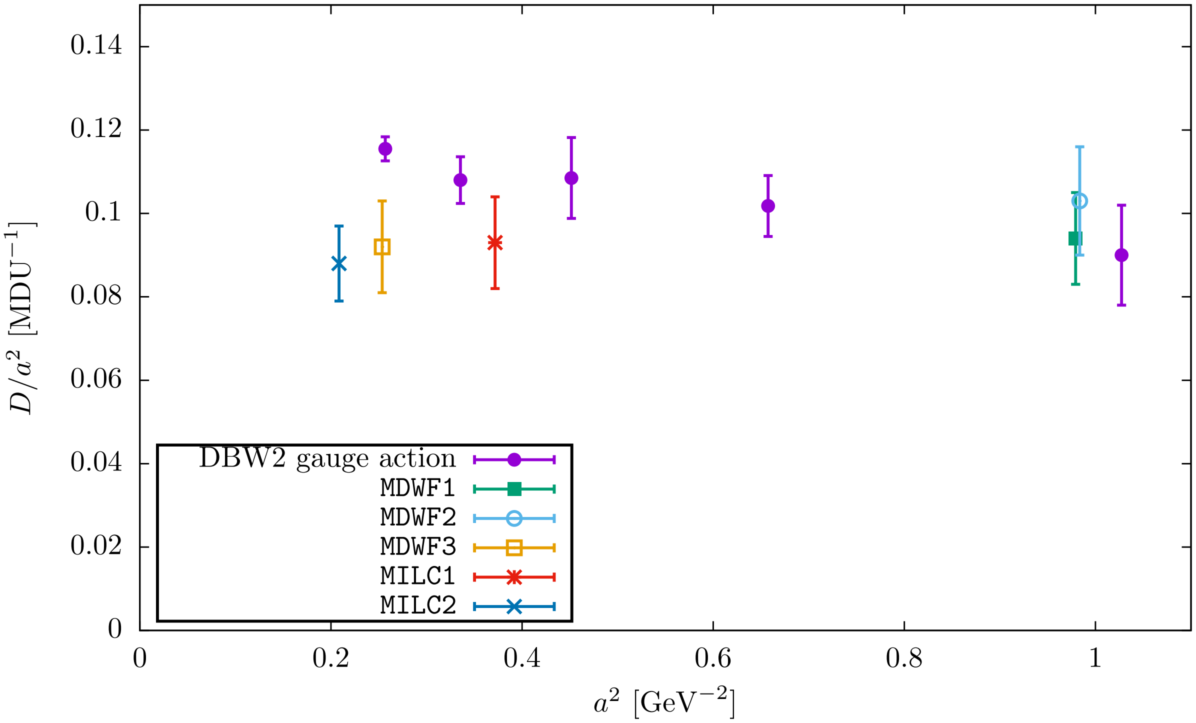

Figure 4: Plot of the diffusion constant D versus lattice spacing a^2.

dynamical lattices with lattice spacing coarser than (2 GeV)$^{-1}$ when focusing on shorter distance correlations. The long distance correlations of the topological charge observables on these coarse lattices does not follow the simple diffusion model behavior, which is different from the quenched case. At present we do not have an explanation for this observation and, since it is not present at weaker coupling, appears to be a lattice artifact. From the point of view of critical slowing down in lattice QCD, this long-distance discrepancy is not relevant given the focus on small lattice spacings when discussing critical slowing down.

We note that the value for D/a^2 is relatively similar between the quenched and dynamical lattices. The presence of fermions does not make a marked difference in the effectiveness of the HMC at moving existing topological charge density around on the lattice. Successful new algorithms beyond the traditional HMC aimed at resolving critical slowing down are expected to have a larger value for D/a^2.

3 Studies on Multiscale Lattice QCD Action Matching

3.1 Abstract

The RBC and UKQCD Collaborations have shown that light hadron masses and meson decay constants measured on 2+1 flavor Mobius DWF ensembles generated with the Iwasaki gauge action and a dislocation suppressing determinant ratio (DSDR) term show few percent $\mathcal{O}(a^2)$ scaling violations for ensembles with $a^{-1} = 1$ GeV. We call this combination the **ID**+MDWF action and this scaling implies that, to a good approximation, these ensembles lie on a renormalization group trajectory, where the form of the action is unchanged and only the bare parameters need to be tuned to stay on the trajectory. Here we investigate whether a single-step APE-like blocking kernel can reproduce this trajectory and test its accuracy via measurements of the light hadron spectrum and non-perturbative renormalization. As we report, we find close matching to the renormalization group trajectory from this simple blocking kernel.

3.2 Introduction

The RBC and UKQCD Collaborations have recently generated several $2+1$ flavor ensembles with the ID+MDWF action at three different lattice spacings and various values for the quark masses. Global fits to these ensembles and ensembles at weaker coupling generated with the Iwasaki gauge action were performed, using fit anzätz from SU(2) chiral perturbation theory, including analytic expansions for variations in the strange quark mass around its physical value [11]. A typical fit takes the form of (equation (9) in [11])

$$X(m, a^2) \simeq X_0(1 + f(m) + c_X a^2), \tag{101}$$

where X are observables, X_0 is the chiral and continuum limit and $f(m)$ is the chiral perturbation theory or analytic function giving the quark mass dependence. We observe that on ID+MDWF ensembles these c_X coefficients are typically ~ 0.02 GeV^2 for $X = f_\pi$ and $X = f_K$.[6] This leads to percent scale scaling errors on our $a^{-1} = 1$ GeV ensembles.

While measurements of more complicated quantities such as B_K and the $\Delta I = 3/2$ $K \to \pi\pi$

[6] m_π, m_K and m_Ω were used to determine the scale so these quantities have zero c_X.

matrix elements are underway, these small scaling errors imply that theories with the ID+DSDR action lie, to a good approximation, on a renormalization group (RG) trajectory, up to a few percent discrepancy. While knowing a good approximation to an RG trajectory might help in producing a multiscale evolution algorithm, it is also of interest to investigate whether a numerically tractable blocking kernel, which *carries* the transformation on this RG trajectory, exists. This latter topic is the subject of this report.

We start with the pair of ID+MDWF ensembles shown in Table 6. These have $m_\pi \sim 300$ MeV, physical kaon masses, lattice spacings that differ by close to a factor of 2 (actually 1.97) and essentially the same physical volume. We refer to these as the coarse and fine ensembles, with actions S_c and S_f respectively, which are both ID+MDWF actions. From the table one can see that they have the same physics at the 5% level and this agreement could be made closer by more careful tuning of the input parameters, but this precision is accurate enough for this study.

In general, a *blocking kernel* $G[U_c, U_f]$ [22] will map a fine configuration with links U_f to a coarse configuration with links U_c. The action for the blocked coarse lattice, $S_c^b[U_c]$ is given by

$$e^{-S_c^b[U_c]} \propto \int [dU_f] e^{-S_f[U_f]} G[U_c, U_f]. \tag{102}$$

Here we seek a numerically tractable blocking kernel that produces a blocked coarse lattice, with action $S_c^b[U_c]$, that is as close as possible to $S_c[U_c]$. We will work with a coarse and a fine ensemble whose lattice spacings differ by a factor of essentially 2. Since we will not have a closed form expression for $S_c^b[U_c]$, we will use measurements of physical quantities on the blocked, coarse lattice as a measure of the agreement between it and the original coarse action ensemble. Figure 5 gives a diagram showing our strategy.

A general blocking kernel $G[U_c, U_f]$ is a functional defined as a product of delta functions, with arguments which are SU(3) links from both the coarse and fine lattice, denoted by U_c and U_f, respectively.

$$G[U_c, U_f] = \prod_{x,\mu} \delta\left(U_c(x,\mu) - g_b[U_f; x, \mu]\right). \tag{103}$$

Here $g_b[U_f; x, \mu]$ is a function which determines the blocking kernel. Figure 6 shows the blocking kernel we experiment with in this paper. The kernel is similar to the well known APE smearing

$$\begin{tikzcd}
S_f[U_f] \arrow[r, leftrightarrow, "\text{good scaling property}"] \arrow[d, "\begin{array}{l}\text{blocking kernel}\\ G[U_c, U_f]\end{array}"'] & S_c[U_c] \arrow[dl, leftrightarrow, "\begin{array}{l}\text{compare low energy meson}\\ \text{observables and NPR factors}\end{array}"] \\
S_c^b[U_c] &
\end{tikzcd}$$

Figure 5: Strategy of our comparison.

method [23], except that it produces a coarse ensemble link from a pair of links in the fine lattice, plus staples spanning the two fine-lattice links. α is an adjustable parameter which we will determine later.

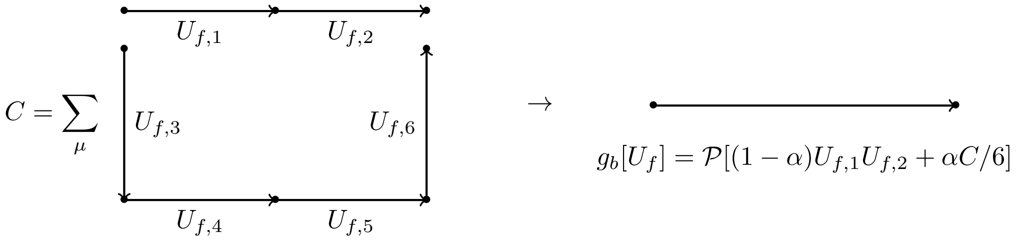

Figure 6: Single-step APE-like blocking kernel. $\mathcal{P}$ is a projection operator from a sum of SU(3) matrices back to SU(3).

3.3 Numerical Methods

We start by generating a coarse and fine ensemble with the ID+MDWF actions, whose lattice spacings differ by a factor of 2. As mentioned, the RBC and UKQCD Collaborations have seen small $\mathcal{O}(a^2)$ errors for this action. To make our studies easier, we use smaller volumes and target $m_\pi \sim 300$ MeV. Table 6 shows the results for basic observables on these ensembles. The lattice spacing comes from $\sqrt{t_0}$ and w_0 measurements and we have used the global fits results [11] to determine the input quark masses. Our input quark masses could be refined to reduce the $2-7\%$ errors seen in the table to perhaps below 3%, but we believe this agreement is accurate enough for our current purposes. Both ensembles have physical spatial volumes of about (2.4 fm)^3.

	$\langle \mathcal{O} \rangle^c$	$\langle \mathcal{O} \rangle^f$	% diff.
size	$12^3 \times 32 \times 12$	$24^3 \times 64 \times 12$	–
β	1.633	1.943	–
am_l	0.008521	0.000787	–
am_h	0.065073	0.019896	–
a^{-1}[GeV]	1.015(16)	2.001(18)	–
am_{res}	0.007439(86)	0.004522(12)	–
m_π[MeV]	307(5)	300(3)	2.3
m_K[MeV]	506(8)	491(5)	3.0
m_Ω[MeV]	1652(27)	1557(71)	5.9
f_π[MeV]	147(2)	138(2)	6.3
f_K[MeV]	166(3)	155(2)	6.8

Table 6: Parameters and measurements of the fine and coarse lattices. The lattice spacing comes from $\sqrt{t_0}$ and w_0 measurements.

Figure 7 allows us to further explain our next numerical test. The shaded plane is the space of all ID+MDWF actions, and the points locate our coarse and fine ensembles. A general RG blocking of the fine lattice will lead out of the plane, as shown by the dashed line. Assuming perfect scaling between the coarse and fine ensemble, there is some RG blocking of the fine lattice which will remain in the ID+MDWF plane; our task is to see if we can get a good approximation to this RG blocking from our single-step APE-like blocking kernel.

A simple minded way to proceed would be to choose a value for α, block the fine ensemble and measure physics observables on the resulting blocked, coarse ensemble and compare with the original coarse ensemble. We believe a better way to do this is to utilize the demon algorithm [24]. Applying this algorithm to the lattices in an ensemble, one can find the coefficients (couplings) for any term in the action which could have appeared in the generation of the ensemble, or in an effective representation of the action. Here we are generating ensembles including fermions and we can use the demon algorithm to find an action, expressed as a sum of Wilson loops, that would

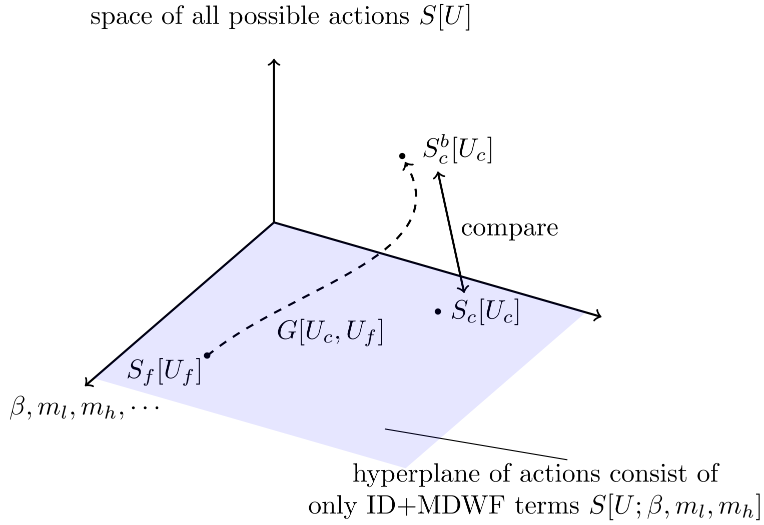
space of all possible actions $S[U]$
$S_c^b[U_c]$
compare
$S_c[U_c]$
$G[U_c, U_f]$
$S_f[U_f]$
$\beta, m_l, m_h, \cdots$
hyperplane of actions consist of
only ID+MDWF terms $S[U; \beta, m_l, m_h]$

Figure 7: Illustration of the comparison. A general blocking kernel $G[U_c, U_f]$ will lead the fine lattice action out of the hyperplane of actions consisting of only ID+MDWF terms, as shown by the dashed line. There exists some blocking kernel which will keep the fine action close to the ID+MDWF hyperplane, giving a good scaling between the coarse and fine ensemble. The complexity of the kernel is not known a priori.

produce the same ensemble.

Given a configuration generated according to some action, possibly including fermions, we can introduce a series of *demon* variables to determine the underlining β_i in an expansion of the action in terms of a set of Wilson loops, denoted by S_i. These terms can be the plaquette(P), rectangular(R), chair loop(C), twist loop(T), etc.

$$\int [DU] \int \prod_i [dE_i] \exp \left[- \sum_i \left(\beta_i S_i[U] + \beta_i E_i \right) \right].$$

The update scheme for the demon consists of two parts:

1. update U's only.

2. update U's and E_i's at the same time while keeping $S_i + E_i$ constant. In this case the accept/reject step does not require knowledge of β_i's.

Since the integration of the E_i's factorizes we can measure the average value of them and probe the underlying β_i's through the relation

$$\langle E_i \rangle = \frac{1}{\beta_i} - \frac{E_{max}}{\tanh(\beta_i E_{max})}.$$

We apply the demon algorithm on the configurations generated by $S_c[U_c]$[7], and configurations generated by $S_c^b[U_c]$ with different α's. The result is shown in table 7. We choose to use $\alpha = 0.688$, which gives the closest match for the coefficient of the plaquette, β_P, between the coarse and blocked coarse ensembles. For this choice of α, we also see that the coefficient of β_R for the coarse and blocked-coarse ensembles differ by 50%, although both are small. No further matching is possible with our single parameter blocking kernel, so we will work with this value of α.

ensemble	β_P	β_R	β_C	β_T
$S_c[U_c]$	2.035(36)	−0.1018(33)	−0.0026(30)	−0.0006(30)
$\alpha = 0.0$	0.617(11)	0.0491(33)	0.0032(32)	0.0010(32)
$\alpha = 0.5$	1.478(35)	−0.0020(44)	0.0043(42)	−0.0016(43)
$\alpha = 0.688$	2.030(28)	−0.1522(30)	−0.0021(24)	0.0038(24)
$\alpha = 0.7$	2.069(33)	−0.1589(33)	0.0009(27)	−0.0003(27)

Table 7: Results of the demon algorithm applied to different blocked coarse ensembles, produced with various values for the blocking parameter, α. The results of applying the demon algorithm to the coarse lattice are given in the first row.

3.4 Results

We can now compare physical observables on the coarse and blocked coarse lattices. For pure gauge quantities, we use the same observable to do the measurement on both ensembles. For fermionic observables, we need to calculate propagators on the blocked coarse lattice, which means we need to choose which gauge fields and quark masses to use in the MDWF Dirac operator. To the extent that the blocked coarse and coarse lattices are equivalent, we can use the same quark masses in both cases. Since the MDWF residual mass may be different between the two cases, we will present results with the total quark masses the same, *i.e.* the input quark mass plus the residual mass.

[7] Fermion determinants are in principle sums of Wilson loops. Our inclusion of only 4 types of Wilson loops is justified by the fact that we only want to determine 1 free parameter α.

Turning first to the Wilson flow, Figure 8 shows the results on the coarse and blocked coarse lattices. One sees very good agreement between the two ensembles up to very large flow times.

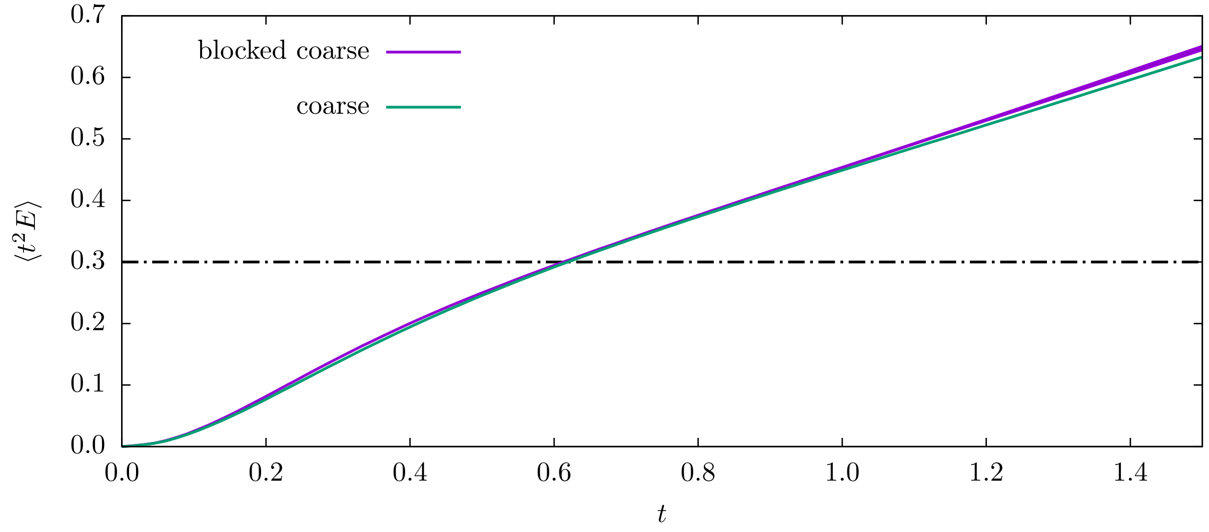

Figure 8: Wilson flow of the coarse and blocked coarse ensembles.

We have also measured some light hadron masses and meson decay constants on the two ensembles and Table 8 shows the results. The differences are at the $1-3\%$ level between the coarse and blocked coarse lattices. We note that for the coarse ensemble, we know the sea quark mass and use this mass for the measurements (they are unitary). For the blocked coarse ensemble we do not know the sea quark mass, but use the same total quark mass as the coarse ensemble. To do this, we have to measure the residual mass on the blocked coarse ensemble, which we do by choosing the fifth dimension, L_s, to be 12, as it is on the coarse ensemble. (The residual mass does depend modestly on the quark mass used in the measurement. This is generally a small effect and we do not correct for it here.) Our condition on the quark masses is

$$[am_l + am_{\text{res}}]_{\text{coarse}} = [am_l + am_{\text{res}}]_{\text{blocked coarse}}. \tag{104}$$

It is worth pointing out that by using the same total quark mass on the coarse and blocked coarse ensembles, and finding the same physical hadron masses, we are implicitly seeing that the renormalization factors for the quarks are the same on these two ensembles. This is a statement not just about the equivalence of long distance physics on both ensembles, but also includes at least

	$\langle\mathcal{O}\rangle^c$	$\langle\mathcal{O}\rangle^{bc}$	% diff.
size	$12^3 \times 32 \times 12$	$12^3 \times 32 \times 12$	–
β	1.633	–	–
am_l	0.008521*	0.007494	–
am_h	0.065073*	0.064150	–
a^{-1}[GeV]	1.015(16)	1.010(16)	0.5(2.3)
am_{res}	0.007439(86)	0.00847(21)	–
am_π	0.3026(13)	0.3050(35)	0.8(1.2)
am_K	0.4982(11)	0.5018(25)	0.7(0.5)
am_Ω	1.628(10)	1.668(25)	2.4(1.6)
af_π	0.14472(64)	0.1491(10)	2.9(0.8)
af_K	0.16333(47)	0.16723(85)	2.3(0.6)

Table 8: Spectrum measurements on coarse and blocked coarse lattices. See section 3.4 for the choice of valence quark mass on the blocked coarse ensemble. Again the lattice spacing comes from $\sqrt{t_0}$ and w_0 measurements.

some of the short distance physics.

To pursue this question further, we have also measured the non-perturbative renormalization (NPR) factors for quark bilinears on the coarse and blocked coarse ensembles using the RI/SMOM scheme [25]. These NPR factors renormalize quark bilinear operators multiplicatively, e.g. for the vector bilinear operator,

$$\left[\bar{\psi}\gamma_\mu\psi\right]_{\text{renormalized}} = \frac{Z_V}{Z_q}\left[\bar{\psi}\gamma_\mu\psi\right]_{\text{lattice}}. \tag{105}$$

A detailed introduction to NPR can be found in section 4.1.

We find generally very good agreement between the two measurements, with differences of $\sim 1\%$ over a range of renormalization scales. Figure 9-13 shows the NPR Z factors, for the bilinear operators, as a function of the renormalization scale on the two ensembles. One sees that the differences of Z's at high energy scales (compared to the lattice spacing) are around 1%. Given the matching between low energy measurements, this further shows that the two actions, $S_c[U_c]$ and $S_c^b[U_c]$ can be considered as the same action, up to an accuracy of a few percent.

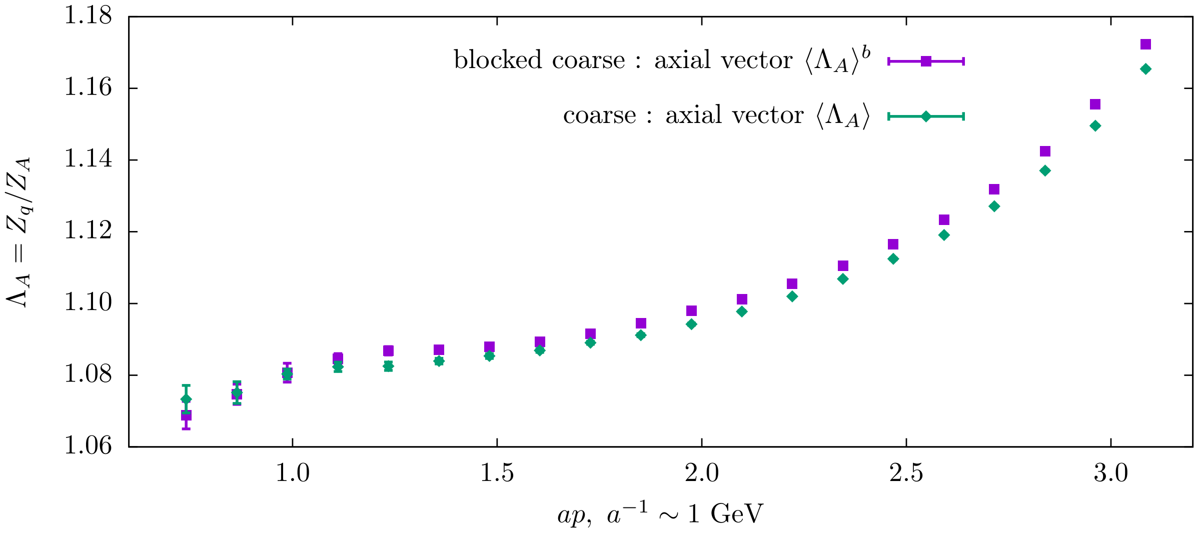

Figure 9: Non-perturbative renormalization factor Z_A of the coarse and blocked coarse ensembles.

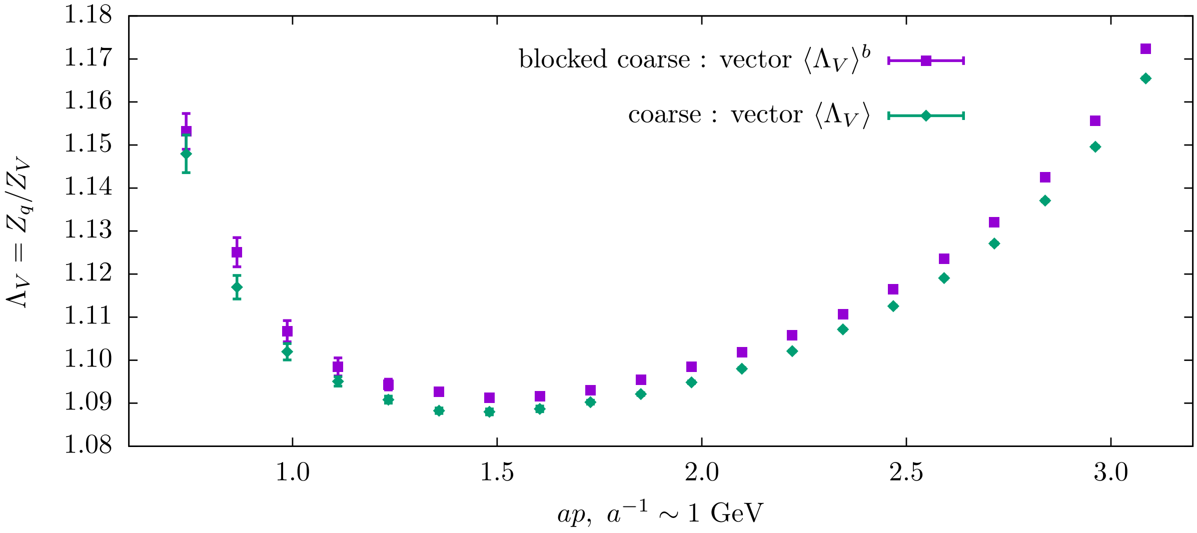

Figure 10: Non-perturbative renormalization factor Z_V of the coarse and blocked coarse ensembles.

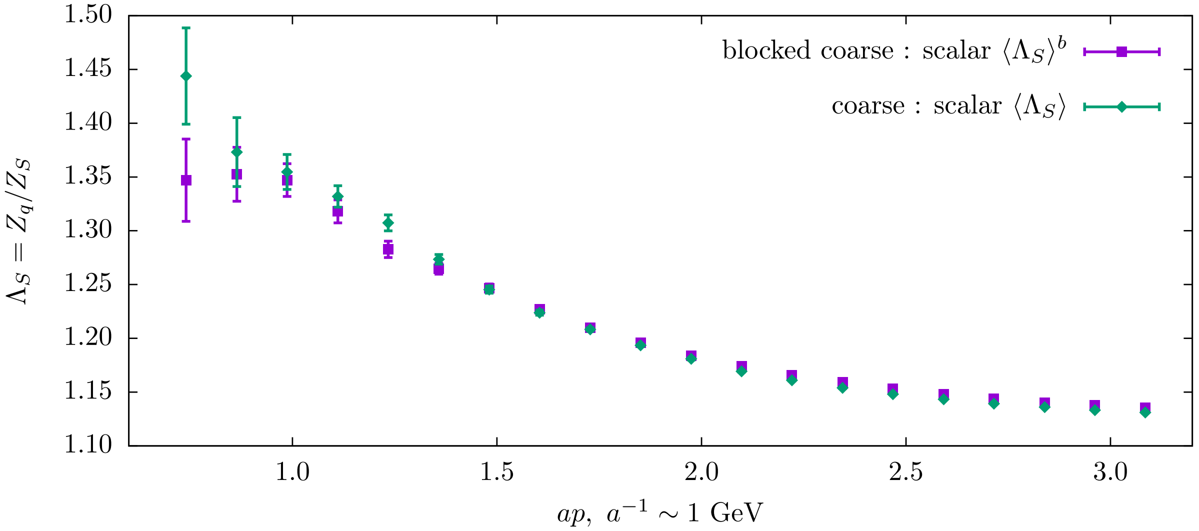

Figure 11: Non-perturbative renormalization factor Z_S of the coarse and blocked coarse ensembles.

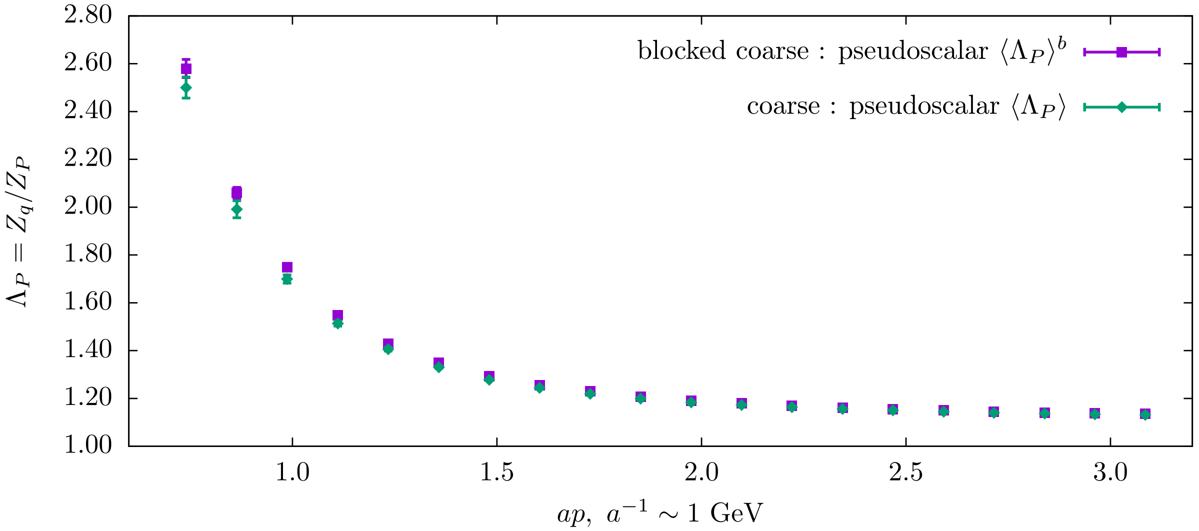

Figure 12: Non-perturbative renormalization factor Z_P of the coarse and blocked coarse ensembles.

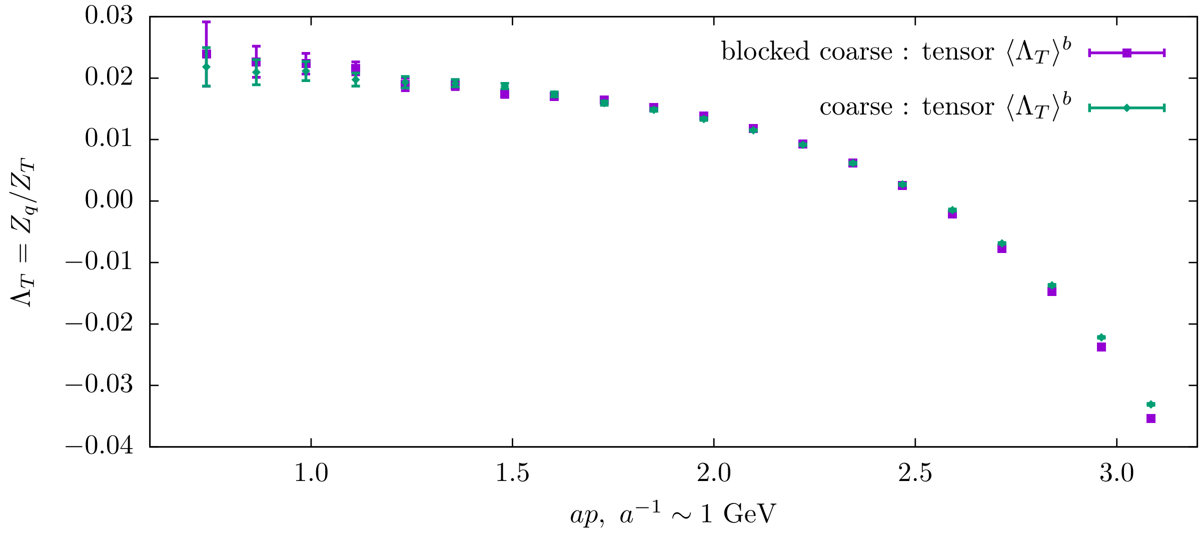

Figure 13: Non-perturbative renormalization factor Z_T of the coarse and blocked coarse ensembles.

3.5 Conclusion

We have generated a matched pair of lattices, coarse and fine, with 300 MeV pions and lattice spacings differing by a factor of 2, using the ID+MDWF actions. Using the demon algorithm, a simple APE-like blocking kernel is tuned to block the fine lattice into a blocked coarse lattice. Measurements of physical hadron masses, meson decay constants and the Wilson flow scale were made on the coarse and blocked coarse ensembles. The results show only a few percent difference. We have also measured renormalization factors using non-perturbative renormalization on the coarse and blocked-coarse ensembles and find good agreement between them. The good scaling properties of the ID+MDWF action imply that the coarse and fine lattices are related, to a good approximation, by an RG blocking transformation. We have found that a simple, numerically tractable, blocking kernel provides a good RG blocking transformation between these lattices.

Beyond understanding the properties of these ensembles better, we were also motivated to undertake this study to think about ways of improving evolution algorithms. If $S_c[U_c]$ and $S_c^b[U_c]$ are very nearly equal, we could imagine evolving both coarse and fine lattice concurrently, by writing the following:

$$\begin{aligned}
\langle \mathcal{O} \rangle &= \frac{\int [dU_f] e^{-S_f[U_f]} \mathcal{O}[U_f]}{\int [dU_f] e^{-S_f[U_f]}} && (106) \\
&= \frac{\int [dU_c] G[U_c, U_f] [dU_f] e^{-S_f[U_f]} \mathcal{O}[U_f]}{\int [dU_c] G[U_c, U_f] [dU_f] e^{-S_f[U_f]}} && (107) \\
&= \frac{\int [dU_c] e^{-S_c^b[U_c]} \frac{\int [dU_f] e^{-S_f[U_f]} G[U_c, U_f] \mathcal{O}[U_f]}{\int [dU_f] e^{-S_f[U_f]} G[U_c, U_f]}}{\int [dU_c] e^{-S_c^b[U_c]}} && (108)
\end{aligned}$$

In the second line we have inserted a $[dU_c]G[U_c, U_f]$ factor which is a constant when integrated over U_c and the third line uses the definition of the blocked action, equation (102).

Equation (108) states that we will get the same value for observables if we perform *constrained* Monte Carlo on the fine lattice in the background of coarse lattices generated with the action $S_c^b[U_c]$. This is a further step from what is described in [26]: this is a multiscale *evolution* algorithm, rather than just a multiscale *thermalization* algorithm. The coarse (free) evolution and the fine constrained evolution can be performed successively with no additional thermalization needed. The coarse evolution gives us short decorrelation length for various quantities including the global topological charge and the fine evolution fills in the much wanted short scale details, which enable the inclusion of heavy quarks.

Unfortunately, an exact evolution algorithm requires that $S_c[U_c]$ and $S_c^b[U_c]$ agree much more accurately than we have seen here. A viable algorithm would need a correction step or a reweighting factor, or other improvements, to facilitate such a multiscale Monte Carlo.

4 Physical Output from the Coarse DSDR Lattice

The RBC/UKQCD collaborations have generated coarse lattice configurations at inverse lattice spacing of around 1 GeV with Iwasaki gauge action, dislocation suppressed determinant ratio (DSDR) and Möbius domain wall fermions (MDWF) for degenerate light quarks with physical masses and a physical mass strange quark (2+1 flavor). The coarse lattice spacing makes it possible to have a large physical volume (e.g. around 4.8 fm$^3 \times 9.6$ fm) with a relatively small lattice volume (e.g. $24^3 \times 64$). Given a fixed amount of machine resources this greatly increases the number of configurations that can be generated as well as the amount of statistics that can be achieved during the measurement phase. These coarse lattices are also ideal testing ground for new measurements of more complicated observables with physical quark masses.

Measurements made on coarse lattices can carry large finite lattice spacing errors, due to the large lattice spacing a. A natural question to ask is whether or not we can reliably estimate the finite lattice spacing errors and make corresponding correction to give physics predictions in the continuum. While global chiral fits results excluding these $a^{-1} = 1$ GeV coarse lattices suggest small $\mathcal{O}(a^2)$ error for ensembles generated with the Iwasaki/DSDR (**ID**) and MDWF actions, it is interesting to see if the simple analytic form of

$$X(m, a^2) \simeq X_0(1 + f(m) + c_X^{\mathbf{A}} a^2) \tag{109}$$

in the global chiral fit still holds at this coarse a or if measurable a^4 and higher order terms are needed. If the answer is yes, while the finite lattice spacing correction $c_X^{\mathbf{A}} a^2$ can be large at this coarse a, the simple linear form puts the a^2 dependence under control and consequently, without the higher order terms in a^2, the continuum limit extrapolation can be performed much more reliably.

Here we distinguish the a^2 coefficient $c_X^{\mathbf{A}}$ between the **ID** and the **IW** lattices as $c_X^{\mathbf{ID}}$ and $c_X^{\mathbf{IW}}$. Lattices with both actions have the same continuum ChPT form $f(m)$ and continuum limit prediction X_0. For a specific physical observable, measurements made on **ID** lattices with different lattice spacing a, together with its continuum limit are examined to see if the data points fit to the form shown in (109).

Recent results of global chiral fits [11] that includes the basic physical quantities[8] measured on the coarse lattices confirms the above description is still valid yet it is still worthwhile to investigate this with more complicated quantities. In this work we measure the neutral kaon mixing parameter B_K and the $\Delta I = 3/2$ $K \to \pi\pi$ amplitude A_2 on the coarse 24**ID**($\beta = 1.633$) lattice[9]. Both quantities require non-perturbative renormalization which is done at this coarse lattice spacing for the first time.

Our investigation gives us a better understanding of the finite lattice spacing effect on the more noisier $\Delta I = 1/2$ $K \to \pi\pi$ process and ϵ'/ϵ that are currently being measured on the **ID** lattices with $a^{-1} = 1.37$ GeV.

4.1 Non-perturbative Renormalization

Typically, renormalization scheme dependent quantities are expressed in dimensional regularization schemes such as the $\overline{\text{MS}}$ scheme. These schemes parametrize the divergences of the gauge theory with a non-integer variadic number of dimensions and relies on perturbative expansion of the coupling constant, both of which are not feasible in a non-perturbative lattice gauge theory simulation.

One approach to translating scheme dependent quantities measured on the lattice to those expressed in dimensional regularization schemes is to have an intermediate scheme which can be used with both a lattice regularization and dimensional regularization. The lattice results are first renormalized to this intermediate scheme with renormalization factors measured on the lattice at an energy scale μ low enough to suppress finite lattice spacing errors:

$$\mu \ll \frac{\pi}{a}. \tag{110}$$

If μ is low enough that perturbative QCD theory is not reliable the target quantity is first step scaled to a higher scale μ' such that

$$\mu \to \mu' \gg \Lambda_{\text{QCD}} \tag{111}$$

through another lattice calculation on a finer lattice where both scales μ and μ' can be used. This

[8] such as m_π, m_K, m_Ω, f_π and f_K.

[9] See table 1

quantity is then converted to the target dimensional regularization scheme with the conversion factor calculated perturbatively at μ'. Note that the desired physical quantity (generally a matrix element) is only calculated on the coarse lattice at scale μ.

As a summary if the lattice simulation itself is also viewed as a renormalization/regularization scheme ("lat") for a matrix element M^{lat} measured on the lattice the following conversion factors are applied before arriving at the corresponding matrix element in the $\overline{\mathrm{MS}}$ scheme:

$$M^{\overline{\mathrm{MS}}} = R^{\mathrm{IS}\to\overline{\mathrm{MS}}}(\mu')\sigma(\mu \to \mu')Z^{\mathrm{lat}\to\mathrm{IS}}(\mu)M^{\mathrm{lat}}, \tag{112}$$

where IS means the intermediate scheme. Later in this section the subscripts indicating the source and destination schemes are skipped for simplicity. Generally under renormalization the underlying operators of the matrix elements mix and as a result (112) generally indicates a matrix multiplication.

The non-perturbative renormalization (NPR) calculations in this work are performed with the Rome-Southampton regularization independent momentum-subtraction renormalization scheme with non-exceptional, symmetric subtraction point (RI-SMOM) [27] as the intermediate scheme.

4.1.1 Renormalization of Bilinear Operators

From the lattice scheme to RI-SMOM, multiplicative conversion factors (Z's) are introduced for the quark fields,

$$\text{quark: } Z_q^{-1/2}\psi \tag{113}$$

as well as each of the five bilinear operators,

$$\text{vector: } (Z_V/Z_q)\bar{\psi}\gamma_\mu\psi, \tag{114}$$

$$\text{axial vector: } (Z_A/Z_q)\bar{\psi}\gamma_\mu\gamma_5\psi, \tag{115}$$

$$\text{scalar: } (Z_S/Z_q)\bar{\psi}\psi, \tag{116}$$

$$\text{pseudo scalar: } (Z_P/Z_q)\bar{\psi}\gamma_5\psi, \tag{117}$$

$$\text{tensor: } (Z_T/Z_q)\bar{\psi}[\gamma_\mu, \gamma_\nu]\psi. \tag{118}$$

In the RI-SMOM scheme, the renormalization factors are determined by first measuring $\bar{\psi}\Gamma\psi$ on the target lattice and then determining (Z_Γ/Z_q) such that $(Z_\Gamma/Z_q)\bar{\psi}\Gamma\psi$ has its[10] tree level value at the scale μ.

Momentum p is inserted into and/or extracted from the bilinear operators by constructing the Fourier transformed *volume source propagator* with a Fourier transformed source and sink,

$$\mathcal{G}(p) = \sum_{x,y} e^{-ip\cdot y}\xi(y,p), \tag{119}$$

where ξ is the volume source propagators and

$$\xi(y,p) = D^{-1}(y,x)e^{ip\cdot x}\mathbf{1}_4 \otimes \mathbf{1}_3, \tag{120}$$

where $D(y,x)^{-1}$ is the usual four dimensional propagator and $\mathbf{1}_4 \otimes \mathbf{1}_3$ is the identity matrix in spin-color. Only a finite number of discrete momenta are allowed in this approach due to the restriction on a lattice with finite volume. Another approach is proposed and used in [28] which allows continuous momenta by applying twist boundary condition on the quark fields in μ-direction,

$$q(x + L\hat{\mu}) = e^{i\theta}q(x),\ \theta/L = p. \tag{121}$$

Now to measure $\bar{\psi}\Gamma\psi$ on the lattice we measure the following *vertex* for operator Γ with non-exceptional momenta p_1 and p_2 with $p_1^2 = p_2^2 = (p_1 - p_2)^2 = q^2$: quark source and sink are attached to the bilinear to form the propagators,

$$\Big[\mathcal{F}_\Gamma(p_1,p_2)\Big]_{ij} = \Big\langle \sum_y e^{i(p_1-p_2)\cdot y}\psi_i(y,p_1)\Big[\bar{\psi}\Gamma\psi\Big](y)\bar{\psi}_j(y,p_2)\Big\rangle \tag{122}$$

$$\overset{\text{Wick}}{=} \sum_y e^{i(p_1-p_2)\cdot y}\Big[\gamma_5\xi^\dagger(y,p_1)\gamma_5\Gamma\xi(y,p_2)\Big]_{ij}, \tag{123}$$

where i,j,k are spin-color indices and the Wick contractions are performed in the second equal sign. The statistical average of the vertex over gauge configurations ($\langle\bullet\rangle_g$) is further amputated by the

[10] the RI-SMOM scheme

propagators,

$$\Big[\Pi_\Gamma(p_1,p_2)\Big]_{ij} = \Big[\gamma_5\langle\mathcal{G}^\dagger(p_1)\rangle_g^{-1}\gamma_5\langle\mathcal{F}_\Gamma(p_1,p_2)\rangle_g\langle\mathcal{G}(p_2)\rangle_g^{-1}\Big]_{ij}. \tag{124}$$

The amputated vertices are still spin-color matrices and they are projected to their corresponding projection operator $\mathcal{P}_\Gamma^{(s)}$'s before the renormalization condition is applied

$$\frac{Z_\Gamma}{Z_q}\frac{\mathrm{tr}\left[\Pi_\Gamma\mathcal{P}_\Gamma^{(s)}\right]}{\mathrm{tr}\left[\Gamma\mathcal{P}_\Gamma^{(s)}\right]} = \frac{Z_\Gamma}{Z_q}\Lambda_\Gamma = 1. \tag{125}$$

Here the subscript (s) indicates the different projection operators corresponds to different schemes in RI-SMOM and $\mathrm{tr}\left[\Gamma\mathcal{P}_\Gamma^{(s)}\right]$ is the tree level value in RI-SMOM. For the γ_μ scheme the projection operator are simply the Γ itself, as shown in table 9; for the $\not{q}$ scheme the projector depends on the input momentum $q = p_1 - p_2$. The two schemes are equivalent in the limit of vanishing finite lattice spacing error yet in the $\not{q}$ scheme it is more substantially suppressed. [27] The existence of these two different schemes allows us to use the difference between their corresponding results to estimate the systematic error from our NPR procedures.

operator name	Γ	$\mathcal{P}_\Gamma^{(\gamma_\mu)}$	$\mathcal{P}_\Gamma^{(\not{q})}$	$\mathrm{tr}\left[\Gamma\mathcal{P}_\Gamma^{(s)}\right]$
vector (V)	γ_μ	γ_μ	$\not{q}\hat{q}_\mu/\not{q}^2$	48
axial vector (A)	$\gamma_\mu\gamma_5$	$\gamma_\mu\gamma_5$	$\not{q}\hat{q}_\mu\gamma_5/\not{q}^2$	48
scalar (S)	$\mathbf{1}$	$\mathbf{1}$	$\mathbf{1}$	12
pseudo scalar (P)	γ_5	γ_5	γ_5	12
tensor (T)	$[\gamma_\mu,\gamma_\nu]$	$[\gamma_\mu,\gamma_\nu]$	$[\gamma_\mu,\gamma_\nu]$	72

Table 9: The projection operators of the γ_μ and $\not{q}$ schemes in RI-SMOM.

4.1.2 Renormalization of the $\Delta S = 2$ Four-quarks Operators

The original $\Delta S = 2$ four-quark operator $\mathcal{O}_{\Delta S=2}$ related to the neutral kaon mixing in the standard model is

$$\mathcal{O}_{\Delta S=2} = \Big[\bar{s}\gamma_\mu(1-\gamma_5)d\Big]\Big[\bar{s}\gamma_\mu(1-\gamma_5)d\Big]. \tag{126}$$

This operator comes from [29]. The spin structure γ_μ's come from the weak interaction between quarks and the W boson W_μ and the $(1-\gamma_5)$ factors come from the fact that only left-handed fermion participate in weak interactions. For lattice NPR calculation purposes it is convenient to only consider the parity even part of the operator

$$\mathcal{O}_{VV+AA} = \Big[\bar{s}\gamma_\mu d\Big]\Big[\bar{s}\gamma_\mu d\Big] + \Big[\bar{s}\gamma_\mu\gamma_5 d\Big]\Big[\bar{s}\gamma_\mu\gamma_5 d\Big]. \tag{127}$$

The $\mathcal{O}_{VV+AA}$ operator transforms under the $(\mathbf{27},\mathbf{1})$ representation[11] of the SU$(3)_L\times$ SU$(3)_R$ chiral transformation for massless QCD. While in the standard model it is the only $\Delta S = 2$ operator there are 4 more operators possible beyond the standard model and they are listed in table 10. Under renormalization these operators mix with each other and a 5 by 5 matrix Z factor is introduced

$$\frac{Z_{ab}\mathcal{O}_b}{Z_q^2}, \tag{128}$$

where a and b are operator indices. On lattices with good chiral symmetry there is little mixing between operators under different representation.

Measuring beyond the standard model (BSM) $\Delta S = 2$ renormalization factors is interesting on its own: with these renormalization factors available the BSM theories are able to give physical predictions on neutral kaon mixings. [30] In addition measuring the $\Delta I = 3/2$ $K \to \pi\pi$ amplitude A_2 also needs renormalization factors for the $(\mathbf{27},\mathbf{1})$ and $(\mathbf{8},\mathbf{8})$ operators.

Similar to the bilinear case quark sources and sinks are attached to the operator to form vertices, except in this case the Wick contraction structure is more complicated,

$$\Big[\mathcal{F}_{\Gamma\Gamma}(p_1,p_2)\Big]_{ijkl} = \Big\langle \sum_y e^{i\cdot 2(p_1-p_2)\cdot y} s_i(y,p_1)\bar{d}_j(y,p_2)\Big[\bar{s}\Gamma d\bar{s}\Gamma d\Big](y) s_k(y,p_1)\bar{d}_l(y,p_2)\Big\rangle \tag{129}$$

$$= \sum_y e^{i\cdot 2(p_1-p_2)\cdot y}\Big\{2\cdot\Big[\gamma_5\xi^\dagger(y,p_1)\gamma_5\Gamma\xi(y,p_2)\Big]_{ij}\Big[\gamma_5\xi^\dagger(y,p_1)\gamma_5\Gamma\xi(y,p_2)\Big]_{kl} \tag{130}$$

$$-2\cdot\Big[\gamma_5\xi^\dagger(y,p_1)\gamma_5\Gamma\xi(y,p_2)\Big]_{il}\Big[\gamma_5\xi^\dagger(y,p_1)\gamma_5\Gamma\xi(y,p_2)\Big]_{kj}\Big\} \tag{131}$$

[11] See appendix A.4 for the derivation of the representations.

The vertex is then amputated

$$\Big[\Pi_{\Gamma\Gamma}(p_1,p_2)\Big]_{ijkl} = \tag{132}$$

$$\Big[\gamma_5\langle\mathcal{G}^\dagger(p_1)\rangle^{-1}\gamma_5\Big]_{im}\Big[\gamma_5\langle\mathcal{G}^\dagger(p_1)\rangle^{-1}\gamma_5\Big]_{ko}\Big[\langle\mathcal{F}_{\Gamma\Gamma}(p_1,p_2)\rangle\Big]_{mnop}\Big[\langle\mathcal{G}(p_2)\rangle^{-1}\Big]_{nj}[\langle\mathcal{G}(p_2)\rangle^{-1}\Big]_{pl}. \tag{133}$$

and linearly combined to form the vertex in a certain representation,

$$\Pi_{\Gamma\Gamma\pm\Sigma\Sigma}(p_1,p_2) = \Pi_{\Gamma\Gamma}(p_1,p_2) \pm \Pi_{\Sigma\Sigma}(p_1,p_2). \tag{134}$$

The two Γ's in the four-quark operator give more flexibility in choosing the projection operator. Each of the Γ's can be projected to either the γ_μ or the $\not{q}$ scheme and for the full operator there are four combinations: (γ_μ,γ_μ), $(\gamma_\mu,\not{q})$, $(\not{q},\gamma_\mu)$ and $(\not{q},\not{q})$. In table 10 we list the corresponding projection operators for the first and last combination. The spin-color tensor convention in the table is listed in (139) and (140). With a chosen scheme the vertices are calculated with

$$\Lambda_{ab} = \Big[\mathcal{P}_a\Big]_{jilk}\Big[\Pi_b\Big]_{ijkl}. \tag{135}$$

The conversion matrix Z_{ab} is determined by applying the renormalization condition

$$\frac{Z_{ab}}{Z_q^2}\Lambda_{bc} = F_{ac},\ \frac{Z_{ac}}{Z_q^2} = F_{ab}\Lambda^{-1}_{bc}, \tag{136}$$

where F are the tree level values in RI-SMOM in a specific scheme, e.g.

$$F^{(\gamma_\mu,\gamma_\mu)}_{ab} = \begin{pmatrix} 3072 & & & & \\ & 2304 & -384 & & \\ & -384 & 576 & & \\ & & & 480 & 288 \\ & & & 288 & 2016 \end{pmatrix}, \tag{137}$$

$$F_{ab}^{(\not{q},\not{q})} = \begin{pmatrix} 768 & & & & \\ & 576 & 192 & & \\ & -96 & -288 & & \\ & & & 432 & 144 \\ & & & 720 & 1008 \end{pmatrix}. \tag{138}$$

index	operator	rep.	$\mathcal{P}^{(\gamma_\mu,\gamma_\mu)}$	$\mathcal{P}^{(\not{q},\not{q})}$
1	$VV+AA$	$(\mathbf{27},\mathbf{1})$	$[\gamma_\mu \times \gamma_\mu + \gamma_\mu\gamma_5 \times \gamma_\mu\gamma_5]_{\text{diag}}$	$\frac{1}{q^2}[\not{q} \times \not{q} + \not{q}\gamma_5 \times \not{q}\gamma_5]_{\text{diag}}$
2	$VV-AA$	$(\mathbf{8},\mathbf{8})$	$[\gamma_\mu \times \gamma_\mu - \gamma_\mu\gamma_5 \times \gamma_\mu\gamma_5]_{\text{diag}}$	$\frac{1}{q^2}[\not{q} \times \not{q} - \not{q}\gamma_5 \times \not{q}\gamma_5]_{\text{diag}}$
3	$SS-PP$		$[1 \times 1 - \gamma_5 \times \gamma_5]_{\text{diag}}$	$\frac{1}{q^2}[\not{q} \times \not{q} - \not{q}\gamma_5 \times \not{q}\gamma_5]_{\text{mix}}$
4	$SS+PP$	$(\mathbf{6},\bar{\mathbf{6}})$	$[1 \times 1 + \gamma_5 \times \gamma_5]_{\text{diag}}$	$\frac{1}{2q^2}q_\mu q_\nu[\sigma_{\mu\rho} \times \sigma_{\nu\rho} + \sigma_{\mu\rho}\gamma_5 \times \sigma_{\nu\rho}\gamma_5]_{\text{diag}}$
5	TT		$\frac{1}{2}[\sigma_{\mu\nu} \times \sigma_{\mu\nu}]_{\text{diag}}$	$\frac{1}{2q^2}q_\mu q_\nu[\sigma_{\mu\rho} \times \sigma_{\nu\rho} + \sigma_{\mu\rho}\gamma_5 \times \sigma_{\nu\rho}\gamma_5]_{\text{mix}}$

Table 10: The beyond standard model $\Delta S = 2$ operators, their representation and the corresponding projection operators. See (139) and (140) for the notations for the projectors.

$$[[\,\bullet \times \bullet\,]_{\text{diag}}]^{ab,cd}_{\alpha\beta,\delta\gamma} = [\,\bullet\,]_{\alpha\beta}[\,\bullet\,]_{\delta\gamma}\delta^{ab}\delta^{cd}; \tag{139}$$

$$[[\,\bullet \times \bullet\,]_{\text{mix}}]^{ab,cd}_{\alpha\beta,\delta\gamma} = [\,\bullet\,]_{\alpha\beta}[\,\bullet\,]_{\delta\gamma}\delta^{ad}\delta^{cb}. \tag{140}$$

4.1.3 Renormalization factors of the $\Delta S = 2$ Operators on the Coarse $24^3 \times 64$ Lattice

Here we show the results for the renormalization factor Z of the $\Delta S = 2$ operators on the coarse $24^3 \times 64$ lattice to the (γ_μ, γ_μ) and $(\not{q}, \not{q})$ schemes in RI-SMOM. The calculation is done on configurations 300 to 440 with increment of 4 and a jackknife bin size of 2. The calculation is done with non-exceptional momenta $p_1 = (-x, 0, x, 0)$ and $p_2 = (0, x, x, 0)$ where $x = 3.8474 \cdot \frac{2\pi}{24}$ thus giving $\mu = 1.4363$ GeV.

The coarse $24^3 \times 64$ lattice has a small chiral symmetry breaking (i.e. small residual mass) thus the mixing between operators under different representations is expected to be small. We show both the block-diagonal (BD) result, where the matrix elements in Λ that connect between different representations are explicitly zeroed before being inverted

$$\frac{Z^{(\gamma_\mu,\gamma_\mu)}}{Z_A^2}[\mathrm{BD}] = \begin{pmatrix} 0.88276(14) & & & & \\ & 1.04748(60) & 0.37374(81) & & \\ & 0.03721(59) & 0.7406(14) & & \\ & & & 0.7968(16) & -0.00832(17) \\ & & & -1.4511(40) & 1.08711(70) \end{pmatrix}, \tag{141}$$

$$\frac{Z^{(\not{a},\not{a})}}{Z_A^2}[\mathrm{BD}] = \begin{pmatrix} 0.96777(35) & & & & \\ & 1.05392(85) & 0.3768(11) & & \\ & 0.07818(92) & 0.9340(24) & & \\ & & & 0.9876(20) & -0.01032(21) \\ & & & -1.7986(48) & 1.34750(90) \end{pmatrix}, \tag{142}$$

as well as the non-block-diagonal (NBD) result where no zeroing is done,

$$\frac{Z^{(\gamma_\mu,\gamma_\mu)}}{Z_A^2}[\mathrm{NBD}] = \begin{pmatrix} 0.88276(14) & -0.00226(39) & -0.00197(43) & 0.00195(41) & -0.000053(53) \\ -0.00202(32) & 1.04749(60) & 0.37360(81) & -0.00619(88) & -0.000167(90) \\ -0.00023(21) & 0.03721(59) & 0.7410(14) & 0.0197(15) & -0.000087(92) \\ 0.00027(35) & 0.00028(31) & 0.0170(12) & 0.7972(16) & -0.00833(17) \\ 0.00101(80) & -0.0019(11) & 0.0172(28) & -1.4506(40) & 1.08711(70) \end{pmatrix}, \tag{143}$$

$$\frac{Z^{(\not{a},\not{a})}}{Z_A^2}[\mathrm{NBD}] = \begin{pmatrix} 0.96777(35) & -0.00061(35) & -0.00254(79) & 0.00209(45) & -0.000058(58) \\ -0.00131(32) & 1.05392(84) & 0.3766(11) & -0.00620(88) & -0.000168(90) \\ -0.00008(41) & 0.07816(91) & 0.9345(24) & 0.0244(19) & -0.00011(11) \\ -0.00024(52) & -0.00052(77) & 0.0194(21) & 0.9882(20) & -0.01032(21) \\ -0.0006(13) & -0.0062(20) & 0.0156(62) & -1.7981(48) & 1.34749(90) \end{pmatrix}. \tag{144}$$

The mixing between different representations is indeed small.

4.2 The Neutral Kaon Mixing Parameter B_K

The neutral kaon mixing parameter B_K plays a central role in explaining the short distance part of the K_L-K_S mass difference within the standard model. The short distance part involves evaluating the box diagrams shown in figure 14 that allow neutral kaon states $|\overline{K^0}\rangle$ and $|K^0\rangle$ to mix with each other. Following the operator product expansion, under the assumption that the energy scale in the process is much lower than the energy scale of the W boson m_W, it is factorized into a high energy part where a perturbative calculation can determine the Wilson coefficients and a non-perturbative part where a lattice calculation is needed to evaluate the matrix elements.

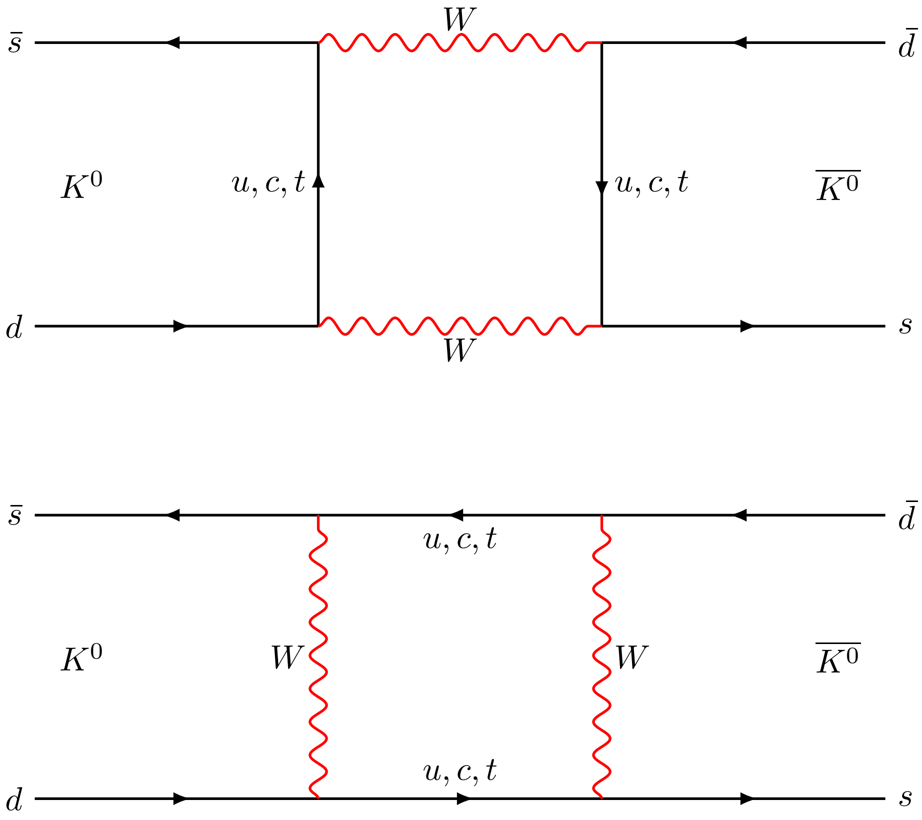

Figure 14: The box diagrams that mix the two neutral kaon states.

B_K is defined as [29]

$$\langle\overline{K^0}|\mathcal{O}_{\Delta S=2}|K^0\rangle = \langle\overline{K^0}|\Big[\bar{s}\gamma_\mu(1-\gamma_5)d\Big]\Big[\bar{s}\gamma_\mu(1-\gamma_5)d\Big]|K^0\rangle = \frac{8}{3}B_K f_K^2 m_K^2, \tag{145}$$

where f_K is the kaon decay constant and m_K is the kaon mass. B_K is defined in the renormalization scheme (usually the $\overline{\text{MS}}$ scheme) and at the energy scale μ where the matrix element is evaluated. Following (112) the path from the matrix element measured on the lattice, which will be denote as $B_K[\text{lat}]$, to that renormalized within the $\overline{\text{MS}}$ scheme $B_K[\overline{\text{MS}}, \mu]$ is

$$B_K[\overline{\text{MS}}, \mu] = B_K[\text{lat}] \cdot Z_{B_K}[(s,s), \mu_0] \cdot \sigma[(s,s), \mu_0 \to \mu] \cdot R[(s,s) \to \overline{\text{MS}}, \mu] = B_K[\text{lat}] \cdot Z[\text{lat} \to \overline{\text{MS}}, \mu]. \tag{146}$$

4.2.1 The Renormalization Factors for B_K

In (146) $Z_{B_K}[(s,s), \mu_0]$ is the renormalization factor for B_K in the (s,s) scheme under RI-SMOM at a relatively low energy scale μ_0. In the definition (145) f_K is defined as

$$\langle 0|A_0|K^0\rangle = f_K m_K, \tag{147}$$

which needs an additional renormalization factor of Z_A. Therefore for B_K

$$Z_{B_K} = \frac{Z_{VV+AA}}{Z_A^2}. \tag{148}$$

The results for these factors are already given in the previous section. The $R[(s,s) \to \overline{\text{MS}}, \mu]$ factor in (146) is the conversion factor from (s,s) scheme under RI-SMOM to $\overline{\text{MS}}$ at a higher energy scale μ and is calculated perturbatively in [31][12]. Finally The Step Scaling Factor $\sigma[(s,s), \mu_0 \to \mu]$ step scales the Z factor from μ_0 to μ within the (s,s) scheme [17][13]. The values used in this work are summarized in table 11, where results from different intermediate schemes (γ_μ, γ_μ) and $(\not{q}, \not{q})$ are reported. In addition another intermediate energy scale ($\mu_0 = 1.1199$ GeV) different from the previous 1.4363 GeV value is also used to give a check on whether or not it actually suppresses the finite lattice spacing error as discussed in (110).

[12] See TABLE V of [31].

[13] See TABLE XLIII of [17].

		(γ_μ, γ_μ)	$(\not{q}, \not{q})$
	$R[(s,s) \to \overline{\mathrm{MS}}, \mu]$	1.00414	0.99112
$\mu_0 = 1.1199$ GeV	$Z_{B_K}[(s,s), \mu_0]$	0.89481(51)	0.9962(10)
	$\sigma[(s,s), \mu_0 \to \mu]$	0.9339(22)	0.8706(37)
	$Z[\mathrm{lat} \to \overline{\mathrm{MS}}, \mu]$	0.5727(15)	0.5866(26)
$\mu_0 = 1.4363$ GeV	$Z_{B_K}[(s,s), \mu_1]$	0.88271(13)	0.96773(37)
	$\sigma[(s,s), \mu_0 \to \mu]$	0.9573(21)	0.9103(31)
	$Z[\mathrm{lat} \to \overline{\mathrm{MS}}, \mu]$	0.5791(14)	0.5959(22)

Table 11: The complete renormalization factors for the coarse $24^3 \times 64$ lattice. Note that the $\sigma[(s,s), \mu_0 \to \mu]$ numbers with $\mu_0 = 1.1199$ GeV are re-measured within this work.

The $Z[\mathrm{lat} \to \overline{\mathrm{MS}}, \mu]$ number should be independent from the intermediate renormalization schemes and energy scales, apart from finite lattice spacing errors, finite number of loops included in the perturbative calculation of the matching factors and finite volume errors. The results shown in the table indicates that the NPR procedures give an overall systematic error of 3 percent to the final result.

4.2.2 The Matrix Element on Lattice B_K[lat]

The last piece in (146) that has not been taken care of is the lattice matrix element itself $B_K[\mathrm{lat}]$. Combining (145) and (147) the correlation functions are fitted as

$$\begin{aligned}
\langle 0|A_0|K^0\rangle &= \sum_{\boldsymbol{x}} \langle \bar{d}(\boldsymbol{x},0)\gamma_4\gamma_5 s(\boldsymbol{x},0)\bar{s}(t)\gamma_5 d(t)\rangle && (149)\\
&= \sum_{\boldsymbol{x}} \mathrm{tr}\left[\gamma_4 D^{-\dagger}(t \to \boldsymbol{x},0)\gamma_5 D^{-1}(t \to \boldsymbol{x},0)\right] && (150)\\
&\to \frac{Z_{K,\mathrm{wall}} f_K}{2Z_A}\left[\mathrm{e}^{-m_K t} - e^{-m_K(T-t)}\right] && (151)
\end{aligned}$$

$$\langle\overline{K^0}|\Big[\bar{s}\gamma_\mu(1-\gamma_5)d\Big]\Big[\bar{s}\gamma_\mu(1-\gamma_5)d\Big]|K^0\rangle \tag{152}$$

$$= \sum_{\boldsymbol{x}}\langle\Big[\bar{d}(t_2)\gamma_5 s(t_2)\Big]\Big[\bar{s}(\boldsymbol{x},t)\Gamma d(\boldsymbol{x},t)\bar{s}(\boldsymbol{x},t)\Gamma d(\boldsymbol{x},t)\Big]_{VV+AA}\Big[\bar{d}(t_1)\gamma_5 s(t_1)\Big]\rangle \tag{153}$$

$$= 2\cdot(\text{trace-trace term}) - 2\cdot(\text{trace term}) \tag{154}$$

$$\to -\frac{8}{3}\frac{B_K[\text{lat}]Z^2_{K,\text{wall}}f_K^2}{4Z_A^2 V}\Big[e^{-m_K(t_2-t_1)}\Big], \tag{155}$$

where the contractions are

$$(\text{trace-trace term}) = \sum_{\boldsymbol{x}}\operatorname{tr}\Big[S_s^\dagger(t_2\to\boldsymbol{x},t)\gamma_5\Gamma S_d(t_2\to\boldsymbol{x},t)\Big]\operatorname{tr}\Big[S_s^\dagger(t_1\to\boldsymbol{x},t)\gamma_5\Gamma S_d(t_1\to\boldsymbol{x},t)\Big], \tag{156}$$

$$(\text{trace term}) = \sum_{\boldsymbol{x}}\operatorname{tr}\Big[S_s^\dagger(t_1\to\boldsymbol{x},t)\gamma_5\Gamma S_d(t_2\to\boldsymbol{x},t)S_s^\dagger(t_2\to\boldsymbol{x},t)\gamma_5\Gamma S_d(t_1\to\boldsymbol{x},t)\Big]. \tag{157}$$

Here we have used the notation for the quark field on a fixed time slice

$$q(t) = \sum_{\boldsymbol{x}} q(\boldsymbol{x},t). \tag{158}$$

For the purpose of performing global chiral fit for B_K we give the result of B_K in $(\not{q},\not{q})$ of RI-SMOM, i.e. (146) but without the $R[(s,s)\to\overline{\text{MS}},\mu]$ factor, measured on the coarse $24^3\times 64$ lattice as

$$B_K[(\not{q},\not{q}),\mu = 3.0\text{ GeV}] = 0.6012(22). \tag{159}$$

The error on this number only includes the statistical error: it does not include the error introduced during the NPR procedures, as discussed in the previous chapter; in addition it does not include the finite lattice spacing error, finite volume error of the lattice and corrections from extrapolation to the physical point, which will be the issues addressed in the global chiral fit in the next section.

4.3 Chiral Fits of B_K

In addition to the global chiral fit for the low energy constants (LECs) for the low energy effective theory for QCD based on chiral perturbation theory there is a similar chiral fit for B_K as well. The fit form is [17]

$$B_{xy} = B_K^0 \left[1 + c^{\mathbf{A}}_{B_K,a^2} a^2 + \frac{c_{B_K,m_l}\chi_l}{f^2} + \frac{c_{B_K,m_x}\chi_x}{f^2} - \frac{\chi_l}{32\pi^2 f^2} \log\left(\frac{\chi_x}{\Lambda_\chi^2}\right)\right] + c_{B_K,m_y}(m_y - m_h^{\text{phys}}) + c_{B_K,m_h}(m_h - m_h^{\text{phys}}) + (\text{F.V. corr.}), \tag{160}$$

where the subscripts x,y mean valence light and heavy quark masses, l, h mean dynamic/sea light and heavy quark masses and the finite volume correction (F.V. corr.) terms can be found in [32]. LECs, including f, B (hiding in all the $\chi_\bullet$'s) and Λ_χ are needed from the global chiral fit. The term $c^{\mathbf{A}}_{B_K,a^2} a^2$ describes the finite lattice spacing effect on the lattices. Different coefficients are introduced for the different gauge actions: The superscript **A** is either **IW** for the lattices with Iwasaki gauge action, or **ID** for the lattices with Iwasaki gauge action plus the DSDR term; all other coefficients are common for the two gauge actions. The second line of (160) is the analytic fit form for the heavy quark around the physical heavy quark mass point. The physical prediction of B_K is obtained by evaluating (160) at the physical light and heavy quark masses.

Previously such B_K chiral fit only includes the **IW** lattices [17] due to the fact that there is only one DSDR lattice in the global chiral fit. Several other DSDR lattices, including the coarse $24^3 \times 64$ lattice, have since been generated and the renormalized $B_K[(\not{q}, \not{q}), \mu = 3.0 \text{ GeV}]$ as well as other basic quantities have been measured on three DSDR lattices. They are listed in table 14. On top of the results presented in [17] the global chiral fit and B_K chiral fit are rerun including these three new DSDR lattices.

In particular we are interested in the lattice spacing dependence of B_K on the DSDR lattice. The fit result is shown in table 15 and figure 22 and 21. The result does not indicate a significant a^4 term needed to describe the finite lattice spacing effect on B_K even on the coarse $24^3 \times 64$ lattice with lattice spacing of $a^{-1} \sim 1$ GeV.

4.3.1 Measurements on the New ID Lattices

Measurements of the basic physical quantities are needed for the global chiral fit and lattice regularized B_K[lat] is needed for B_K chiral fit. Here we summarize the results on the 24**ID**($\beta = 1.633$) and 32**ID**($\beta = 1.75$) physical lattices.

Observable	Fit
$am_{\text{res}}(m_l = 0.0001)$	0.0018915(75)
$am_{\text{res}}(m_h = 0.045)$	0.0017091(51)
am_π	0.10468(32)
am_K	0.35581(43)
am_Ω	1.2030(36)
am'_Ω	1.610(21)
af_π	0.09490(20)
af_K	0.11339(17)
Z_A	0.68779(11)
Z_V^π	0.68339(82)
Z_V^K	0.6911(25)
B_K	0.61632(91)

Table 12: Spectrum from 33 measurements (configuration 120 to 430 with increments of 10), with no binning, on the 32**ID**($\beta = 1.75$) physical lattice. For this ensemble eigenvectors of the Dirac operator have already been calculated for other measurements. They are used to speed up light quark propagator calculations, which are done to 10^{-8} precision on all time slices, i.e. without all Mode Averaging (AMA).

Observable	Fit
$am_{\text{res}}(m_l = 0.00107)$	0.0022824(70)
am_π	0.13975(10)
am_K	0.504154(89)
am_Ω	1.6726(25)
am'_Ω	2.040(63)
af_π	0.13055(11)
af_K	0.15815(13)
Z_A	0.73457(11)
Z_V^π	0.72672(35)
Z_V^K	0.7390(11)
B_K	0.68253(70)

Table 13: Spectrum from 378 measurements (configuration 300 to 1808 with increments of 4), binned over every 3 successive measurements, on the 24**ID**($\beta = 1.633$) physical lattice, measured by David Murphy. All Mode Averaging (AMA) is done with sloppy (10^{-4} CG precision) solves over all time slices and precise (10^{-8} CG precision) solves over 7 time slices from each configuration.

4.3.2 Global Chiral Fit Including the New ID Lattices

See table 14 and its caption for detailed setup and result of the global chiral fit including the latest measurement results for the coarse DSDR lattices.

m_π^{cut}		370 MeV
χ^2/dof		0.78(22)
a^{-1}	32**I**	2.378(10)
	24**I**	1.7798(66)
	48**I***	1.7301(29)
	64**I***	2.3607(78)
	32**I**-fine	3.142(21)
	32**ID**($\beta = 1.75$)*	1.3787(48)
	24**ID**($\beta = 1.633$)	1.0230(20)
	32**ID**-kaon($\beta = 1.633$)	1.0230(20)
	32**ID**($\beta = 1.633$)	1.0230(20)
	32**ID**($\beta = 1.943$)*	2.069(16)
f_π [GeV]		0.13002(79)
f_K [GeV]		0.15516(76)
$\sqrt{t_0}$[GeV]		0.7283(46)
w_0 [GeV]		0.8743(44)
$m_\pi a_0^2$		$-0.0409(16)$

Table 14: The global chiral fit results. The 48**I** and 64**I** are overweighted, i.e. their χ^2 weight 5000 times more than other lattices. For the 32**ID**($\beta = 1.75$) lattice only unitary data with $m_l = 0.0001$ and $m_h = 0.045$ is included. The lattices with $\beta = 1.633$ are restricted to have the same lattice spacing. The 32**ID**($\beta = 1.943$) lattice is still included in spite of its pion mass ($\sim$ 400 MeV) being over the 370 MeV cut.

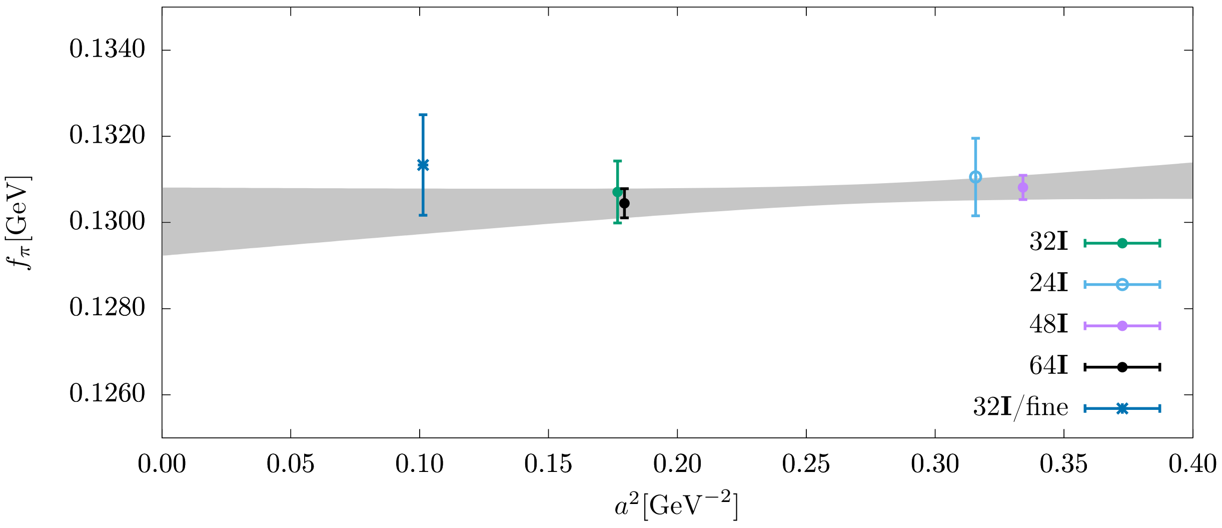

Figure 15: The a^2 dependence of f_π for the **IW** lattices.

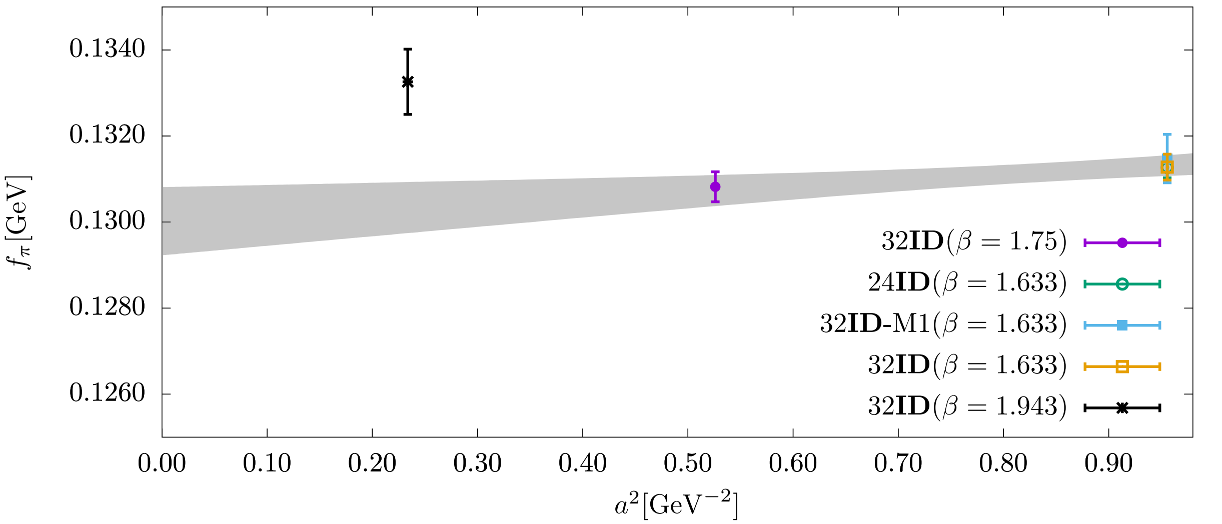

Figure 16: The a^2 dependence of f_π for the **ID** lattices.

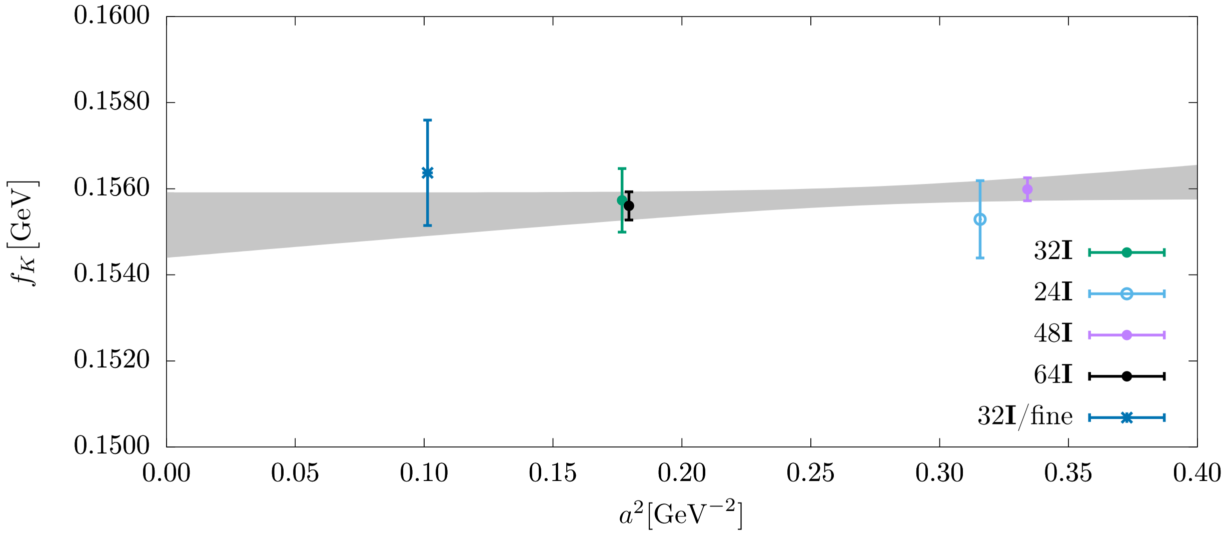

Figure 17: The a^2 dependence of f_K for the **IW** lattices.

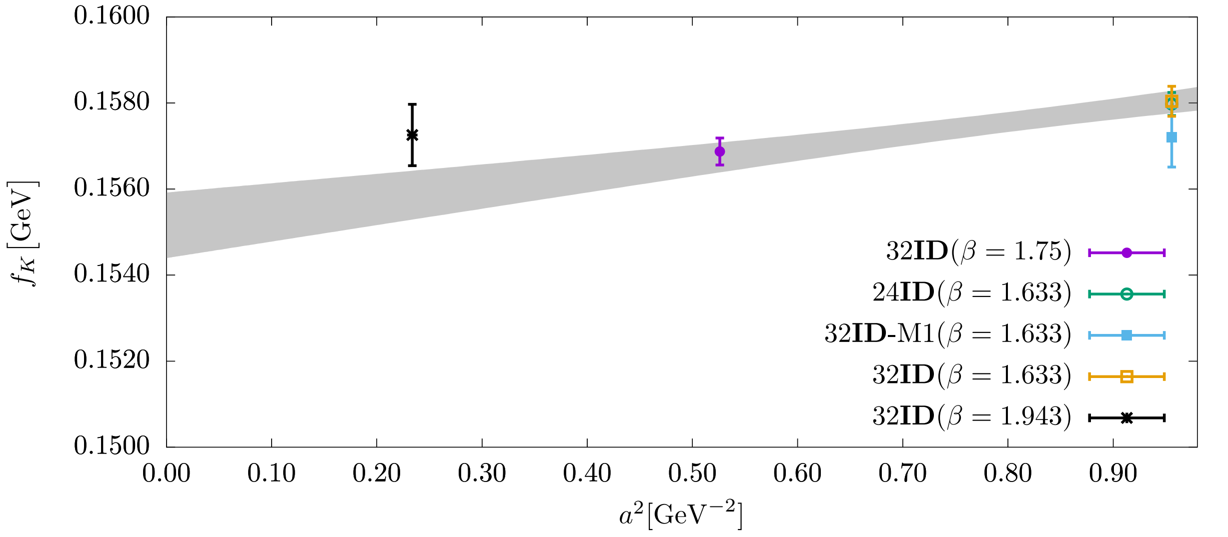

Figure 18: The a^2 dependence of f_K for the **ID** lattices.

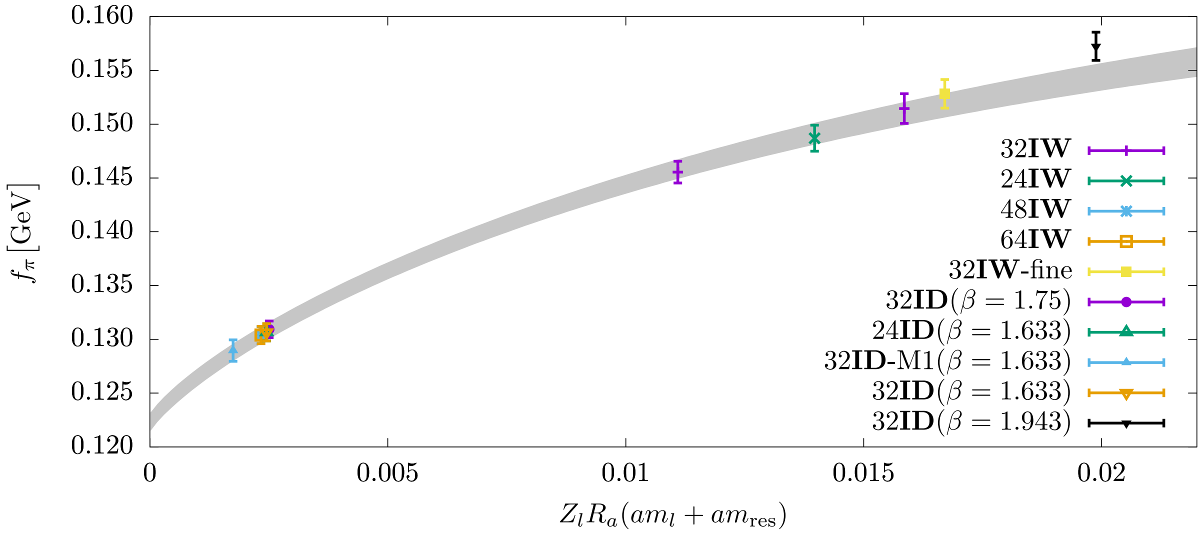

Figure 19: Light quark mass dependence of f_π.

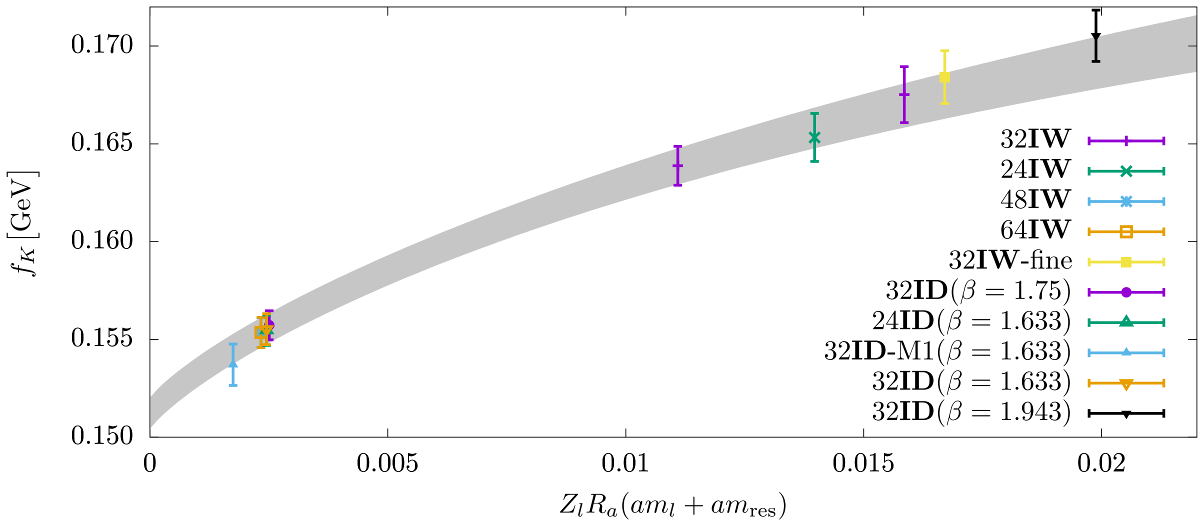

Figure 20: Light quark mass dependence of f_K.

4.3.3 B_K Chiral Fit Including the New ID Lattices

The B_K fit is performed based on the Global Chiral Fit with the 64**I** and 48**I** lattices overweighted. The result is shown in table 15. The much wanted a^2 dependence for the **IW** and **ID** lattices are shown in figure 21 and 22.

	ChPTFV	ChPTFV[17]
χ^2/dof	0.28(32)	–
$B_K^{\mathrm{phys}}[(\not{q},\not{q}),\mu = 3.0\ \mathrm{GeV}]$	0.5349(19)	0.5341(18)
B_K^0	0.5280(17)	0.5278(16)
$c^{\mathbf{IW}}_{B_K,a^2}[\mathrm{GeV}^2]$	0.115(11)	0.128(12)
$c^{\mathbf{ID}}_{B_K,a^2}[\mathrm{GeV}^2]$	0.1352(72)	0.153(15)
c_{B_K,m_l}	−0.0074(10)	−0.00728(95)
c_{B_K,m_x}	0.00449(67)	0.00420(64)
c_{B_K,m_h}	−0.16(18)	−0.06(18)
c_{B_K,m_y}	1.208(29)	1.324(32)

Table 15: The B_K chiral fit result. The “phys” superscript indicates extrapolation to the physical quark masses.

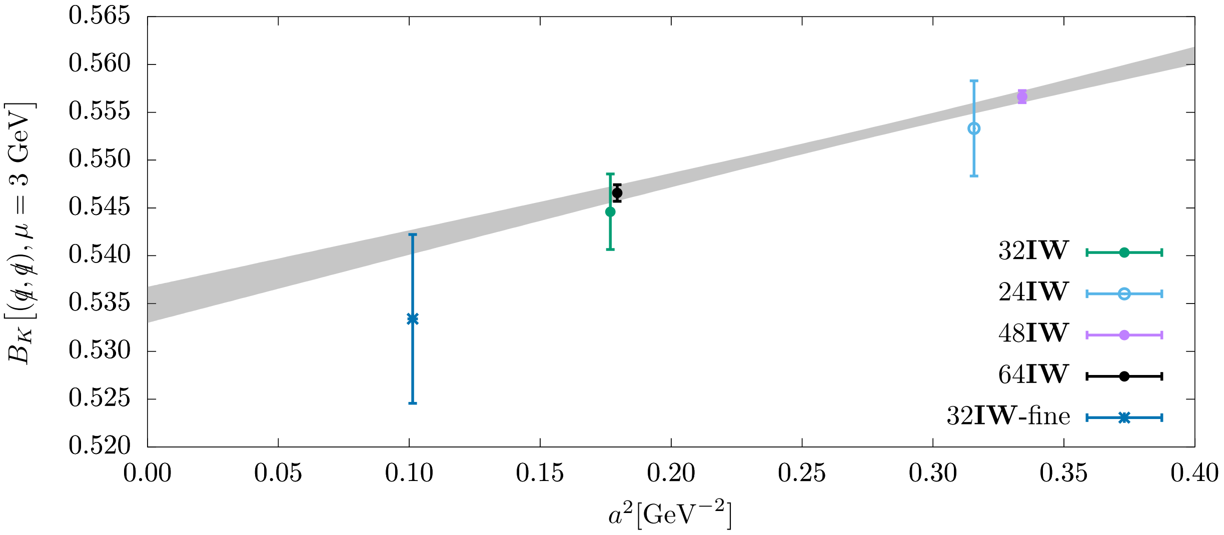

Figure 21: The a^2 dependence of B_K for the **IW** lattices.

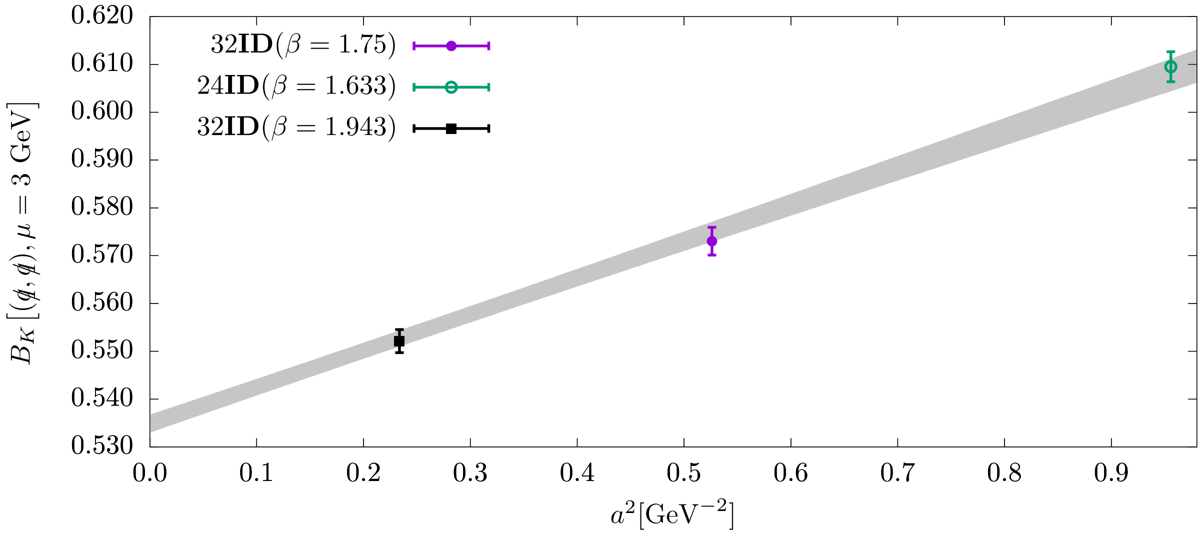

Figure 22: The a^2 dependence of B_K for the **ID** lattices.

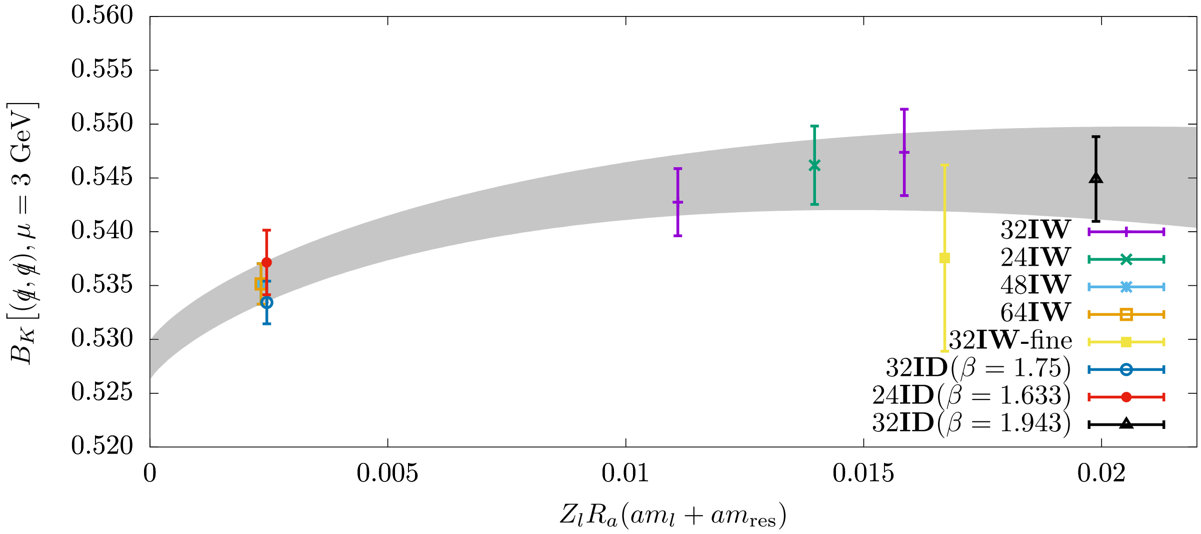

Figure 23: Light quark mass dependence of B_K.

4.4 $\Delta I = 3/2$ $K \to \pi\pi$ Decay Amplitude A_2

4.4.1 Introduction

The decay amplitude A_2 for a kaon to decay into two-pion state with isospin $I = 2$ plays an important role in understanding the kaon decay processes within the framework of the standard model. The fact that the state-of-art lattice QCD simulation is able to calculate these physical quantities continues to prove the power and capability of the lattice gauge theory. Further physical implications of A_2 are found in [33].

Here in this work we describe the method and results of the A_2 calculation on the 24**ID**($\beta = 1.633$) lattice. This calculation allows us to probe the lattice spacing a^2 dependence of the quantity at $a^{-1} \sim 1$ GeV scale, which is interesting in its own right, as well as in that it gives insight into the scaling behavior of the more complicated A_0 calculation.

4.4.2 The Wigner-Eckart Trick

The $\Delta I = 3/2$ $K \to \pi\pi$ decay amplitude A_2 is defined as

$$\sqrt{2}A_2 = \langle (\pi^0\pi^0)^{I=2}_{I_3=0} | H_w | K^0 \rangle, \tag{161}$$

where on lattice H_w is the weak Hamiltonian after the operator product expansion,

$$H_w = \frac{G_F}{\sqrt{2}} V^*_{ud} V_{us} \sum_i C_i(\mu) Q_i(\mu). \tag{162}$$

Here $C_i(\mu)$ are the Wilson coefficients and $Q_i(\mu)$ are the $\Delta I = 3/2, \Delta I_3 = 1/2$ operators, both renormalized under $\overline{\text{MS}}$ at μ. Explicitly the operators are

$$Q_{(27,1)} = (\bar{s}_\alpha d_\alpha)_{V-A}(\bar{u}_\beta u_\beta - \bar{d}_\beta d_\beta)_{V-A} + (\bar{s}_\alpha u_\alpha)_{V-A}(\bar{u}_\beta d_\beta)_{V-A}, \tag{163}$$

$$Q_{(8,8),\text{unmix}} = (\bar{s}_\alpha d_\alpha)_{V-A}(\bar{u}_\beta u_\beta - \bar{d}_\beta d_\beta)_{V+A} + (\bar{s}_\alpha u_\alpha)_{V-A}(\bar{u}_\beta d_\beta)_{V+A}, \tag{164}$$

$$Q_{(8,8),\text{mix}} = (\bar{s}_\alpha d_\beta)_{V-A}(\bar{u}_\alpha u_\beta - \bar{d}_\beta d_\alpha)_{V+A} + (\bar{s}_\alpha u_\beta)_{V-A}(\bar{u}_\beta d_\alpha)_{V+A}. \tag{165}$$

The subscripts $(\bullet, \bullet)$ are the corresponding representation of the operator under $\text{SU}(3)_L \times \text{SU}(3)_R$ in the chiral limit. From this point on we will focus on evaluating the right hand side of (161), specifically the renormalized matrix elements

$$M_i^{\overline{\text{MS}}}(\mu) = Z_{ij}(\mu) \langle \pi^0\pi^0 | Q_j | K^0 \rangle = Z_{ij}(\mu) M_j^{\text{lat}}. \tag{166}$$

These three operators, however, make it difficult to construct and extract the on shell $\pi\pi$ state. Physically $m_K > 2m_\pi$ so on shell the final $\pi\pi$ state will have a zero center of mass momentum but each pion will possess a non-zero momentum. On the lattice these excited states are difficult to construct, extract or even identify without changing the ground state of the $\pi\pi$ state. In [34] two lattice boundary conditions are proposed to change the $\pi\pi$ ground state on the lattice: the **G**-parity boundary condition and the **H**-parity boundary condition. The later one, which is used in this work, is able to change the ground state of the $|\pi^+\pi^+\rangle$ (and $|\pi^-\pi^-\rangle$) without generating new lattice configurations. We will discuss this in the this section.

Still our final $\pi\pi$ state is $|\pi^0\pi^0\rangle$ instead of $|\pi^+\pi^+\rangle$. The Wigner-Eckart theorem is used to change the final state to the desired one. The $|\pi^0\pi^0\rangle$ state has $I_3 = 0$ and $|\pi^+\pi^+\rangle$ state has $I_3 = 2$. Applying the Wigner-Eckart theorem gives

$$\langle(\pi^0\pi^0)^{I=2}_{I_3=0}|Q^{\Delta I=3/2}_{\Delta I_3=+1/2}|(K^0)^{I=1/2}_{I_3=-1/2}\rangle = \frac{\langle\frac{1}{2},-\frac{1}{2};\frac{3}{2},\frac{1}{2}|2,0\rangle}{\langle\frac{1}{2},\frac{1}{2};\frac{3}{2},\frac{3}{2}|2,2\rangle}\langle(\pi^+\pi^+)^{I=2}_{I_3=2}|Q^{\Delta I=3/2}_{\Delta I_3=3/2}|(K^+)^{I=1/2}_{I_3=1/2}\rangle. \tag{167}$$

Thus

$$\langle\pi^0\pi^0|Q^{\Delta I=3/2}_{\Delta I_3=+1/2}|K^0\rangle = \frac{1}{\sqrt{2}}\langle\pi^+\pi^+|Q^{\Delta I=3/2}_{\Delta I_3=+3/2}|K^+\rangle. \tag{168}$$

The $Q^{\Delta I=3/2}_{\Delta I_3=+3/2}$ operators are obtained by applying raising operator of isospin to the $Q^{\Delta I=3/2}_{\Delta I_3=+1/2}$ operators in (169)-(171),

$$Q^{\Delta I_3=3/2}_{(27,1)} = (\bar{s}_\alpha d_\alpha)_{V-A}(\bar{u}_\beta d_\beta)_{V-A}, \tag{169}$$

$$Q^{\Delta I_3=3/2}_{(8,8),\text{unmix}} = (\bar{s}_\alpha d_\alpha)_{V-A}(\bar{u}_\beta d_\beta)_{V+A}, \tag{170}$$

$$Q^{\Delta I_3=3/2}_{(8,8),\text{mix}} = (\bar{s}_\alpha d_\beta)_{V-A}(\bar{u}_\beta d_\alpha)_{V+A}. \tag{171}$$

In addition the raising operator adds a normalization factor of $\sqrt{3}$ to each of the operators. This process is physically forbidden from the standard model by EM charge conservation and the matrix element only exists on the lattice.

The decay amplitude is now

$$\sqrt{2}A_2 = \frac{G_F}{\sqrt{2}}V_{ud}V^*_{us}\cdot\frac{\sqrt{3}}{\sqrt{2}}\sum_{ij}C_i(\mu)Z_{ij}(\mu)M^{\text{lat}}_j, \tag{172}$$

where

$$M^{\text{lat}}_j = \langle\pi^+\pi^+|Q^{\Delta I=3/2}_j|K^+\rangle. \tag{173}$$

4.4.3 Extraction of $\pi\pi$ State

The **H**-parity boundary condition [34] applies anti-periodic boundary condition on the down quark in one or more of the spatial directions and keeps periodic boundary condition for other directions and other quarks:

$$(u, d)(\boldsymbol{x}_i + L) = (u, -d)(\boldsymbol{x}_i). \tag{174}$$

With this boundary condition the π^+ state has an anti-periodic boundary in these spatial directions,

$$|\pi^+\rangle(\boldsymbol{x}_i + L) = -|\pi^+\rangle(\boldsymbol{x}_i), \tag{175}$$

which raises the ground state energy of π^+ to

$$E_\pi(n_{tw}) = \sqrt{m_\pi^2 + \left(n_{tw}\frac{\pi}{L}\right)^2}, \tag{176}$$

where n_{tw} is the number of spatial direction with anti-periodic boundary condition, or the number of *twists*. In other words a momentum of π/L is inserted into the π^+ for each twist applied. Pions with more energy leads to $\pi^+\pi^+$ states with more energy, whose ground state is now a $\pi^+\pi^+$ state with zero center of mass momentum but the two pion each possess $+\pi/L$ or $-\pi/L$ of relative momentum in the directions where **H**-parity is applied. For later convenience this momentum is label as $\boldsymbol{p}$.

The benefit of adopting **H**-parity boundary condition is that it does not require generating new gauge configurations. This is because **H**-parity is equivalent to applying the following phase transformation

$$\psi(x) \to e^{i\boldsymbol{p}\cdot\boldsymbol{x}}\psi(x). \tag{177}$$

A fermion field redefinition in (20) keeps the action unchanged while the measurement function, or the integrand, of the path integral is changed to

$$\psi(x_1)\bar{\psi}(x_2) \to \psi(x_1)\bar{\psi}(x_2)e^{-i\boldsymbol{p}\cdot\boldsymbol{x}_1}e^{+i\boldsymbol{p}\cdot\boldsymbol{x}_2}. \tag{178}$$

As a result the gauge field configuration can be reused while the measurement forms acquire additional phase factors. Since **H**-parity is applied on the down quark its propagator is calculated with *cosine-wall* source,

$$S_d(\boldsymbol{x}, t_1 \leftarrow t_2; \boldsymbol{p}) = \cos(\boldsymbol{p}_1\boldsymbol{y}_1)\cos(\boldsymbol{p}_2\boldsymbol{y}_2)\cos(\boldsymbol{p}_3\boldsymbol{y}_3) S_d(\boldsymbol{x}, t_1 \leftarrow \boldsymbol{y}, t_2). \tag{179}$$

The use of cosine, instead of the $e^{\pm i\boldsymbol{p}_1\boldsymbol{y}_1}$ form, averages over $e^{+i\boldsymbol{p}_1\boldsymbol{y}_1}$ and $e^{-i\boldsymbol{p}_1\boldsymbol{y}_1}$ in one propagator and reduces the number of expensive inversion needed to be performed [33]. At the location of the sink additional phase factors are needed before contractions are summed over. The $\pi^+\pi^+$ to $\pi^+\pi^+$ correlator, for example, is constructed as

$$C_{\pi\pi}(t_1, t_2) = \langle [\bar{d}\gamma_5 u \bar{d}\gamma_5 u](t_2)[\bar{u}\gamma_5 d \bar{u}\gamma_5 d](t_1)\rangle = 2 \cdot (\text{trace-trace term}) - 2 \cdot (\text{trace term}). \tag{180}$$

$$(\text{trace-trace term}) = \sum_{\boldsymbol{x},\boldsymbol{y}} \mathrm{tr}\left[S_u(t_2 \to t_1) S_d^\dagger(\boldsymbol{x}, t_1 \leftarrow t_2; \boldsymbol{p}) e^{+i\boldsymbol{p}\cdot\boldsymbol{x}}\right] \cdot \mathrm{tr}\left[S_u(t_2 \to t_1) S_d^\dagger(\boldsymbol{y}, t_1 \leftarrow t_2; \boldsymbol{p}) e^{-i\boldsymbol{p}\cdot\boldsymbol{y}}\right], \tag{181}$$

$$(\text{trace term}) = \sum_{\boldsymbol{x},\boldsymbol{y}} \mathrm{tr}\left[S_u(t_2 \to t_1) S_d^\dagger(\boldsymbol{x}, t_1 \leftarrow t_2; \boldsymbol{p}) e^{+i\boldsymbol{p}\cdot\boldsymbol{x}} \cdot S_u(t_2 \to t_1) S_d^\dagger(\boldsymbol{y}, t_1 \leftarrow t_2; \boldsymbol{p}) e^{-i\boldsymbol{p}\cdot\boldsymbol{y}}\right], \tag{182}$$

Compared to the usual correlator with periodic boundary condition in all spatial directions it bears with it a factor of $1/2^{n_{tw}}$:

$$\cos(\boldsymbol{p}_1\boldsymbol{y}_1)\cos(\boldsymbol{p}_2\boldsymbol{y}_2)\cos(\boldsymbol{p}_3\boldsymbol{y}_3)\cos(\boldsymbol{p}_1\boldsymbol{x}_1)\cos(\boldsymbol{p}_2\boldsymbol{x}_2)\cos(\boldsymbol{p}_3\boldsymbol{x}_3) = \tag{183}$$

$$\frac{1}{2^{2n_{tw}}} \sum_{\pm} e^{\pm i\boldsymbol{p}_1(\boldsymbol{x}_1 - \boldsymbol{y}_1)} e^{\pm i\boldsymbol{p}_2(\boldsymbol{x}_2 - \boldsymbol{y}_2)} e^{\pm i\boldsymbol{p}_3(\boldsymbol{x}_3 - \boldsymbol{y}_3)} + \cdots \tag{184}$$

In other words the cosine-wall source already sums over terms with all possible combinations of $\pm\pi/L$ in each twisted spatial direction with total momentum of zero (e.g. the summation over ±). There are $2^{n_{tw}}$ terms of this kind. The terms with non-zero momentum are annihilated at the sinks with the above phase choice. The factor of $1/2^{2n_{tw}}$ in front of the summation sign comes from the 2's in $\cos x = (e^{ix} + e^{-ix})/2$. Therefore the overall factor is

$$\frac{1}{2^{2n_{tw}}} \cdot 2^{n_{tw}} = \frac{1}{2^{n_{tw}}}. \tag{185}$$

This numerical factor will be useful for the fitting form of the $K \to \pi\pi$ decay amplitude.

It is possible to twist an arbitrary number of spatial directions but in practice the number of twists are chosen to make the ground state energy of the $\pi\pi$ state close to the kaon state thus making the measured matrix elements measured close to being on shell.

4.4.4 Measurements of the Matrix Elements

One of the major difficulties of measuring the $K \to \pi\pi$ decay matrix element in the continuum and infinite volume limit is how to extract it from Euclidean correlation functions measured on a lattice with finite volume: there is no apparent or simple relationship between these two for the desired $\pi\pi$ final state. For a single particle final state it is shown that the finite volume effect vanishes exponentially with $e^{-m_\pi L}$, which is the reason why this problem does not draw much tension when measuring the basic quantities or B_K. In [35] the needed relationship is derived as

$$|A|^2 = 8\pi \left[q\frac{\partial\phi}{\partial q} + k\frac{\partial\delta_0}{\partial k} \right]_{k=k_\pi} \left(\frac{m_K E_{\pi\pi}^2}{k_\pi^3} \right) |M|^2. \tag{186}$$

Here A is the decay amplitude in infinite volume and M is the matrix element measured on a finite volume lattice. δ_0 is the $\pi\pi$ s-wave scattering phase shift and k_π is the center of mass pion momentum for the $\pi\pi$ we are calculating,

$$k_\pi = \frac{1}{2}\sqrt{E_{\pi\pi}^2 - 4m_\pi^2}, \tag{187}$$

$$n\pi - \delta_0(k) = \phi(q), \;\; q = \frac{kL}{2\pi}, \tag{188}$$

where $\phi(q)$ is a mathematical function defined as

$$\tan\phi(q) = -\frac{\pi^{3/2} q}{\mathcal{Z}_{00}^{\mathbf{d}}(1;q^2)}, \;\; \phi(0) = 0 \tag{189}$$

and the function $\mathcal{Z}_{00}^{\mathbf{d}}(1;q^2)$ is to be calculated according to (A8) of [36] with $\mathbf{d}$ indicates the twisted spatial directions[14]. On the other hand in order to evaluate $\partial\delta_0/\partial k$ a cubic ansatz is used to

[14] Technically speaking in that formula $\mathbf{d}$ indicates the total momentum of the $\pi\pi$ state, which is not the case for us: our $\pi\pi$ final state always has zero total momentum. We, however, has an anti-periodic boundary condition in some of the directions, which requires to sum over the modes that satisfies it in (A1) of [36] (e.g. $\mathbf{d} = (1/2, 1/2, 0)$ indicates the x and y directions are twisted) and with $\hat{\gamma} = 1$. Therefore effectively (A8) is used in our case with a different interpretation for $\mathbf{d}$.

parametrize $\delta_0(k)$,

$$\delta_0(k) = a_0 k + a_1 k^3. \tag{190}$$

To summarize, in order to calculate the factor in (186) the following steps are taken: for each spatial boundary condition,

1. k_π is calculated based on the corresponding $E_{\pi\pi}$ with (187).

2. The phase shift $\delta_0(k = k_\pi)$ and q is calculated with k_π using (188).

3. The $q\partial\phi/\partial q$ term in (186) is calculated.

4. a_0 and a_1 are fitted with all available k's and their corresponding $\delta_0(k)$'s. With a_0 and a_1 the $k\partial\delta_0/\partial k$ term in (186) is calculated.

This numerical factor is usually referred to as the Lüscher-Lellouch (LL) factor

$$\sqrt{\mathrm{LL}} = \sqrt{8\pi \left[q\frac{\partial\phi}{\partial q} + k\frac{\partial\delta_0}{\partial k} \right]_{k=k_\pi} \left(\frac{m_K E_{\pi\pi}^2}{k_\pi^3} \right)}. \tag{191}$$

4.5 Fitting Strategy

The kaon correlator is constructed and fitted as

$$C_K(t_1, t_2) = \langle [\bar{u}\gamma_5 s](t_2)[\bar{s}\gamma_5 u](t_1) \rangle \to |Z_K|^2 \Big(e^{-m_K \Delta t} + e^{-m_K (T-\Delta t)} \Big),\ \Delta t = t_2 - t_1. \tag{192}$$

The $\pi\pi$ correlator is constructed in (180) and is fitted as

$$C_{\pi\pi}(t_2, t_1) \to \frac{1}{2^{n_{tw}}} |Z_{\pi\pi}|^2 \Big[e^{-E_{\pi\pi}\Delta t} + e^{-E_{\pi\pi}(T-\Delta t)} + C \Big]. \tag{193}$$

The constant C in the fitting form accounts for all the around world effects and the additional factor $\frac{1}{2^{n_{tw}}}$ comes from the source we use for the down quark. An new *ratio* method is used in order to reduce the noise in the extraction of the $\pi\pi$ state. The $\pi\pi$ correlator is combined with the π correlator to form the following ratio:

$$R_{\pi\pi}(t) \equiv \frac{\partial_t C_{\pi\pi}}{\partial_t C_\pi^2} = \frac{Z_{\pi\pi}^2 E_{\pi\pi}}{2 Z_\pi^4 m_\pi} e^{-\Delta_{\pi\pi} T/2} \left[\cosh(\Delta_{\pi\pi}\hat{t}) + \sinh(\Delta_{\pi\pi}\hat{t}) \coth(2m_\pi \hat{t}) \right], \tag{194}$$

where $\hat{t} = T/2 - t$.

The $K \to \pi\pi$ correlator is constructed as the following. For a four-quark operator with spin structure Γ_1 and Γ_2 the contractions are calculated as

$$O[\Gamma_1, \Gamma_2; \text{diag}](t) = \sum_{\boldsymbol{x}} [\bar{s}_\alpha(\boldsymbol{x}, t)\Gamma_1 d_\alpha(\boldsymbol{x}, t)][\bar{u}_\beta(\boldsymbol{x}, t)\Gamma_2 d_\beta(\boldsymbol{x}, t)] \tag{195}$$

$$O[\Gamma_1, \Gamma_2; \text{mix}](t) = \sum_{\boldsymbol{x}} [\bar{s}_\alpha(\boldsymbol{x}, t)\Gamma_1 d_\beta(\boldsymbol{x}, t)][\bar{u}_\alpha(\boldsymbol{x}, t)\Gamma_2 d_\beta(\boldsymbol{x}, t)] \tag{196}$$

The matrix element

$$C_{\Gamma_1,\Gamma_2;\text{diag}} \equiv \langle K(t_1)|O[\Gamma_1, \Gamma_2; \text{diag}](t)|\pi\pi(t_2)\rangle \tag{197}$$

$$= \sum_{\boldsymbol{x}} \langle 0|\bar{u}(t_1)\gamma_5 s(t_1)[\bar{s}_\alpha(\boldsymbol{x}, t)\Gamma_1 d_\alpha(\boldsymbol{x}, t)][\bar{u}_\beta(\boldsymbol{x}, t)\Gamma_2 d_\beta(\boldsymbol{x}, t)]\bar{d}(t_2)\gamma_5 u(t_2)\bar{d}(t_2)\gamma_5 u(t_2)|0\rangle \tag{198}$$

$$= \sum_{\boldsymbol{x}} \text{tr}_{sc}\,[p_1\Gamma_1]\,\text{tr}_{sc}\,[p_2\Gamma_2] - \text{tr}_{sc}\,[p_1\Gamma_1 p_2\Gamma_2], \tag{199}$$

$$C_{\Gamma_1,\Gamma_2;\text{mix}} \equiv \langle K(t_1)|O[\Gamma_1, \Gamma_2; \text{mix}](t)|\pi\pi(t_2)\rangle \tag{200}$$

$$= \sum_{\boldsymbol{x}} \langle 0|\bar{u}(t_1)\gamma_5 s(t_1)[\bar{s}_\alpha(\boldsymbol{x}, t)\Gamma_1 d_\beta(\boldsymbol{x}, t)][\bar{u}_\beta(\boldsymbol{x}, t)\Gamma_2 d_\alpha(\boldsymbol{x}, t)]\bar{d}(t_2)\gamma_5 u(t_2)\bar{d}(t_2)\gamma_5 u(t_2)|0\rangle \tag{201}$$

$$= \sum_{\boldsymbol{x}} \text{tr}_c\,[\,\text{tr}_s\,[p_1\Gamma_1]\,\text{tr}_s\,[p_2\Gamma_2]] - \text{tr}_s\,[\,\text{tr}_c\,[p_1\Gamma_1]\,\text{tr}_c\,[p_2\Gamma_2]]. \tag{202}$$

where

$$p_1 = S_d(\boldsymbol{x}, t \leftarrow t_2)\gamma_5 S_u(t_2 \leftarrow t_1)\gamma_5 S_s(t_1 \leftarrow \boldsymbol{x}, t), \tag{203}$$

$$p_2 = S_d(\boldsymbol{x}, t \leftarrow t_2)\gamma_5 S_u(t_2 \leftarrow \boldsymbol{x}, t). \tag{204}$$

For the three operators in (169)-(171) the correlators are calculated with plugging in the corresponding Γ_1 and Γ_2. The fitting form is[15]

$$C_{K\to\pi\pi,j}(t_1, t, t_2) \to \frac{1}{2^{n_{tw}}}|Z_K||Z_{\pi\pi}|M_j^{\text{lat}} e^{-m_K(t_1-t)} e^{-E_{\pi\pi}(t-t_2)}. \tag{205}$$

[15] Compared to the fitting for B_K the volume factor V does not appear here since in [35] the states are normalized to unity instead of $\langle n|n\rangle = \frac{1}{2E_n V}$.

With all the numerical factors introduced in here the physical $\Delta I = 2$, $K \to \pi\pi$ amplitude A_2 is

$$\sqrt{2}a^3 A_2 = \frac{G_F}{\sqrt{2}} V_{ud} V_{us}^* \cdot \underbrace{\sqrt{\frac{3}{2}}}_{\Delta I_3 = 3/2} \cdot \underbrace{\left[\frac{1}{\pi q^{3/2}} \sqrt{q \frac{\partial \phi}{\partial q} + k \frac{\partial \delta}{\partial k}} \sqrt{m_K} E_{\pi\pi} L^{3/2} \right]_J}_{\sqrt{\mathrm{LL}}} \cdot \sum_{ij} C_i(\mu) Z_{ij}(\mu) \left[M_j^{\mathrm{lat}} \right]_J, \quad (206)$$

where J is jackknife index and i, j are operator indices. $C_i(\mu)$ are the Wilson coefficients. a, A_2 and G_F have units of GeV^{-1}, GeV and GeV^{-2}, respectively, and all other quantities are dimentionless.

4.5.1 Non-perturbative Renormalization

The structure of the $K \to \pi\pi$ operators is very similar to the operators in the BSM $\Delta S = 2$ operators. If we follow the same NPR procedures and carries out the same calculations in (129) the contractions of the vertices are exactly the same except without the factors of 2's,

$$2 \cdot \begin{pmatrix} \Lambda[Q_{(27,1)}^{\Delta I_3 = 3/2}] \\ \Lambda[Q_{(8,8),\mathrm{unmix}}] \\ \Lambda[Q_{(8,8),\mathrm{mix}}] \end{pmatrix}_{K \to \pi\pi} \overset{\text{parity even}}{=} \begin{pmatrix} \Lambda[VV + AA] \\ \Lambda[VV - AA]_{\mathrm{unmix}} \\ \Lambda[VV - AA]_{\mathrm{mix}} \end{pmatrix}_{\Delta S = 2} . \quad (207)$$

The $\Lambda[VV - AA]_{\mathrm{mix}}$ term can be rewritten using the Fiertz identity,

$$\begin{pmatrix} [VV + AA] \\ [VV - AA]_{\mathrm{unmix}} \\ [VV - AA]_{\mathrm{mix}} \end{pmatrix}_{\Delta S = 2} = \begin{pmatrix} Q_1 \equiv [VV + AA] \\ Q_2 \equiv [VV - AA]_{\mathrm{unmix}} \\ -2Q_3 \equiv -2[SS - PP]_{\mathrm{unmix}} \end{pmatrix}_{\Delta S = 2} . \quad (208)$$

Therefore the renormalization conversion factors previously calculated for B_K can be reused with the following modification:

$$Z^{K \to \pi\pi} = \begin{pmatrix} Z_{11}^{\Delta S = 2} & & \\ & Z_{22}^{\Delta S = 2} & -\frac{1}{2} Z_{23}^{\Delta S = 2} \\ & -2 Z_{32}^{\Delta S = 2} & Z_{33}^{\Delta S = 2} \end{pmatrix} . \quad (209)$$

Note that in this case there is no Z_A^2 factor in the denominator.

4.5.2 $\pi\pi$ Results

We show the results for $\pi\pi$ part of the measurements, where results from both the ratio method and the more conventional direct fitting are included.

Observable	Fit	error %
am_π	0.13993(35)	0.25
am_K	0.50425(49)	0.10
$\rightarrow n_{tw} = 3$		
$aE_{\pi\pi}^{I=2}[R]$	0.5634(40)	0.72
$aE_{\pi\pi}^{I=2}[C]$	0.5511(54)	0.98
$a\Delta_{\pi\pi}^{I=2}$	0.2836(40)	1.39
$\delta_0^{I=2}(p)$	$-14.5(19)$	12.94
$\sqrt{\text{LL}}$	30.00(13)	0.42
$\rightarrow n_{tw} = 2$		
$aE_{\pi\pi}^{I=2}[R]$	0.4768(17)	0.36
$aE_{\pi\pi}^{I=2}[C]$	0.475(11)	2.25
$a\Delta_{\pi\pi}^{I=2}$	0.1969(16)	0.82
$\delta_0^{I=2}(p)$	$-8.6(11)$	12.75
$\sqrt{\text{LL}}$	37.91(13)	0.33
$\rightarrow n_{tw} = 0$		
$aE_{\pi\pi}^{I=2}[R]$	0.28221(70)	0.25
$aE_{\pi\pi}^{I=2}[C]$	0.2809(47)	1.68
$a\Delta_{\pi\pi}^{I=2}$	0.002351(91)	3.87
$\delta_0^{I=2}(p)$	$-0.363(21)$	5.65
$\sqrt{\text{LL}}$	45.1(48)	10.71

Table 16: Spectrum from 58 AMA measurements [1000..2140..20], with no binning. The LL factor is calculated by fitting $\delta_0^{\Delta I=2}(p) = a_1 p + a_3 p^3$.

Both the ratio (R) and the conventional method (C) are used to extract the $\pi\pi$ state.

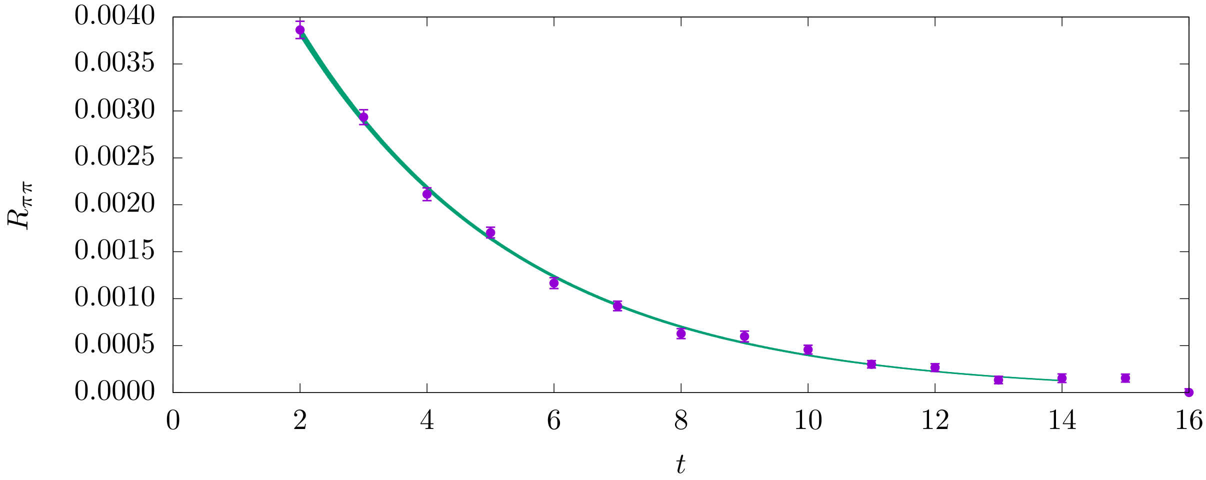

Figure 24: Effective mass plot for $R_{\pi\pi}$ with $n_{tw} = 3$.

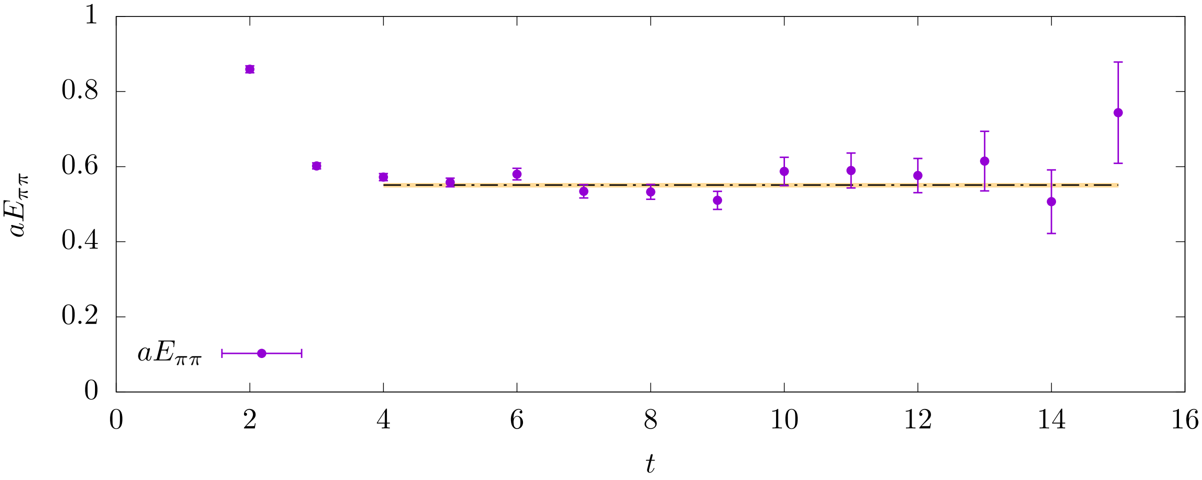

Figure 25: Effective mass plot for $E_{\pi\pi}$ with $n_{tw} = 3$.

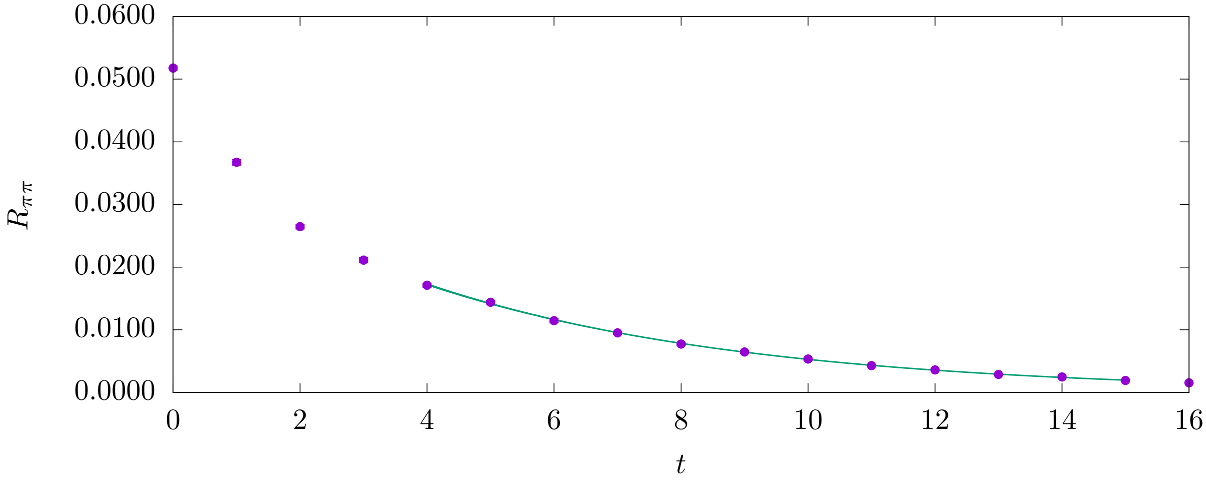

Figure 26: Effective mass plot for $R_{\pi\pi}$ with $n_{tw} = 2$.

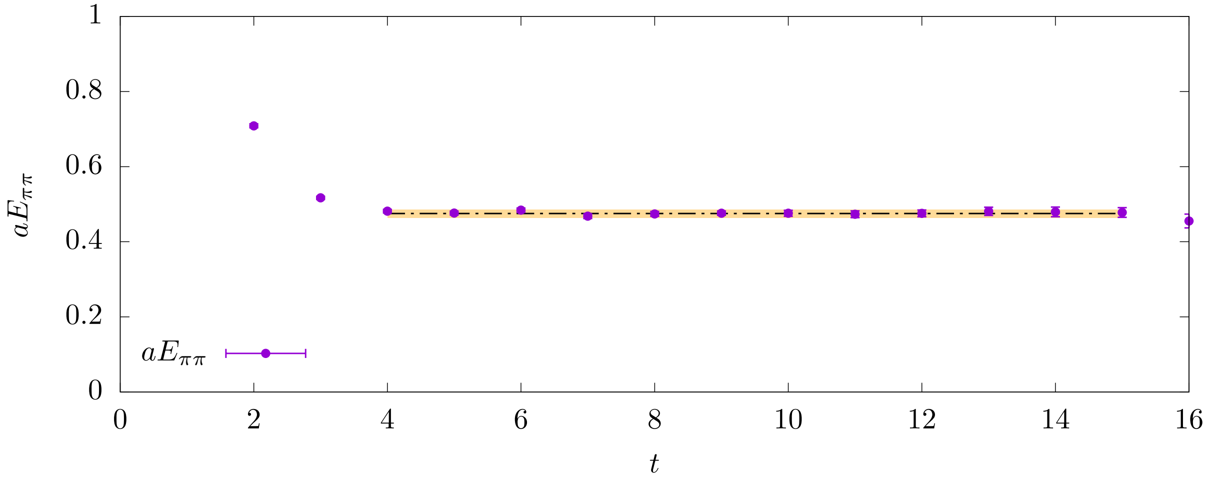

Figure 27: Effective mass plot for $E_{\pi\pi}$ with $n_{tw} = 2$.

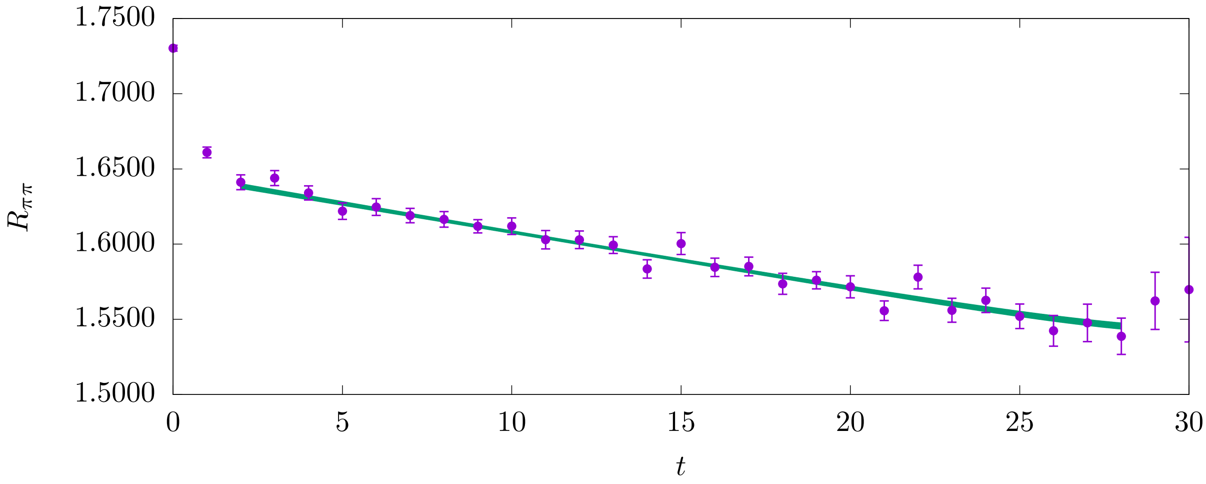

Figure 28: Effective mass plot for $R_{\pi\pi}$ with $n_{tw} = 0$.

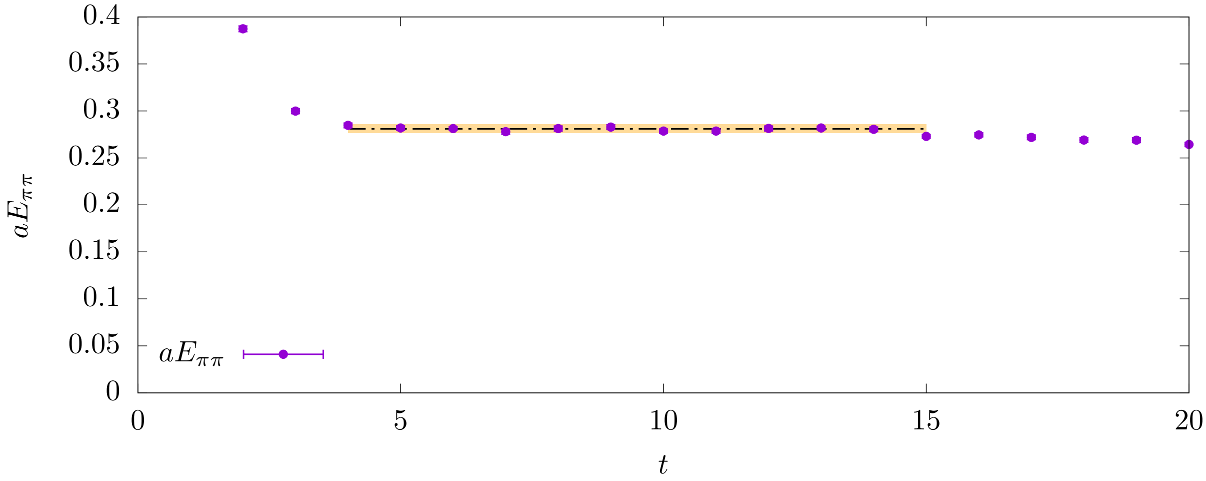

Figure 29: Effective mass plot for $E_{\pi\pi}$ with $n_{tw} = 0$.

4.5.3 $K \to \pi\pi$ Results

In table 17 we show the results from measuring the relevant matrix elements for A_2 with different number of twists and the final result for A_2 is shown in table 18. Since the $E_{\pi\pi}$ from $n_{tw} = 2$ and $n_{tw} = 3$ are both different from m_K we interpolate A_2 with physical kinematics (where $m_K = E_{\pi\pi}$) with an linear anzatz in $E_{\pi\pi}^2$ in figure 30. Our final result for A_2 on the 24**ID**($\beta = 1.633$) is

$$\text{Re}[A_2] = 1.508(8)_{\text{stat.}}(50)_{\text{NPR}} \times 10^{-8} \text{ GeV}, \;\; [24\textbf{ID}(\beta = 1.633), \text{this work}] \tag{210}$$

$$\mathrm{Im}[A_2] = -5.77(13)_{\mathrm{stat.}}(43)_{\mathrm{NPR}} \times 10^{-13}\ \mathrm{GeV}.\ [24\mathbf{ID}(\beta = 1.633),\ \text{this work}] \tag{211}$$

Again for NPR the central value is taken from the $(\not{q}, \not{q})$ scheme and the difference between $(\not{q}, \not{q})$ and (γ_μ, γ_μ) is taken as the systematic error from our NPR procedure.

Observable M^{lat}	Fit	error %
$\to n_{tw} = 3$		
$(27, 1)$	0.000924(20)	2.15
$(8, 8)$ mixed	0.00944(20)	2.13
$(8, 8)$ unmixed	0.03257(69)	2.12
$\to n_{tw} = 2$		
$(27, 1)$	0.0008595(49)	0.56
$(8, 8)$ mixed	0.011594(58)	0.50
$(8, 8)$ unmixed	0.03909(20)	0.51
$\to n_{tw} = 0$		
$(27, 1)$	0.0007372(20)	0.27
$(8, 8)$ mixed	0.025405(81)	0.32
$(8, 8)$ unmixed	0.08211(26)	0.32

Table 17: Spectrum from 58 AMA measurements [1000..2140..20], with no binning.

n_{tw}	am_K	$aE_{\pi\pi}^{I=2}$	$\mathrm{Re}[A_2][10^{-8}\ \mathrm{GeV}]$	$\mathrm{Im}[A_2][10^{-13}\ \mathrm{GeV}]$
3	0.50425(49)	0.5634(40)	$1.7125(68)_{\mathrm{stat.}}(575)_{\mathrm{NPR}}$	$-5.27(15)_{\mathrm{stat.}}(41)_{\mathrm{NPR}}$
2	0.50425(49)	0.4768(17)	$1.4206(57)_{\mathrm{stat.}}(476)_{\mathrm{NPR}}$	$-5.98(16)_{\mathrm{stat.}}(45)_{\mathrm{NPR}}$
0	0.50425(49)	0.28221(70)	$0.7132(39)_{\mathrm{stat.}}(233)_{\mathrm{NPR}}$	$-8.32(20)_{\mathrm{stat.}}(58)_{\mathrm{NPR}}$
$*$	0.50425(49)	am_K	$1.5079(80)_{\mathrm{stat.}}(505)_{\mathrm{NPR}}$	$-5.77(13)_{\mathrm{stat.}}(43)_{\mathrm{NPR}}$

Table 18: NPR is done in $\overline{\mathrm{MS}}$, $(\not{q}, \not{q})$ scheme and $\mu = 3$ GeV. $a^{-1} = 1.0083$ GeV. The NPR error is taken as the difference between $(\not{q}, \not{q})$ and (γ_μ, γ_μ) scheme. $*$ is the result from linear extrapolation in $E_{\pi\pi}^2$ to physical kinematics.

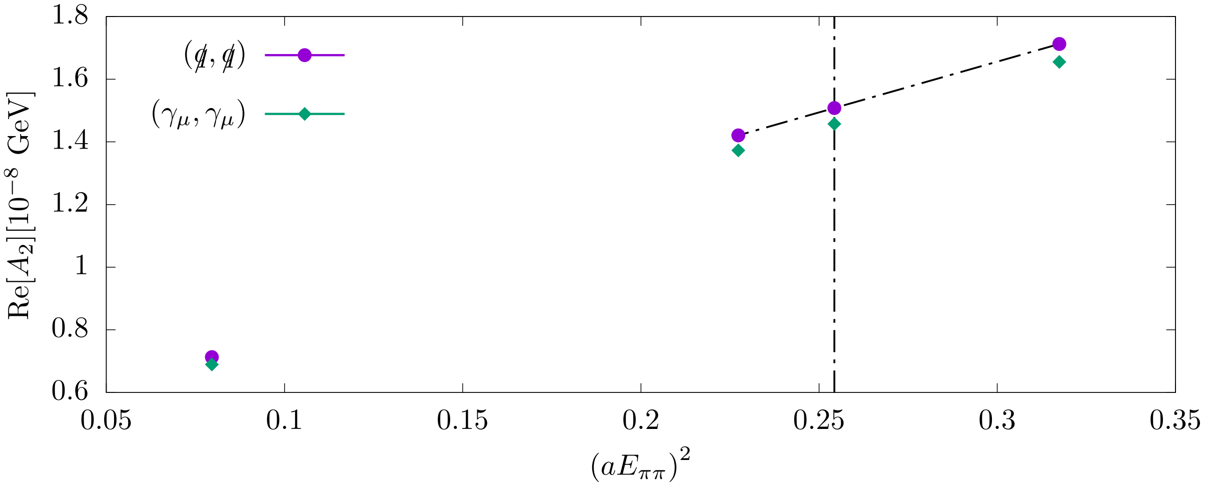

Figure 30: Interpolation to physical kinematics.

In addition the a^2 dependence of A_2 is plotted in figure 31 and 32. The continuum limit value is quoted from [33] where A_2 is measured from a continuum extrapolation between the 48**I** and 64**I** lattices:

$$\mathrm{Re}[A_2] = 1.50(4)_{\mathrm{stat.}}(14)_{\mathrm{NPR}} \times 10^{-8}\ \mathrm{GeV},\ \text{[continuum limit]} \tag{212}$$

$$\mathrm{Im}[A_2] = -6.99(20)_{\mathrm{stat.}}(84)_{\mathrm{NPR}} \times 10^{-13}\ \mathrm{GeV}.\ \text{[continuum limit]} \tag{213}$$

The 32**ID**($\beta = 1.75$) value is quoted from [37]:

$$\mathrm{Re}[A_2] = 1.381(46)_{\mathrm{stat.}}(258)_{\mathrm{syst.}} \times 10^{-8}\ \mathrm{GeV},\ [32\mathbf{ID}(\beta = 1.75),\text{ no interp. to physical kinematics}] \tag{214}$$

$$\mathrm{Im}[A_2] = -6.54(46)_{\mathrm{stat.}}(120)_{\mathrm{syst.}} \times 10^{-13}\ \mathrm{GeV}.\ [32\mathbf{ID}(\beta = 1.75),\text{ no interp. to physical kinematics}] \tag{215}$$

These values, however, have not been interpolated to the physical kinematics: the error due to unphysical kinematics is included in the systematic (syst.) error. With the results measured on the 24**ID**($\beta = 1.633$), in this work, these quoted 32**ID**($\beta = 1.75$) results are corrected as follows:

$$\Delta[32\mathbf{ID}] = \Delta[24\mathbf{ID}] \cdot \frac{(m_K^2 - E_{\pi\pi}^2)[32\mathbf{ID}]}{(m_K^2 - E_{\pi\pi}^2)[24\mathbf{ID}]}, \tag{216}$$

where Δ is the difference between the A_2 measured with $n_{tw} = 2$ and the A_2 that has been interpolated to physical kinematics,

$$\Delta[\mathrm{lat}] = A_2[n_{tw} = 2, \mathrm{lat}] - A_2[\text{physical kinematics}, \mathrm{lat}],\ \ \mathrm{lat} = 24\mathbf{ID}/32\mathbf{ID}. \tag{217}$$

$$\mathrm{Re}[A_2] = 1.445(46)_{\mathrm{stat.}}(258)_{\mathrm{syst.}} \times 10^{-8}\ \mathrm{GeV},\ \ [32\mathbf{ID}(\beta = 1.75),\ \text{corrected}] \tag{218}$$

$$\mathrm{Im}[A_2] = -6.39(46)_{\mathrm{stat.}}(120)_{\mathrm{syst.}} \times 10^{-13}\ \mathrm{GeV}.\ [32\mathbf{ID}(\beta = 1.75),\ \text{corrected}] \tag{219}$$

The error bars shown in figure 31 and 32 only include statistical errors (does not include systematic errors).

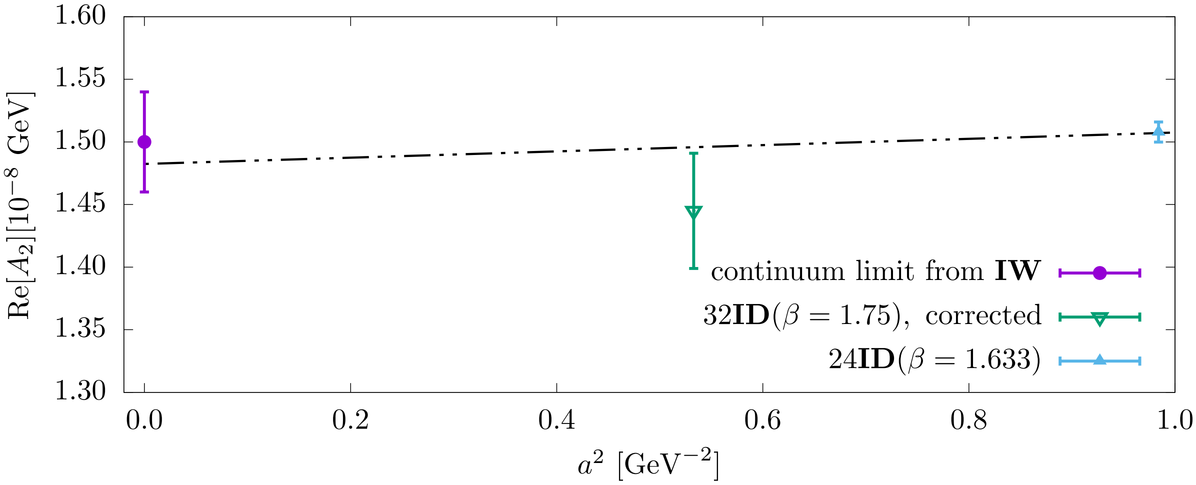

Figure 31: a^2 dependence of $\mathrm{Re}[A_2]$. The error bars shown are statistical only.

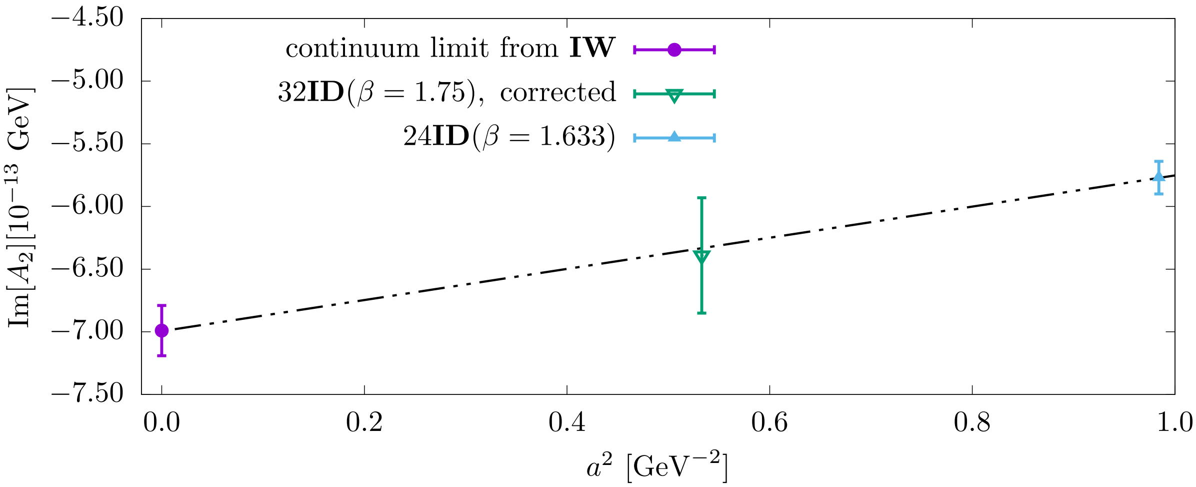

Figure 32: a^2 dependence of $\mathrm{Im}[A_2]$. The error bars shown are statistical only.

4.6 Conclusion

Based on the a^2 dependence plots of B_K and A_2, figure 22, figure 31 and figure 32, even only considering the statistical errors, neither B_K nor A_2 show the presence of a significant a^4 term in their continuum limit extrapolation: given the level of statistical error for these two quantities shown in this work, the simple analytic form of (109) is still enough to capture the their behavior running towards the $a = 0$ limit.

This linearity of the a^2 dependence (with the absence of a large a^4 dependence) of B_K and A_2 on the 24**ID**($\beta = 1.633$) and the 32**ID**($\beta = 1.75$) lattices, on top of the similar behaviors shown in the basic meson spectrums, makes these coarse **ID** lattices ideal for making cheap measurements of low energy physics while having finite lattice spacing errors under control to give continuum predictions.

Specifically for the $\Delta I = 1/2$ $K \to \pi\pi$ amplitude A_0 and the corresponding ϵ'/ϵ, this linearity makes it reasonable to perform a linear continuum limit extrapolation in a^2 with results measured on as few as two **ID** lattices with different lattice spacings, which greatly reduces the amount of computational resources needed.

5 The Multisplitting Preconditioned Conjugate Gradient Algorithm

5.1 Overview of the Problem

The cost of lattice QCD simulations with dynamical fermions is dominated by the solution of the Dirac equation in both the evolution phase and the measurement phase. The Dirac matrix, which is the gauge field dependent discretization of the fermion part of the continuous QCD action, is a large sparse linear system and inverting the corresponding Dirac equation poses tremendous numerical difficulty. For domain wall fermions the conjugate gradient (CG) algorithm proves to be a stable algorithm to solve the Dirac equation but the convergence rate is limited by the condition number of the Dirac matrix, which is typically large in simulations with physical pion mass.

For the measurement phase various eigen-space methods, including EigCG [38] and the implicitly restarted Lanczos algorithm with Chebyshev polynomial preconditioning [39], have been successfully developed to speed up the inversion. Low-lying eigenvectors (eigenvectors corresponding to small eigenvalues) of the Dirac matrix are generated and the previously large condition number is effectively reduced to improve the convergence rate of CG. In this phase for one gauge field configuration typically a large number of Dirac equations with the same Dirac matrix but different right hand sides (RHS, or sources) are solved. The large number of sources amortizes the cost of eigenvector generation and the total computation time is reduced.

This is, however, not the case for the evolution phase. During a typical hybrid Monte Carlo evolution of a gauge field as few as one Dirac equation is solved for a single Dirac matrix. This renders it not worthwhile to generate the low-lying eigenvectors for a particular Dirac matrix.

The development of supercomputers has greatly increased the number of floating point operations per second (flops) that can be performed on each processor. Modern lattice simulations usually divide the gauge field and pseudo-fermion fields into sub-fields that are stored and computed locally on different processors of a large parallel computer. This increases the total theoretical floating point operation capability. Inter-processor data transfer (communication), however, is needed to perform coherent operations, including the Dirac matrix multiplication. Computations locally performed on one processor require contents of the sub-fields that are stored and updated on other processors. For a specific operation if the rate of communication can not keep up with the local flops then communication becomes the bottleneck and the high flops are not utilized.

For the standard CG solver with DWF one Dirac matrix multiplication is performed for each iteration. The precise requirement varies with the size of the lattice and processor grid, but roughly this requires one byte of communication for each local floating point operation. On some of the newest machines, for example the SUMMIT machine at Oak Ridge National Laboratory (ORNL), inter-processor communication speed is much less than the requirement set by their high local floating point operation capability.

In [40] a domain decomposition algorithm is proposed for solving the Dirac equation with Wilson fermions. Local inversions are performed on two halves of the lattice iteratively. However, attempts to apply the same or similar algorithms to the inversion of the DWF Dirac equation have not been successful.

In this work we report on our investigation into a preconditioned CG solver for solving the DWF Dirac equation for the ensemble generation phase of the simulation. We find a preconditioner that decreases the number of CG iterations needed for a solution, while increasing the local computation required per iteration, thus changing the balance of local computation to off-processor communication.

5.2 The Conjugate Gradient Algorithm

The conjugate gradient (CG) algorithm is a Krylov space iterative method to solve linear equations $Ax = b$, where A is a hermitian and positive definite matrix. The solution vector is updated in every iteration to minimize the residual in the newly expanded Krylov space. Here we show the original algorithm first. The convergence rate of the CG algorithm largely depends on the conditioned number κ, which is the ratio between the largest eigenvalue and smallest eigenvalue, of the underlying matrix of interest. A widely quoted result[16] on the upper bound of the residual after k iterations is

$$\|b - Ax_k\|_A \leq 2r^{-k}\|b - Ax_0\|_A, \tag{220}$$

[16] For example see [41].

Algorithm 1 Conjugate Gradient $Ax = b$

```
$r_0 = b - Ax_0$
$p_0 = r_0$
$k = 0$
**while** have not converged **do**
    $\alpha_k = \langle r_k, r_k \rangle / \langle p_k, Ap_k \rangle$
    $x_{k+1} = x_k + \alpha_k p_k$
    $r_{k+1} = r_k - \alpha_k A p_k$
    $\beta_k = \langle r_{k+1}, r_{k+1} \rangle / \langle r_k, r_k \rangle$
    $p_{k+1} = r_{k+1} + \beta_k p_k$
    $k = k + 1$
**end while**
```

where $\|x\|_A^2 = \langle x, Ax \rangle$ and r is (lower bound of) the convergence rate,

$$r = \frac{\sqrt{\kappa(A)} + 1}{\sqrt{\kappa(A)} - 1} \overset{\kappa(A) \gg 1}{\simeq} 1 + \frac{2}{\sqrt{\kappa(A)}}. \tag{221}$$

In practice this lower bound is usually too pessimistic and is only conceptually useful. The spectrum structure in the middle of the eigenvalue range does affect the actual convergence rate [41]. Very few theoretical studies, however, have been done on this topic.

5.2.1 The Preconditioned Conjugate Gradient Algorithm

One way to reduce the condition number and increase the convergence rate is to *precondition* the original matrix A with a *preconditioner* A_{pre}. The idea is to find a preconditioning matrix (preconditioner) A_{pre} that is easy to invert while making the combination AA_{pre}^{-1} (or $A_{\text{pre}}^{-1}A$) have a smaller condition number than A's.[17] If the preconditioner A_{pre} is also hermitian and positive definite we have the following preconditioned conjugate gradient algorithm.

[17] The lattice QCD community uses the word *precondition* to describe a number of different things. A nonexclusive list includes the general preconditioning to reduce the condition number and the even-odd preconditioning, where a lattice operator is decomposed according to sites with even or odd coordinates (see section 1.2.2 for further details).

Algorithm 2 Preconditioned Conjugate Gradient $Ax = b$

```
$r_0 = b - Ax_0$
$z_0 = M^{-1} r_0$
$p_0 = z_0$
$k = 0$
while have not converged do
    $\alpha_k = \langle r_k, z_k \rangle / \langle p_k, Ap_k \rangle$
    $x_{k+1} = x_k + \alpha_k p_k$
    $r_{k+1} = r_k - \alpha_k A p_k$
    $z_{k+1} = A_{\mathrm{pre}}^{-1} r_{k+1}$
    $\beta_k = \langle z_{k+1}, r_{k+1} \rangle / \langle z_k, r_k \rangle$
    $p_{k+1} = z_{k+1} + \beta_k p_k$
    $k = k + 1$
end while
```

For later convenience and consistency the preconditioning inversion step $z_{k+1} = A_{\mathrm{pre}}^{-1} r_{k+1}$ will be referred to as the *inner inversion*, as compared to the outer inversion. The iterations of the preconditioned CG will be called *outer iterations*. If the inner inversion is performed also with an iterative method, say, an unpreconditioned CG, its iterations are to be called *inner iterations*.

Note that the above preconditioned CG algorithm requires the preconditioner to be fixed. In practice this is usually a requirement too expensive to satisfy. On one hand for inversions performed on computers, depending on the underlying floating point number representation, finite floating point precision compromises exact arithmetics. On the other hand, the application of A_{pre}^{-1} is usually itself an inversion problem. Requiring this inner inversion to be performed to very high precision, if not infinite precision, amortizes the possible speed up gained from preconditioning the linear equation solving in the first place.

An inexact preconditioned conjugate gradient method with inner-outer iterations is proposed with its convergence proved in [42]. The inner inversion is performed using CG to a certain precision while the convergence is still maintained. The algorithm stays unchanged except the calculation of β_k, which is marked with blue in algorithm 2, is changed to

$$\beta_k = \langle z_{k+1}, r_{k+1} - r_k \rangle / \langle z_k, r_k \rangle. \tag{222}$$

In the literature the above formula for β_k is often referred as the Polak-Ribière version while the original $\beta_k = \langle z_{k+1}, r_{k+1}\rangle / \langle z_k, r_k\rangle$ is referred as the Fletcher-Reeves version.

5.2.2 Mixed Precision Solver and Residual Replacement Strategies

When solving the Dirac equation we want the solution vector in higher precision while lower precision arithmetics gives higher computation flops and memory traffic bandwidth. A combination of these two, whose idea is to perform most of the iterations using lower precision while make higher precision corrections at the *right* moments, gives precise the solution more quickly.

It is well known that even in higher precision the *accumulated residual*, accumulated through steps in red in algorithm 2, differs from the *true residual* $r_n = b - Ax_n$ significantly due to the floating point rounding errors occur during the accumulations. Mixed precisions makes things worse: the deviation of accumulated residual from the true residual in addition receives contribution from the lower precision arithmetic. Over the course of the iterations corrections with higher precision arithmetic is needed. This correction is referred to as a reliable update, defect correlation or residual replacement in the literature.

The non-trivial aspect of this is *when* to make the correlation. Making corrections with higher precision more frequently than needed amortizes the speed up gained from using lower precision while making corrections less frequently than needed compromises the convergence due to the inexact nature of the lower precision.

Naive conditions include performing the correction whenever the residual is reduced by a constant factor from the last corrected residual or the initial residual, or simply always performing correction after a fixed number of lower precision iterations have been executed. A more sophisticated approach, however, is to dynamically estimate the accumulated floating point error during the lower precision iterations and perform the higher precision correlation whenever this estimated error is larger than a constant threshold [43]. For (preconditioned) CG the floating point error accumulated d_n during the lower precision iterations can be simplified as

$$d_k = d_{k-1} + \epsilon_l \left(\|A\| \cdot \|x_k\| + \|r_k\| \right) \tag{223}$$

and a higher precision correlation is made if $d_k > \sqrt{\epsilon_l}\|r_k\|$.

5.3 The Numerical Situation

5.3.1 Hardware Specifications of SUMMIT

The SUMMIT machine at Oak Ridge National Laboratory (ORNL) is a pre-exascale supercomputer with 200 peta-flops of peak double precision computation capability. Its flops are concentrated on the $27,648$ NVIDIA Tesla V100-SXM2-16GB graphic processor units (GPUs) distributed on $4,608$ compute nodes of the machine (6 GPUs per node).[18] Specifications of each of the Tesla V100 GPUs that will be useful in this thesis are listed in table 19.

Double Precision Performance	7.8 tera-flops
Single Precision Performance	15.7 tera-flops
Half Precision Tensor Core Performance	125 tera-flops
(Global) Memory Bandwidth	900 GB/sec

Table 19: Relevant specifications of the Tesla V100 GPU.

Because MDWF requires more flops to solve, due to its fifth dimension, the pseudo-fermion fields are usually spread out to more processors, shrinking the local 4d volume, increasing the surface area to volume ratio on each processor and requiring more inter-processor communications.

5.3.2 Target Lattice Generation

We first summarize the parameters of the 96**I** lattice whose configurations we are currently generating on SUMMIT in table 20. It is designed to be a 2+1 flavor lattice with Iwasaki gauge action, $a^{-1} \sim 2.8$ GeV and physical quark mass, which will serve as a solid data point on the finite lattice spacing a^2 extrapolation curve together with the 48**I** and 64**I** lattices with lattice spacings of $a^{-1} \sim 1.73$ GeV and $a^{-1} \sim 2.36$ GeV. The larger volume puts the finite volume error under control.

[18] See https://www.olcf.ornl.gov/for-users/system-user-guides/summit/summit-user-guide/.

size	$96^3 \times 128 \times 12$
β	2.31
am_l	0.00054
am_h	0.02132
Möbius scale α	2.0

Table 20: Running parameters of the target lattice generation on SUMMIT.

The evolution time is dominated by the Dirac matrix solving for the light and heavy quark determinants, which accounts for over 90% of the total time. The detailed evolution timing, including the strategies and optimizations described in this work, will be given in the result section.

5.4 The Theory

5.4.1 The Multisplitting Algorithm

In [44] a *multisplitting* algorithm is proposed for solving generic large linear systems distributed across a parallel computer. Compared to the domain decomposition algorithm in [40], it does not require checkerboarding. Before each iteration the boundary content of the solution field on each of the processors is communicated to its neighbors. During each iteration, the algorithm uses this communicated neighboring solution field as the Dirichlet boundary condition to perform the inversion of a local matrix on each processor. After each iteration, the updated boundary content is again communicated to prepare for the next iteration.

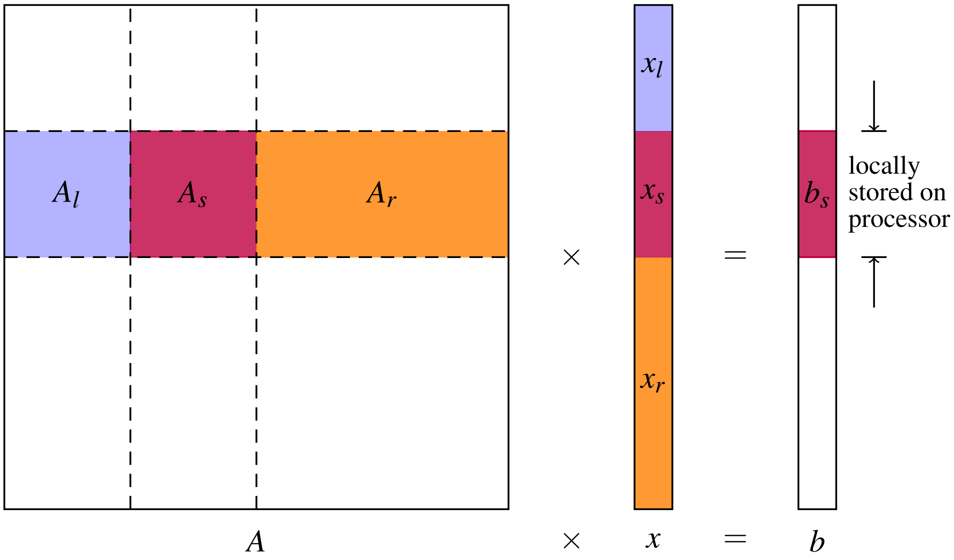

Figure 33: Decomposition of the matrix A, the solution vector x and the right-hand-side(RHS) vector b into local parts on each processor.

Following [45], suppose the equation to be solved is $Ax = b$. For a *particular* processor the matrix A and vectors x and b are decomposed according to figure 33, where x_s and b_s are the part that is locally stored on this processor. On each processor the original equation turns into

$$A_s x_s + A_l x_l + A_r x_r = b_s. \tag{224}$$

The $A_l x_l + A_r x_r$ part involves off-processor content and is calculated before each iteration via communication. A_s is the part of the matrix that requires only the locally stored part of x on a certain processor s, i.e. x_s. Then for each iteration the algorithm solves the equation

$$A_s x_s = b_s - A_l x_l - A_r x_r \tag{225}$$

locally for x_s on this processor. The updated solution x_s will then be communicated to the neighboring processors. This whole procedure can be done concurrently on all processors once the communication work to calculate $A_l x_l + A_r x_r$ is done.

5.4.2 Further Domain Wall Fermion Simulation Strategies

Modern numerical implementations of DWF utilize the fact that only the matrix elements that connect the *even* sites to *odd* sites and those connecting *odd* sites to *even* sites (M_{eo} and M_{eo} in equation (29)) depend the gauge field. The matrix entries that connect *even* sites to *even* sites and those connect *odd* sites to *odd* sites (M_{ee} and M_{oo} in equation (29)) are constant, which makes them easy to invert explicitly. Here the even-odd parity is defined by the 4 dimensional coordinate of a site:

$$\text{parity} \equiv (x+y+z+t) \mod 2. \tag{226}$$

In the 4 dimensional even-odd preconditioning form the Möbius DWF Dirac equation can be written as,

$$\begin{pmatrix} M_{ee} & M_{eo} \\ M_{oe} & M_{oo} \end{pmatrix} \begin{pmatrix} \psi_e \\ \psi_o \end{pmatrix} = \begin{pmatrix} M_{ee} & M_{eo} \\ 0 & M_{oo} \end{pmatrix} \begin{pmatrix} 1 - M_{ee}^{-1} M_{eo} M_{oo}^{-1} M_{oe} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ M_{oo}^{-1} M_{oe} & 1 \end{pmatrix} \begin{pmatrix} \psi_e \\ \psi_o \end{pmatrix} = \begin{pmatrix} \phi_e \\ \phi_o \end{pmatrix}. \tag{227}$$

This is equivalent to solving the following even-odd preconditioned Dirac equation,

$$D_{prec}\psi_e = \hat{\phi}_e,\ D_{prec} \equiv 1 - M_{ee}^{-1} M_{eo} M_{oo}^{-1} M_{oe},\ \hat{\phi}_e \equiv M_{ee}^{-1}(\phi_e - M_{eo} M_{oo}^{-1} \phi_o) \tag{228}$$

and with ψ_e the odd parity solution ψ_o can be trivially obtained by

$$\psi_o = M_{oo}^{-1}(\phi_o - M_{oe}\psi_e). \tag{229}$$

Here $M_{eo/oe}$ includes the Wilson hopping term $D_{w[x,y]}$ that connects sites to their nearest 4 dimensional space-time neighbors,

$$M_{oe/eo} = -\kappa_b D_{w[x,y]} M_\phi,\ D_{w[x,y]} \equiv \sum_\mu \left[(1+\gamma_\mu) U^\dagger_{x-\hat{\mu},\mu} \delta_{x-\hat{\mu},y} + (1-\gamma_\mu) U^\dagger_{x,\mu} \delta_{x+\hat{\mu},y} \right]. \tag{230}$$

The CG algorithm requires the matrix to be hermitian and positive definite. A common practice to satisfy this requirement is to multiply both sides of (228) with $D^\dagger_{prec}$ and solve the equation with the normal operator $D^\dagger_{prec}D_{prec}$ and the new RHS $D^\dagger_{prec}\hat{\phi}_e$ instead,

$$D^\dagger_{prec}D_{prec}\psi_e = D^\dagger_{prec}\hat{\phi}_e. \tag{231}$$

5.4.3 Dirichlet Boundary Condition on the 4-Hop Normal Operator

There are four Wilson hopping terms, one in each $M_{eo/oe}$, in the normal operator $D^\dagger_{prec}D_{prec}$, which is the underlying matrix A to be solved,

$$A = D^\dagger_{prec}D_{prec} = [1 - M_{ee}^{-1}{\color{red}M_{eo}}M_{oo}^{-1}{\color{red}M_{oe}}]^\dagger[1 - M_{ee}^{-1}{\color{red}M_{eo}}M_{oo}^{-1}{\color{red}M_{oe}}]. \tag{232}$$

To apply the multisplitting algorithm to equation (231) Dirichlet boundary conditions are to be enforced on the normal operator $D^\dagger_{prec}D_{prec}$, i.e. local parts (the A_s in (224)) of this normal operator needs to be constructed. As the vector content is distributed across the processors according to its 4 dimensional space-time location, the local parts of $D^\dagger_{prec}D_{prec}$ includes *snake* terms that hop out of the boundary and hop back in as the various components in (232) are evaluated. Figure 34 illustrates this and gives some examples of the snake terms. These terms are truncated if Dirichlet boundary conditions are enforced on each of the four $M_{eo/oe}$ hopping terms sequentially. Our simulation results show that the inclusion of these snake terms is crucial to the convergence when applying the multisplitting algorithm to solve the Dirac equation.

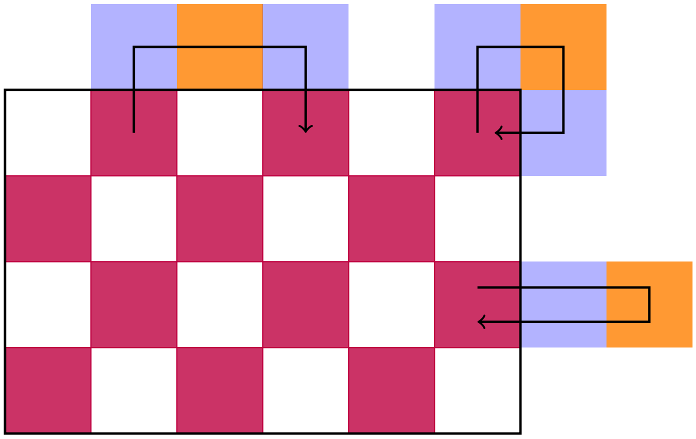

Figure 34: The normal operator $D^\dagger_{prec}D_{prec}$ has as many as 4 Wilson hopping terms. Enforcing Dirichlet boundary condition on it requires the inclusion of the *snake* terms, e.g. the black arrows.

The importance of these snake terms can be understood from another prospective. Applying the local part of the normal operator (A_s of equation (225)) without truncations to local parts of the vector x on each processor (x_s of equation (225)) is equivalent to applying the full normal operator A to the extended vector $\hat{x}_s$ with the same local content with x_s on *this* processor s but zero everywhere else, and then extracting the local part, i.e. ($[\bullet]_s$ means extracting the local part on s)

$$A_s x_s = [A\hat{x}_s]_s,\ \hat{x}_s = \begin{cases} x_s, & \text{on processor } s, \\ 0, & \text{on processors other than } s. \end{cases} \tag{233}$$

Thus the local operations share the same eigenspace structure with the normal operator. For example if A has a eigenspace structure of $\{\lambda, v_\lambda\}$

$$[A\hat{x}_s]_s = A_s x_s = \lambda \langle v_\lambda, x_s \rangle [v_\lambda]_s \tag{234}$$

The low lying components of the local operation are suppressed in the same way as the full operation with small λ's. If, however, the snake terms are truncated this equivalence of eigenspace structure is broken and the convergence is jeopardized.

5.4.4 Multisplitting Algorithm as a Preconditioner of CG

In [40] to achieve faster convergence the domain decomposition algorithm is eventually used as a preconditioner of the generalized conjugate residual (GCR) method. In this work we use the multisplitting algorithm as a preconditioner of CG. While paying the price of performing one inner inversion, which does not need communication, for each outer iteration, the outer iteration count is reduced by the preconditioning. Given that communication is much more costly than local memory bandwidth and computation under the current numerical situation such trade off is able to bring an overall speed up.

Now for the preconditioning step in the preconditioned CG (gray part in algorithm 2) we use the multisplitting algorithm to solve for z_{k+1} in

$$Az_{k+1} = r_{k+1}. \tag{235}$$

To avoid inter-processor communication, a zero initial guess ($x_l = x_r = 0$) is used in (225) and only the first iteration is performed. With r_{k+1} as the RHS and z_{k+1} the solution,

$$A_s x_s = b_s - A_l x_l - A_r x_r \rightarrow A_s z_{k+1,s} = r_{k+1,s}. \tag{236}$$

This is equivalent to using the local part of the matrix A, A_s, on each processor as the preconditioner A_{pre} in the preconditioned CG,

$$A_{\text{pre}} = \bigoplus_s A_s,\ \ s = \text{processor index.} \tag{237}$$

The local nature of A_s makes it possible to perform the preconditioning step concurrently on all the processors without communication. We refer to this as multisplitting preconditioned CG (MSPCG).

We note that while the multisplitting algorithm can split the general matrix A in a variety of ways, the splitting presented here, used as a preconditioner in CG, makes it equivalent to the additive Schwarz algorithm, which has been used for the Dirac equation inversion for the fermions [46, 47]. We use the name MSPCG, as it is through the process of applying the multisplitting algorithm to the DWF Dirac equation that we realize the necessity of including the snake terms in the local matrix.

5.5 Implementation Details

MSPCG reduces the amount of communication needed between processors by reducing the number of outer iterations at the expense of performing inner inversions. Comparing to plain CG it changes the demand for network bandwidth, GPU memory bandwidth and GPU compute flops. Being able to performing fast inner inversions is crucial to achieving an overall time-to-solution speed up since expensive local solves amortize the speed up gained from communication reduction.

The inner inversions are performed with plain CG, whose cost is dominated by the local preconditioner matrix multiplication (the inner dslash). In this section we detail the techniques used to improve the inner dslash performance on the SUMMIT machine.

The production version of MSPCG, specifically targeting SUMMIT, is implemented within the framework of QUDA[19] [47], a library for performing calculations in lattice QCD on GPUs leveraging NVIDIA's CUDA platform.

In this section we detail several of the important aspects of the implementation of MSPCG as well as the underlying technical motivation behind them.

5.5.1 The Original Hierarchy of Compute Flops, Memory and Network Bandwidth

Using CG to solve the Dirac equations without the preconditioning introduced in this work requires one Dirac matrix multiplication per iteration. As an example if the the $96 \times 96 \times 96 \times 128$ lattice is distributed onto $6 \times 8 \times 8 \times 16$ GPUs with one rank[20] per GPU the local 4 dimensional-volume is $8 \times 12 \times 12 \times 12$. The fifth dimension size is $L_s = 12$, the combined color-spin-complex (c-s-c) has a size of 24 and to represent a half-precision floating point number we need $(2 + 4/24)$ bytes[21].

The size of the fermion vector needed to be read from and save to GPU memory for each iteration is

$$\underbrace{8 \times 12 \times 12 \times 12}_{\text{local 4-d vol.}} \times \underbrace{12}_{L_s} \times \underbrace{24}_{\text{c-s-c}} \times \underbrace{2.17}_{\text{(2+4/24) byte}} \times \underbrace{2}_{\text{r/w}} \times \underbrace{4}_{\text{4 hops}} \times \underbrace{8 \cdot 50\%}_{\text{stencils}} = 276.4 \text{ MB.} \tag{238}$$

Note that this number already takes into account the kernel fusion that fuses the matrix multiplications in the fifth dimension into neighboring four dimensional hopping operations of (232). Details of this fusion will be covered later. The last factor of 50% takes into account the approximate L2 cache hit rate.

Meanwhile the amount of data that needs to be transferred between different nodes for one GPU is

$$\underbrace{8 \times 12 \times 12}_{\text{face surface area}} \times \underbrace{6}_{\text{6 faces}} \times \underbrace{12}_{L_s} \times \underbrace{24}_{\text{c-s-c}} \times \underbrace{2.17}_{\text{(2+4/24) byte}} \times \underbrace{\frac{1}{2}}_{\text{spin proj.}} \times \underbrace{2}_{\rightarrow/\leftarrow} \times \underbrace{4}_{\text{4 hops}} = 17.3 \text{ MB.} \tag{239}$$

A four dimensional hypercube has 8 faces but only 6 of them need inter-processor communication

[19] See https://github.com/lattice/quda.

[20] The term *rank* usually means the maximum software programing unit within which different threads share (CPU and/or GPU) memory. Threads on different ranks needs to exchange data through communication.

[21] For half precision QUDA represents the floating point numbers with a 16-bit signed integer together with a single precision floating point number that as the overall scale for the 24 numbers on a lattice site.

since the ranks distributed in the first dimension are placed on the same node. The factor $\frac{1}{2}$ comes from *spin projection* in which the γ-matrices' property of only being of rank-2 matrices is utilized.

For later discussions we also calculate the number of floating point operations (ops.) needed

$$\underbrace{8\times12\times12\times12}_{\text{local 4-d vol.}}\times\underbrace{12}_{L_s}\times\underbrace{24}_{\text{c-s-c}}\times[\underbrace{2\cdot(3\times2)-1}_{M_\phi}+\underbrace{8\times2\cdot(6\times2)-1}_{D_w}+\underbrace{2\cdot(12\times2)-1}_{M_5^{-1}}]\times\underbrace{4}_{\text{4 hops}}=3965\,\text{mega-ops.} \tag{240}$$

In table 21 we compare the compute flops[22] and the bandwidth we need (demand) to those we have on SUMMIT (supply). Notes that this comparison is only on-paper: latencies of memory and network transfers are not taken into account; GPU's parallelization strategy is more complicated than the over simplified picture represented by this table, which merely gives a proof-of-concept illustration. It is, however, clear that the overall performance is almost completely limited by the network bandwidth. Any effort in performance improvement needs to be focusing on reducing and/or hiding the communication overhead. One should, however, be aware that algorithmic changes could result in the change of demand for compute flops and bandwidth and previously insignificant part could become the dominant contributor to computational cost, which proves to be the case in this work.

	supply	demand per iteration with plain CG
compute flops (fp32)	15.7 tera-flops	2166 mega-ops
memory bandwidth	900 GB/sec	276.4 MB
network bandwidth	8.3 GB/sec	17.3 MB
compute/memory	17.4 ops/byte	14.3 ops/byte (0.8×supply)
network/memory	0.009	0.062 (7.0×supply)

Table 21: Compute flops, memory and network bandwidth demand versus supply on SUMMIT.

5.5.2 Capturing the Snake Terms

The first challenge even before all the later optimizations is how to implement the local preconditioning matrix multiplication (the inner dslash) to effectively capture the snake terms. Our

[22] For half precision in QUDA the actual floating point computing part is still performed with single precision

approach here is to pad the fermion fields so that the existing functions that performs the various multiplications can be used with no or little modifications. There are several reasons why this approach is chosen:

- Compared to writing new code to calculate the snake terms directly reusing the existing code reduces the amount of coding needed to be done.
- This approach inherits the already optimized parallelizing strategy used by the existing functions.
- Compared to calculating each of the 4-hop paths independently this approach calculates one hop at a time and allows the later hops to used the results of the previous ones, which greatly increases computation reuse.

As our strategy the fermion field is padded in both directions of each dimension by size of 2 and the original field is placed at the center of the padded field while the padded region is initialized to zero. The operators in equation (232) of the full normal operator are then applied to the padded lattice vector sequentially with zero Dirichlet boundary condition on the padded boundary. The center of the resulting vector thus includes the snake terms: their 4 hops are propagated through the padded and center region. The padding size is chosen to be 2 since the result we want is the vector content at the center. With only 4 hops, in order to go back to the center the snake terms are only able to hop out of the original boundary by as far as 2. An illustration can be found in figure 35.

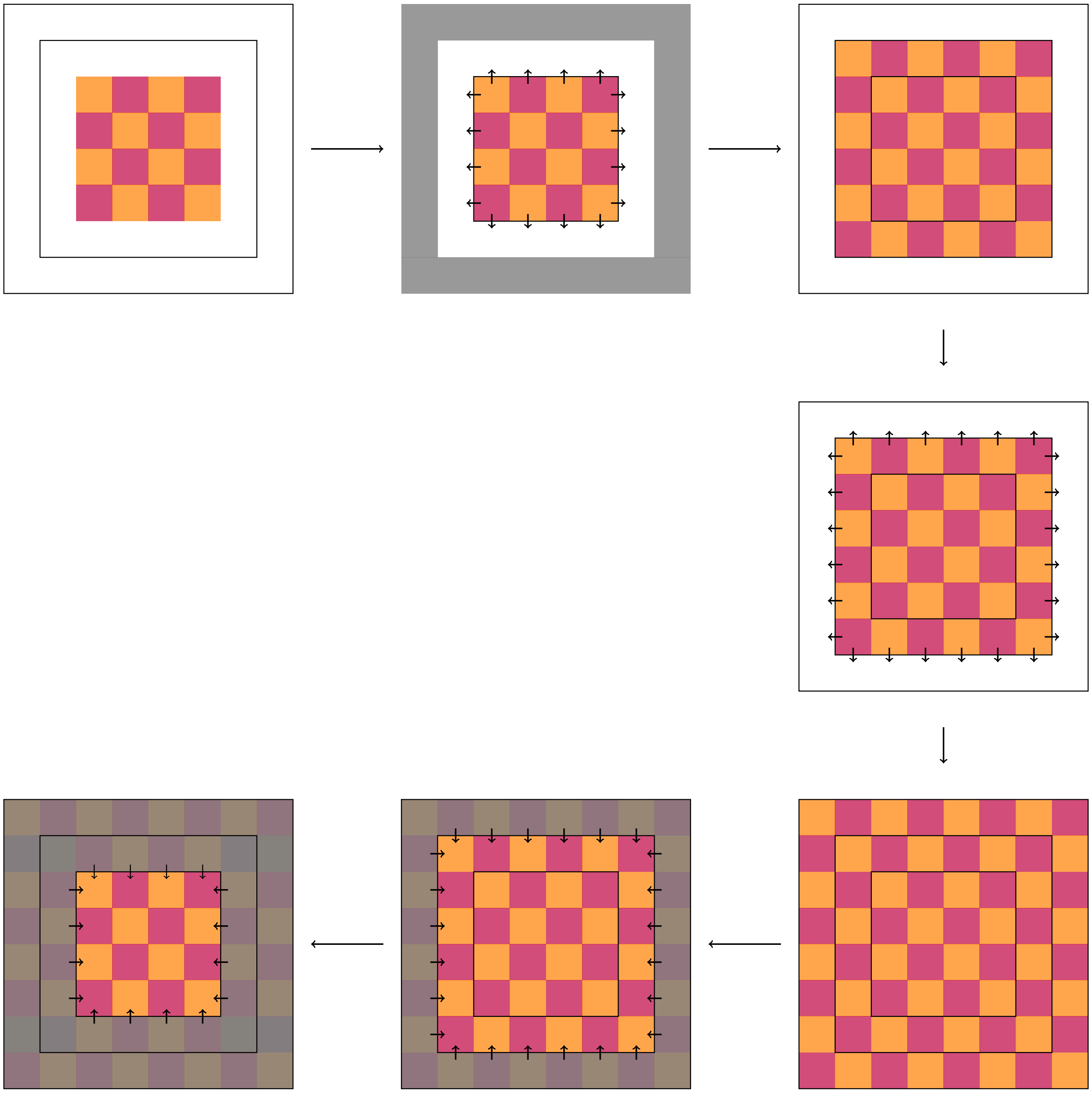

Figure 35: An illustration on using padded fermion field to capture the snake terms by applying the hopping terms sequentially.

5.5.3 Kernel Fusion

A closer look at (232) gives the following series of operators.

$$A = D_{prec}^{\dagger} D_{prec} = \left[1 - \kappa_b^2 M_\phi^\dagger D_w^\dagger M_5^{-\dagger} M_\phi^\dagger D_w^\dagger M_5^{-\dagger}\right]\left[1 - \kappa_b^2 M_5^{-1} D_w M_\phi M_5^{-1} D_w M_\phi\right]. \tag{241}$$

Mathematically the operators are matrices to be left multiplied on the vector sequentially; numerically the operators are functions to be called to apply these multiplications; on the GPUs the operators are implemented as *kernels* to be launched. Without diving into details of GPU programming each of these kernels carries out these three steps:

1. Load the input vector from memory into the registers;
2. Perform computation with the data loaded in the registers;
3. Store the output vector from the registers to memory.

Skipping the $[1 - ...][1 - ...]$ part in (241) for simplicity we have 12 operators/kernels to be applied sequentially and input data of the kernels is exactly the output data of the ones right before them. Instead of storing the vector all the way down to memory and then loading the very same vector all the way up to the registers a better strategy is to keep all the data in cache to avoid the memory traffic and latency from the repeated loading and storing. This cache reuse is automatically done by the hardware but it is limited by the cache size: the lowest level L2 cache size is 6 MB on a Tesla V100 while in our previous example the data in the vector has a size of around 9 MB: the L2 cache is not large enough to hold all the data.

In this work more cache reuse is achieved by utilizing the structure of the matrices and the fact that Tesla V100 has a programable L1 cache (called *shared memory* in NVIDIA's terminology). The D_w's in (241) are matrices that hop in space-time dimensions (4d operator) and are diagonal in the fifth dimension; the M_ϕ's and M_5^{-1}'s in (241) are matrices that only operate in the fifth dimension (5d operators) and are diagonal in the space-time dimensions. The vector is divided into blocks according to its space-time coordinates. For each block the 4d operators are first applied and the result are held in the L1 cache; for the 5d operators right after this what is held in the L1 cache is all they need for performing the computation due to their diagonality in the space-time dimensions. Naturally the 4d and 5d operators/kernels are fused into one combined kernel. The size of the block is chosen such that the size of the L1 cache is large enough to hold all the intermediate data.

The kernel fusion approach outperforms the previous strategy for two reasons:

- The L1 cache, as a higher level cache than L2, has faster traffic and lower latency.

- With cache reuse explicitly programed at the software level it is more consistent than relying on run time hardware level cache reuse.

Thus instead of launching 12 kernels for the 12 matrices only 5 kernels are launched ($[1-...][1-...]$ part is again skipped for simplicity):

$$M_{\phi}^{\dagger}D_{w}^{\dagger}, \quad M_5^{-\dagger}M_{\phi}^{\dagger}D_{w}^{\dagger}, \quad M_5^{-\dagger}M_5^{-1}D_w, \quad M_{\phi}M_5^{-1}D_w, \quad M_{\phi}. \tag{242}$$

5.5.4 Changed Hierarchy

Before discussing further into the implementation details a review of the numerical situation is needed to motivate the use of the tensor cores available on the Tesla V100 GPUs.

With the introduction of the preconditioning inversions in each CG iteration more local floating point operations and local memory traffic is demanded. In our simulations typically 6 inner dslashes are needed for the inner inversion; these matrix multiplications operate on the padded fermion field and thus consume more flops and memory bandwidth than their non-padded counterparts. As an illustration both the compute flops and memory bandwidth are estimated to increase by a factor of 12. The updated supply-demand comparison is shown in table 22.

	supply	demand per iteration with plain CG	demand per outer iteration with MSPCG
compute flops (fp32)	15.7 tera-flops	3965 mega-ops	3965 × 12 mega-ops
memory bandwidth	900 GB/sec	276.4 MB	276.4 × 12 MB
network bandwidth	8.3 GB/sec	17.3 MB	17.3 MB
compute/memory	17.4 ops/byte	14.3 ops/byte (0.8×supply)	14.3 ops/byte (0.8×supply)
network/memory	0.009	0.062 (7.0×supply)	0.005 (0.6×supply)

Table 22: Compute flops, memory and network bandwidth demand versus supply on SUMMIT with MSPCG.

The overall performance is no longer only limited by the network bandwidth: the compute flops and memory bandwidth also get close to the hardware limits and consumes a considerable amount of computing time.

5.5.5 Tensor Core

The M_5^{-1} operator in (241) is a dense matrix multiplication in the fifth dimension. Following the kernel fusion strategy described earlier the input data is held in L1 cache (shared memory) before the matrix multiplication is applied: no memory loading is needed. On the other hand compute flops become one of the bottlenecks of the overall performance of MSPCG, motivating us to use the *tensor cores* available on the Tesla V100 GPUs to accelerate the M_5^{-1} operations.

The tensor cores used in this work perform matrix multiplication and accumulation of the form $D = AB + C$ at the hardware level intrinsically instead of performing the same operation with the traditional column-row loops. These cores perform the multiplication with half precision[23] and the accumulation with either half or single precision with a stunning peak performance of 125 tera-flops, as compared to that of the traditional single precision operations of 15.7 tera-flops. The use of tensor cores fits well in the kernel fusion framework: it does not introduce additional memory/cache management overhead. On the compute side it speeds up the 5d operation part of the matrix multiplication by almost an order of magnitude.

The only overhead introduced by using tensor cores comes from the much narrower representation range of half precision (6.1×10^{-5} to 65504), compared to that of single precision (1.18×10^{-38} to 1.4×10^{45}). The maximum absolute value of the all the numbers in the block is found and the numbers need to be scaled accordingly to be representable in half precision before the tensor core are used. After the matrix multiplication the numbers are scaled back before being stored to memory as output. With highly parallelized reduction and scaling code this overhead is negligible compared to the overall performance.

While the tensor cores are designed mostly to "accelerate large matrix operations, which are at the heart of AI, and perform mixed-precision matrix multiply and accumulate calculations in a single operation"[24] we are able to use them, possibly for the first time in high performance computing, in our simulation.

5.5.6 L1 Cache Management Design

"Lattice coding is *simply* index calculations." – said someone who has done lots of lattice coding.

Turning lattice calculation ideas (brilliant or not) into correctly running code on a specific hardware involves assigning data to different threads, retrieving the assigned data, performing the desired calculation and finally storing the results back to memory. The whole process relies on a

[23] In this case half precision is the IEEE half precision standard, instead of the QUDA one.

[24] Quoted from https://www.nvidia.com/en-us/data-center/tensorcore/.

clear mapping between the physical indices in lattice QCD, including the four spatial dimensions, the fifth dimension, spin, color, complex, and possibly the four directions for the gauge links, and one single address location (machine index) in the data buffer.

Almost all code bases used in this work, either directly or indirectly, calculate the machine index with the lexicographic order of a permutation of the physical indices. Usually the permutations are simply called *the order*. One possible order, for example, from the slowest changing index to the fastest changing index, is

- the four spatial dimensions, the fifth dimension, spin, color, complex.

The order can be very different depending on the code base, fermion formulation, collaboration history and personal taste.

In this work for the kernel fusion and the later tensor core usage a separate mapping between physical indices and the machine index on the programmable L1 cache is needed. For kernel fusion any 1-on-1 mapping satisfies the demand but in order to use the tensor cores the fifth dimension and the spin indices need to be separated from the other indices. The order we end up choosing is the following,

- the four spatial dimensions, color, complex, the fifth dimension, spin.

Note that the four spatial dimensions here is only a subset of the entire four spatial dimension. This order is illustrated in figure 36.

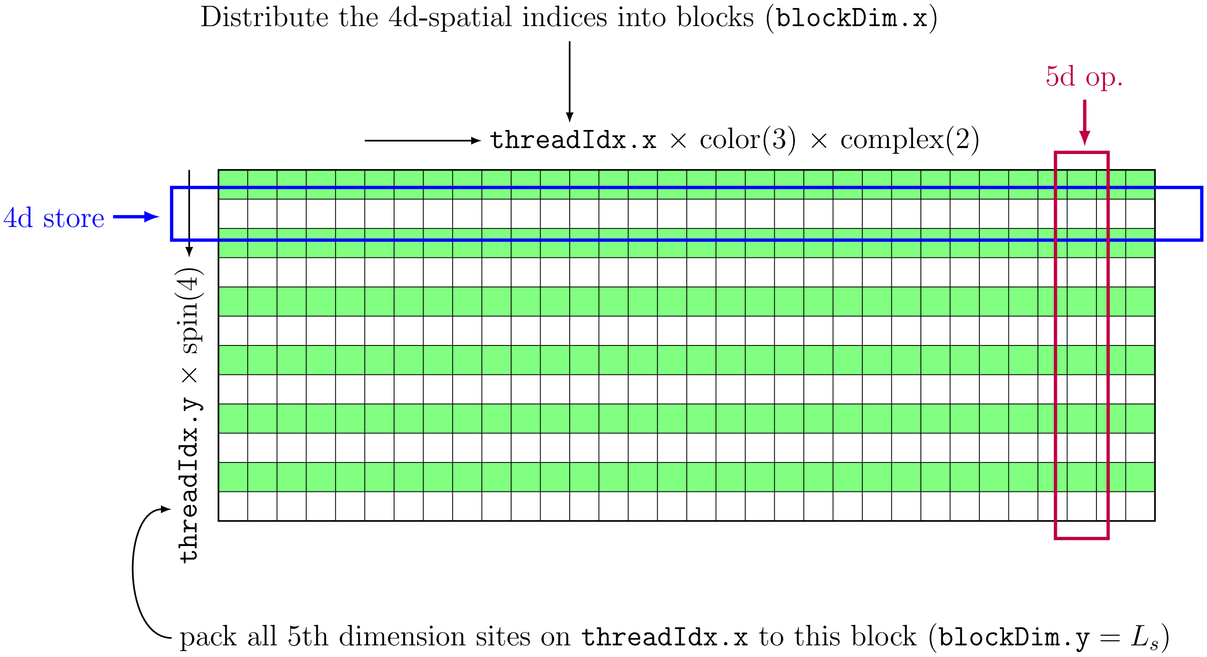
Distribute the 4d-spatial indices into blocks (blockDim.x)
5d op.
threadIdx.x × color(3) × complex(2)
4d store
threadIdx.y × spin(4)
pack all 5th dimension sites on threadIdx.x to this block (blockDim.y = L_s)

Figure 36: Illustration of the L1 cache memory mapping between physical indices and the machine index. The `blockDim.x`, `threadIdx.x`, `blockDim.y`, `threadIdx.y` are CUDA terminologies. See http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html.

With this memory design the tensor core matrix multiplications are performed with the sub-block multiplication fashion shown in figure 37.

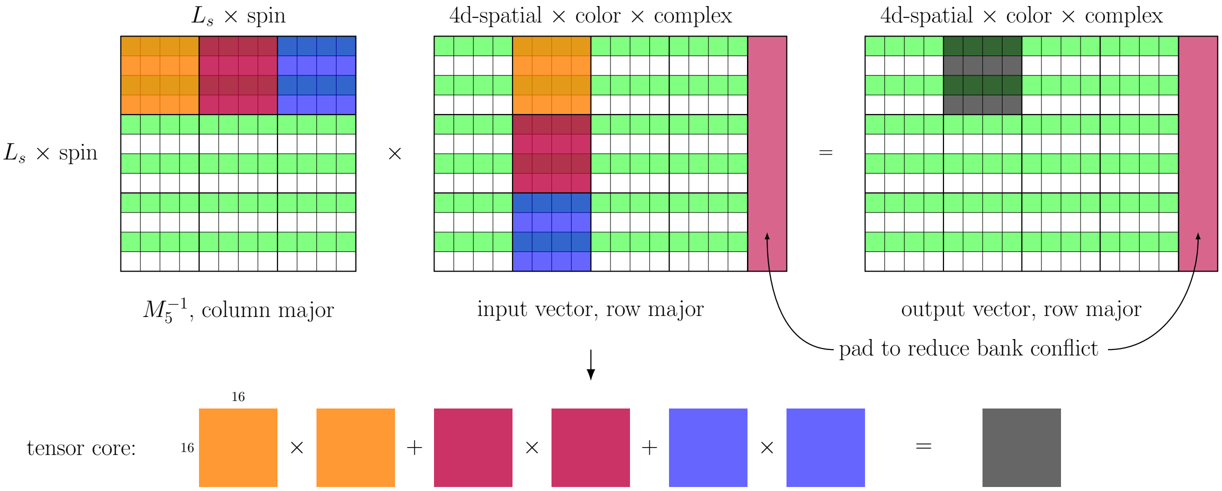

Figure 37: Illustration of the way tensor core matrix multiplications are performed for the fifth dimension operations.

5.6 The Results

5.6.1 The Light Quark Sector

The multisplitting preconditioned CG is applied to solving the Dirac equations for the light quark determinant in our target evolution run on SUMMIT.

Standard CG is used to perform the inner inversion. Instead of adopting a precision based stopping condition, a fixed number of inner iterations, are performed for these preconditioning solves. In table 23 the numbers of outer iterations and time-to-solution needed for the preconditioned CG to converge are reported on the different number of SUMMIT nodes, together with the performance achieved and the timing partition. The numbers of iterations to reach the same precision with standard CG are also included for comparison.

With 6 inner iterations the preconditioned CG reduces the outer iteration count by a factor of 3. More inner iterations reduce the outer iteration count more but the reduction saturates as the inner iteration count increases: with large number of inner iterations the inner CG solves the preconditioning inversion completely and no further numerical benefit can be exploited.

The preconditioned CG converges algorithmically if the preconditioner (A_{pre} in algorithm 2) is fixed and its inversions are performed exactly. In our case with as few as 6 inner iterations the preconditioner is only solved to a very relaxed stopping condition (See figure 38) and effectively the preconditioner becomes unfixed and inexact. While the convergence of inexact preconditioned CG method with inner-outer iteration is proved in [42] our stopping condition is far more relaxed than their convergence condition. This pessimism from the given convergence condition is mentioned in [42] itself as well.

The CG inversion we perform includes a combination of inexact preconditioning and mixed double (`fp64`) and half (`fp16`) precision. It is found that the use of the Polak-Ribière formula and the more sophisticated reliable update scheme, both introduced in 5.2.1, is necessary to secure a stable and smooth convergence of MSPCG. The residuals and iteration counts are plotted in figure 38 for the solve performed on 1024 SUMMIT nodes. In table 23 the strong scaling performance data is given.

nodes	local volume	solver	inner iter.	(outer) iter.	r.u.	performance/node	time	speed up
256	$16 \cdot 24 \cdot 12 \cdot 24$	CG	–	42133	471	4.66	486.3	1.10x
		MSPCG	05	16903	195	1.56(01)/5.45(35)/37.29(53)	456.0	
		MSPCG	06	14860	173	1.56(01)/5.51(31)/37.60(58)	442.6	
		MSPCG	07	13787	161	1.56(01)/5.48(28)/37.49(60)	460.2	
		MSPCG	08	12922	151	1.56(01)/5.44(26)/37.55(63)	469.5	
512	$16 \cdot 12 \cdot 12 \cdot 24$	CG	–	42427	474	3.85	296.6	1.13x
		MSPCG	05	17625	203	1.26(01)/4.54(37)/36.21(52)	271.0	
		MSPCG	06	15425	179	1.27(01)/4.55(33)/36.26(57)	262.1	
		MSPCG	07	14409	168	1.26(01)/4.57(30)/36.39(60)	268.3	
		MSPCG	08	13597	159	1.27(01)/4.53(28)/36.35(63)	276.0	
1024	$16 \cdot 12 \cdot 12 \cdot 12$	CG	–	42482	474	2.93	195.2	1.22x
		MSPCG	05	18250	210	1.00(01)/3.68(34)/34.62(45)	183.3	
		MSPCG	06	15959	185	1.01(01)/3.68(35)/34.79(54)	159.7	
		MSPCG	07	14985	174	1.01(01)/3.68(32)/35.06(58)	163.6	
		MSPCG	08	14287	167	1.00(01)/3.69(29)/34.76(61)	169.1	

Table 23: Strong scaling of the MSPCG on SUMMIT solving Dirac equation $M^\dagger M x = y$ to the accuracy of 10^{-12} on the $96^3 \times 192 \times 12$, 2+1 flavor Möbius domain wall fermion, $a^{-1} \simeq 2.8$ GeV lattice with physical pion mass. y is a gaussian random source vector. The time are time-to-solutions given in units of second. r.u. means number of reliable updates performed. Performance numbers are expressed in tera-flops per node. For CG solves the performance is given as the total performance, including precise and sloppy dslash and linear algebra operations. For MSPCG solves the performance is given in format of precise/sloppy/precondition dslash with their respective time percentage in parentheses.

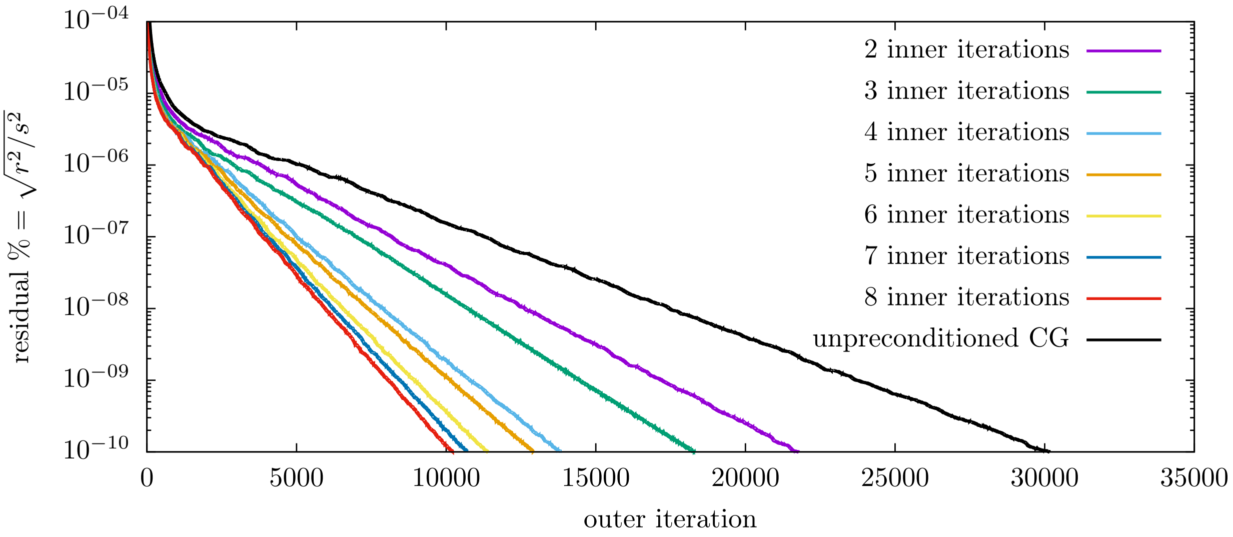

Figure 38: Residual versus (outer) iteration of the MSPCG on SUMMIT solving Dirac equation $M^\dagger Mx = M^\dagger y$ to the accuracy of 10^{-10} on the $96^3 \times 192 \times 12$, 2+1 flavor Möbius domain wall fermion, $a^{-1} \simeq 2.8$ GeV lattice with physical pion mass. y is a gaussian random source vector.

5.6.2 The Heavy Quark Sector and the Rest

The EOFA algorithm is, for the first time, targeting GPU based hardware. It is implemented in QUDA to calculate the fermion determinant for strange quark in our target evolution. Compared to the RHMC algorithm, EOFA speeds up the heavy quark part of the evolution significantly by greatly reducing the amount of force term calculations. The achieved evolution timing, including the optimizations for both the light quark and heavy quark sectors, is shown in table 24.

heavy quark (EOFA)	18.2 %
light quark (quotient)	73.9 %
gauge	7.2 %
total time (512 nodes)	6328 seconds

Table 24: Achieved timing of the target lattice generation on SUMMIT.

5.7 Conclusion and Discussion

As a conclusion MSPCG is able to reduce the (outer) iteration count by a factor of 3 with as few as 6 inner preconditioning iterations and achieve an around 20% time-to-solution speed up on 1024

SUMMIT nodes utilizing the tensor core units on the Tesla V100 GPUs, together various other optimizations and improvements. This is the first time the tensor cores, which are specifically designed for deep learning operations, are used to speed up lattice QCD simulations.

The speed up is not, admittedly, significant compared to other exciting numbers the computer science field is seeing, yet it proves the possibility of using local preconditioning to improve scaling of lattice QCD simulations with (Möbius) domain wall fermion. Currently the performance of the preconditioning step is mainly limited by the GPU memory bandwidth and its cache efficiency, which are likely to be greatly improved in the near future, especially on the exascale supercomputers. The network bandwidth, however, is not likely to have a comparable increase, given that lattice QCD (with domain wall fermion) is almost the only field that has such large demand for inter-processor communication.

As an example we compare the SUMMIT system to the TITAN system, which was also launched at ORNL in 2010. Each TITAN node has a network bandwidth of 25 GB/sec and 1 Kepler 20X GPU, which has a memory bandwidth of 235 GB/sec.[25] On the other hand each SUMMIT node has a network bandwidth of 50 GB/sec and 6 Tesla V100 GPUs, each has a memory bandwidth of 900 GB/sec. As a node to node comparison over the course of eight years the network bandwidth has increased by a factor of 2 while the total GPU memory bandwidth has increased by a factor of 23.

MSPCG is expected to bring more significant speed up with this ever growing inequality between network and memory bandwidth (under the lattice QCD perspective). The idea of using local preconditioning to improve overall scaling can also be applied to lattice QCD simulation with other fermion formulations and other scientific computing fields.

[25] See https://www.olcf.ornl.gov/wp-content/uploads/2013/02/Titan_Architecture_1-JL.pdf.

6 Conclusion and Future Works

We conclude this thesis by considering how future research can be built upon this work.

6.1 The Weak Coupling Limit

Since the lattice spacing a serves as the ultraviolet cut off for lattice gauge theory, lattice QCD simulations have to scale towards the weak coupling limit ($a \to 0$) to be able to reliably address physical processes involving valence heavy quarks (charm, bottom, and even top quark). Two aspects of the critical slowing down, however, poses significant difficulties to this scaling.

As the lattice spacing grows finer, the lattice volume that is needed to have a large enough physical volume increases according to a^{-4}. While simulating with physical quark masses the computational resources needed to perform the fermion matrix inversions increase as $a \to 0$. This is the solver aspect of the critical slowing down. Over the years large parallel supercomputers have been designed and built to provide the computational capabilities needed. The inter-processor communication on these state-of-art supercomputers, however, is increasingly becoming the bottleneck of the overall performance, despite their high on-processor flops and memory bandwidth.

In section 5 we show that it is possible to trade local on-processor flops and memory bandwidth for inter-processor communication, thus improving the scaling of the computational resources for lattice QCD simulations and achieving a speed up with the multisplitting preconditioned conjugate gradient (MSPCG) algorithm. The algorithm is expected to bring greater speed up on exascale supercomputers, where even greater inequality between local on-processor computational power and inter-processor communication is expected.

There is also an ongoing effort to reduce the inner inversion cost to further speed up the MSPCG on the current pre-exascale supercomputers by using cheap approximation to perform the inner inversions.

The algorithmic aspect of the critical slowing down, where the amount of molecular dynamics time needed to generate independent gauge configurations by hybrid Monte Carlo (HMC) simulations increases dramatically, proves to be an even more serious problem for lattice QCD. It has been observed that the integrated autocorrelation time of the topological charge, for example, of the gauge configurations grows as a^{-5}. Correlations between gauge configurations makes it difficult to reliably estimate the error of the quantities measured on these configurations.

In section 2 we apply the diffusion model, a model that describes how topological charge density evolves during HMC evolutions, to lattices with dynamical fermions and observe the expected diffusing behavior.

In section 3 we studied matching lattice actions at different scales to have the same low energy physics. A blocking kernel is tuned to block a fine lattice into a coarse one. The measurements made on this blocked coarse lattice are found to deviate only a few percent away from those made on a simulated blocked lattice. While exactly matched lattice actions at two scales lead to a multiscale evolution algorithm without critical slowing down, the high numerical accuracy needed is not found in our study.

Despite these studies, evolution algorithms that are free from the critical slowing down are still needed to move lattice simulations towards the weak coupling limit. There are multiple ongoing efforts, including part of the exascale computing project (ECP), that are expected to solve or at least levitate the problem.

6.2 The Strong Coupling Limit

Lattice QCD simulations are always performed with finite amount of computational resources. It is worthwhile to explore ways to perform cheap lattice simulations while having errors, especially the finite lattice spacing error, under control. The coarse lattices, which the RBC/UKQCD collaborations have generated with Iwasaki gauge action and the dislocation suppressed determinant ratio (DSDR) term (**ID**) as well as Möbius domain wall fermions, serve as compelling candidates towards this goal.

In section 4 we measure the kaon bag parameter B_K and the $\Delta I = 1/2$ $K \to \pi\pi$ amplitude A_2 on the coarse 24**ID**($\beta = 1.633$) lattice. The finite lattice spacing corrections are examined by comparing the results with those measured on the 32**ID**($\beta = 1.75$) lattice, as well as the continuum limit values obtained from two finer Iwasaki lattices. The a^2 dependences of these two quantities does not reveal a measurable a^4 term in the continuum limit extrapolation. This simple a^2 scaling property makes these coarse **ID** lattices among the ideal ones to measure low energy physics with low computational cost.

There is, however, evidence from hadronic vacuum polarization (HVP) calculations that a^4

terms are needed to perform continuum limit extrapolation for matrix elements involving the local vector current. We have no evidence, to date, of a^4 scaling terms being needed in the spectrum (eigenvalues of the Hamiltonian). Further studies on the a^2 scaling property on these coarse **ID** lattices are still needed.

References

[1] M. F. Atiyah and I. M. Singer, Bulletin of the American Mathematical Society **69**, 422 (1963).

[2] E. J. Weinberg, *Classical solutions in quantum field theory: Solitons and Instantons in High Energy Physics* (Cambridge University Press, 2012).

[3] R. J. Crewther et al., Physics Letters B **88**, 123 (1979).

[4] H. B. Nielsen and M. Ninomiya, Physics Letters B **105**, 219 (1981).

[5] R. C. Brower, H. Neff, and K. Orginos, Nuclear Physics B - Proceedings Supplements **140**, 686 (2005), [arXiv:hep-lat/0409118].

[6] P. Vranas, Nuclear Physics B - Proceedings Supplements **53**, 278 (1997), [arXiv:hep-lat/9608078].

[7] A. D. Kennedy, Nuclear Physics B (Proceedings Supplements) **4**, 576 (1988).

[8] A. D. Kennedy, Nuclear Physics B - Proceedings Supplements **128**, 107 (2004), [arXiv:hep-lat/0402037].

[9] C. Jung et al., Physical Review D **97** (2018), [arXiv:1706.05843].

[10] M. Lüscher, Communications in Mathematical Physics **54**, 283 (1977).

[11] P. A. Boyle et al., Physical Review D **93**, 054502 (2016), [arXiv:1511.01950].

[12] M. Lüscher, Journal of High Energy Physics **2010**, 1 (2010), [arXiv:1006.4518].

[13] S. Borsányi et al., Journal of High Energy Physics **2012** (2012), [arXiv:arXiv:1203.4469v2].

[14] S. Schaefer, R. Sommer, and F. Virotta, Nuclear Physics B **845**, 93 (2011), [arXiv:1009.5228].

[15] G. McGlynn and R. D. Mawhinney, Physical Review D **90**, 1 (2014), [arXiv:1406.4551].

[16] P. de Forcrand, M. García Pérez, and I.-O. Stamatescu, Nuclear Physics B **499**, 409 (1997), [arXiv:hep-lat/9701012].

[17] T. Blum et al., Physical Review D **93** (2016), [arXiv:1411.7017].

[18] J. Sexton and D. Weingarten, Nuclear Physics B (1992).

[19] R. Arthur et al., Physical Review D - Particles, Fields, Gravitation and Cosmology **87** (2013), [arXiv:1208.4412].

[20] A. Bazavov et al., (2017), [arXiv:1712.09262].

[21] N. Madras and a. D. Sokal, J. Stat. Phys. **50**, 109 (1988).

[22] Y. Iwasaki, UTHEP **118**, 20 (1983), [arXiv:1111.7054].

[23] M. Albanese et al., Physics Letters B **192**, 163 (1987).

[24] M. Hasenbusch, K. Pinn, and C. Wieczerkowski, Physics Letters B **338**, 308 (1994), [arXiv:hep-lat/9406019].

[25] Y. Aoki et al., Physical Review D **78**, 1 (2008), [arXiv:0712.1061].

[26] M. G. Endres et al., Physical Review D **92** (2015), [arXiv:1510.04675].

[27] C. Sturm et al., Physical Review D - Particles, Fields, Gravitation and Cosmology **80** (2009), [arXiv:0901.2599].

[28] R. Arthur and P. A. Boyle, Physical Review D **83**, 1 (2011), [arXiv:1006.0422].

[29] A. J. Buras, (1998), [arXiv:hep-ph/9806471].

[30] N. Garron, R. J. Hudspith, and A. T. Lytle, Journal of High Energy Physics **2016**, 1 (2016), [arXiv:hep-lat/1609.03334].

[31] P. A. Boyle et al., Journal of High Energy Physics **2017**, 1 (2017), [arXiv:1708.03552].

[32] D. Beirevi and G. Villadoro, Physical Review D - Particles, Fields, Gravitation and Cosmology **69** (2004), [arXiv:hep-lat/0311028].

[33] T. Blum et al., Physical Review D - Particles, Fields, Gravitation and Cosmology **91** (2015), [arXiv:1502.00263].

[34] C. Kim, Nuclear Physics B - Proceedings Supplements **140**, 381 (2005).

[35] L. Lellouch and M. Lüscher, Communications in Mathematical Physics **219**, 31 (2001), [arXiv:hep-lat/0003023].

[36] T. Yamazaki et al., Physical Review D - Particles, Fields, Gravitation and Cosmology **70** (2004), [arXiv:hep-lat/0402025].

[37] T. Blum et al., Physical Review D - Particles, Fields, Gravitation and Cosmology **86**, 074513 (2012), [arXiv:1206.5142].

[38] A. Stathopoulos and K. Orginos, SIAM Journal on Scientific Computing **32**, 439 (2010).

[39] Y. Saad, SIAM Journal on Numerical Analysis **17**, 687 (1980).

[40] M. Lüscher, Computer Physics Communications **156**, 209 (2004), [arXiv:hep-lat/0310048].

[41] R. Winther and W. Hacksbusch, Mathematics of Computation **64**, 1759 (2006).

[42] G. H. Golub and Q. Ye, SIAM Journal on Scientific Computing **21**, 1305 (1999).

[43] H. A. van der Vorst and Q. Ye, SIAM Journal on Scientific Computing **22**, 835 (2000).

[44] D. P. O'Leary and R. E. White, SIAM Journal on Algebraic Discrete Methods **6**, 630 (1985).

[45] F. Jezequel, R. Couturier, and C. Denis, Journal of Supercomputing **59**, 1517 (2012).

[46] Y. Osaki and K. I. Ishikawa, Proceedings of Science **105**, 1 (2010), [arXiv:1011.3318].

[47] R. Babich et al., Proceedings of 2011 SC - International Conference for High Performance Computing, Networking, Storage and Analysis (2011), [arXiv:1109.2935].

[48] R. G. Miller, Biometrika **61**, 1 (1974).

[49] B. Hall, *Lie Groups, Lie Algebras, and Representations - An Elementary Introduction* (2015).

Appendices

A.1 Minkowski γ-matrices

The Minkowski γ-matrices are defined by the following anti-commutation relations:

$$\{\gamma^\mu, \gamma^\nu\} = 2g^{\mu\nu}\mathbb{1}. \tag{243}$$

A.1.1 The Dirac Representation

$$\gamma^0 = \begin{pmatrix} +\mathbb{1} & 0 \\ 0 & -\mathbb{1} \end{pmatrix}, \quad \gamma^i = \begin{pmatrix} 0 & +\sigma^i \\ -\sigma^i & 0 \end{pmatrix}, \quad \gamma^5 = \begin{pmatrix} 0 & +\mathbb{1} \\ +\mathbb{1} & 0 \end{pmatrix}. \tag{244}$$

A.1.2 The Weyl/Chiral Representation

$$\gamma^0 = \begin{pmatrix} 0 & +\mathbb{1} \\ +\mathbb{1} & 0 \end{pmatrix}, \quad \gamma^i = \begin{pmatrix} 0 & +\sigma^i \\ -\sigma^i & 0 \end{pmatrix}, \quad \gamma^5 = \begin{pmatrix} -\mathbb{1} & 0 \\ 0 & +\mathbb{1} \end{pmatrix}. \tag{245}$$

A.2 Lattice (Euclidean) γ-matrices

A.2.1 Definition

The Euclidean/lattice γ-matrices are defined by the following anti-commutation relation and we will be considering the four dimensional case only, i.e. $\mu, \nu = 1, 2, 3, 4$.

$$\{\gamma_\mu, \gamma_\nu\} = 2\delta_{\mu\nu}\mathbb{1}. \tag{246}$$

Several representations exist, yet the question of how to transform between different representations is not well presented in most of the literature. Specifically if we have two representations, γ_μ and $\tilde{\gamma}_\mu$, they are related by an similarity transformation,

$$S^{-1}\tilde{\gamma}_\mu S = \gamma_\mu. \tag{247}$$

The question is therefore given these two representations how do we find this transformation S?

A.2.2 Construction of the Algebra

The way to construct the algebra needed is detailed in [Jauch, Rohrlich, 1976]. In the following repeated symbol does not imply summation.

Five sets of matrices $\Gamma_t (t = 1, 2, 3, 4, 5)$ are constructed from the four matrices γ_μ's:

$$\begin{aligned}
&\Gamma_1 : \ \mathbb{1}, \\
&\Gamma_2 : \ \gamma_\mu, \\
&\Gamma_3 : \ \gamma_\mu \gamma_\nu (\mu < \nu) \\
&\Gamma_4 : \ \gamma_5 \gamma_\mu, \\
&\Gamma_5 : \ \gamma_5 = \gamma_1 \gamma_2 \gamma_3 \gamma_4.
\end{aligned}$$

These 16 matrices constructed, which will be called $\gamma_r (r = 1, 2, \cdots, 16)$, are linearly independent and form a basis Γ. They square to ± 1, i.e.

$$\gamma_r \gamma_r = \xi_r = \pm 1. \tag{248}$$

Furthermore, the product of any two elements of Γ is equal to another element up to a sign,

$$\gamma_r \gamma_s = \epsilon_{rs} \gamma_{\langle r,s \rangle}. \tag{249}$$

Due to the anitcommuting nature we have

$$\langle r, s \rangle = \langle s, r \rangle.$$

Now with some trick,

$$\gamma_r \gamma_s \gamma_r \gamma_s = \epsilon_{rs}^2 \gamma_{\langle r,s \rangle} \gamma_{\langle r,s \rangle} = \xi_{\langle r,s \rangle}, \tag{250}$$

left multiply by γ_r and right multiply by γ_s,

$$\xi_r \xi_s \gamma_s \gamma_r = \xi_{\langle r,s \rangle} \gamma_r \gamma_s \rightarrow \xi_r \xi_s \epsilon_{rs} \gamma_{\langle r,s \rangle} = \xi_{\langle r,s \rangle} \epsilon_{sr} \gamma_{\langle r,s \rangle}. \tag{251}$$

or,

$$\xi_r \xi_s \epsilon_{rs} = \xi_t \epsilon_{sr}. \tag{252}$$

A.2.3 Construction of the Transformation Matrix

The claim is that the transformation matrix in equation (247) is

$$S = \sum_r \xi_r \tilde{\gamma}_r F \gamma_r, \tag{253}$$

where the matrices $\tilde{\gamma}_r$'s are constructed in the way γ_r's are constructed and F is an arbitrary 4 by 4 matrix that makes S invertible. Here is the proof:

$$\begin{aligned}
\tilde{\gamma}_s S \gamma_s &= \sum_r \xi_r \tilde{\gamma}_s \tilde{\gamma}_r F \gamma_r \gamma_s \\
&= \sum_r \xi_r \epsilon_{sr} \epsilon_{rs} \tilde{\gamma}_{\langle r,s \rangle} F \gamma_{\langle r,s \rangle} \\
&= \sum_r \xi_r \xi_{\langle r,s \rangle} \xi_r \xi_s \tilde{\gamma}_{\langle r,s \rangle} F \gamma_{\langle r,s \rangle} \\
&= \sum_r \xi_s \xi_{\langle r,s \rangle} \tilde{\gamma}_{\langle r,s \rangle} F \gamma_{\langle r,s \rangle} \\
&= \xi_s S,
\end{aligned}$$

left multiply by S^{-1} and right multiply by γ_s and the claim is proved.

A.2.4 CPS/Grid Representation

$$\gamma_{1,3} = \begin{pmatrix} 0 & +i\sigma^{1,3} \\ -i\sigma^{1,3} & 0 \end{pmatrix}, \; \gamma_2 = \begin{pmatrix} 0 & -i\sigma^2 \\ +i\sigma^2 & 0 \end{pmatrix}, \; \gamma_4 = \begin{pmatrix} 0 & +\mathbb{1} \\ +\mathbb{1} & 0 \end{pmatrix}, \; \gamma_5 = \begin{pmatrix} +\mathbb{1} & 0 \\ 0 & -\mathbb{1} \end{pmatrix}.$$

A.2.5 QUDA Representation

$$\gamma_i = \begin{pmatrix} 0 & +i\sigma^i \\ -i\sigma^i & 0 \end{pmatrix}, \; \gamma_4 = \begin{pmatrix} +\mathbb{1} & 0 \\ 0 & -\mathbb{1} \end{pmatrix}, \; \gamma_5 = \begin{pmatrix} 0 & +\mathbb{1} \\ +\mathbb{1} & 0 \end{pmatrix}.$$

A.3 Jackknife Resampling Technique [48]

Let $[X_1, \cdots, X_n]$ be n (*independent and identically distributed*) random variables. We split this sample into g groups of size h each, $n = gh$[26]. Let T be an estimator of some parameter θ based on sample size n, T_{-i} be the corresponding estimator based on the sample of size $(g-1)h$, where the i-th group of size h has been excluded:

$$T = f[X_1, \cdots, X_n],$$

$$T_{-i} = f[X_1, \cdots, X_{(i-1)h}, X_{ih+1}, \cdots, X_n].$$

Write $\bar{T}_\bullet$ as the mean of the T_{-i}'s, the jackknife estimator is defined as

$$T_j = gT - (g-1)\bar{T}_\bullet, \ \bar{T}_\bullet = \langle T_{-i} \rangle_i = \frac{1}{g}\sum_{i=1}^{g} T_{-i},$$

and the estimator of the variance of the T_j is

$$\text{var}[T_j] = \frac{g-1}{g}\sum_{i=1}^{g}(T_{-i} - \bar{T}_\bullet)^2.$$

Usually we use T instead of T_j, since they have the same expectation value.

A.4 Quark Operators as Representations of the Chiral Group

Irreducible representations of SU(3) are classified according to its *highest weights* as explained in Theorem 5.9 of [49]. An irreducible representation π with highest weights (m_1, m_2) with m_1 and m_2 being non-negative integers is of dimension

$$\dim(\pi) = \frac{1}{2}(m_1 + 1)(m_2 + 1)(m_1 + m_2 + 2). \tag{254}$$

In physics m_1 is the number of quarks and m_2 is the number of anti-quarks due to the fact that quarks live in the fundamental, or the $(1, 0)$, representation of flavor SU(3) group while anti-

[26] If h is longer than the autocorrelation time of the random variables it seems the requirement that the random variables be *independent and identically distributed* can be skipped.

quarks live in the dual of the fundamental, or the $(0,1)$, representation of SU(3). Quark field combinations generally form reducible representations. For example $\bar{q}_1\gamma_\mu q_2$ forms a dimension 9 reducible representation of $(1,0)\otimes(0,1)$ which can be reduced to $(1,1)\oplus(0,0)$. We say an operator, or a quark field combination, transforms under the representation with the highest dimension after reduction, thus $\bar{q}_1\gamma_\mu q_2$ transforms under the $(1,1)$ representation, or simply using dimension number, the **8** representation of SU(3). Several other examples are[27]:

$$(\bullet\Gamma q_1)(\bullet\Gamma q_1) \to (1,0)\otimes(1,0) = (2,0)\oplus(0,1) = \mathbf{6}\oplus\bar{\mathbf{3}} \to \mathbf{6}. \tag{255}$$

$$(\bar{q}_1\Gamma q_2)(\bar{q}_1\Gamma q_2) \to (1,0)\otimes(0,1)\otimes(1,0)\otimes(0,1) = \mathbf{27}\oplus\mathbf{10}\oplus\overline{\mathbf{10}}\oplus\mathbf{8}\oplus\mathbf{8}\oplus\mathbf{8}\oplus\mathbf{8}\oplus\mathbf{1}\oplus\mathbf{1} \to \mathbf{27}. \tag{256}$$

To summarize the underlying representation of an operator is determined by counting the number of quarks and anti-quarks.

Under $\mathrm{SU}(3)_L \otimes \mathrm{SU}(3)_R$ we have two copies of SU(3) representation tensor product together. The $VV+AA$ operator in (127) is decomposed into left and right hand components:

$$\mathcal{O}_{VV+AA} = 2\cdot\Big[(\bar{s}_L\gamma_\mu d_L)(\bar{s}_L\gamma_\mu d_L) + (\bar{s}_R\gamma_\mu d_R)(\bar{s}_R\gamma_\mu d_R)\Big]. \tag{257}$$

It is a combination of $(\mathbf{27},\mathbf{1})$ and $(\mathbf{1},\mathbf{27})$ representations. Here the first number is the representation number for the left handed quarks and second number is that for the right handed quarks. Similarly

$$\mathcal{O}_{VV-AA} = 2\cdot\Big[(\bar{s}_L\gamma_\mu d_L)(\bar{s}_R\gamma_\mu d_R) + (\bar{s}_R\gamma_\mu d_R)(\bar{s}_L\gamma_\mu d_L)\Big], \tag{258}$$

$$\mathcal{O}_{SS-PP} = 2\cdot\Big[(\bar{s}_L d_R)(\bar{s}_R d_L) + (\bar{s}_R d_L)(\bar{s}_L d_R)\Big]. \tag{259}$$

By counting the number of quarks and anti-quarks they both transform under the $(\mathbf{8},\mathbf{8})$ representation. Finally

$$\mathcal{O}_{TT} = \Big[(\bar{s}_L\gamma_\mu d_R)(\bar{s}_L\gamma_\mu d_R) + (\bar{s}_R\gamma_\mu d_L)(\bar{s}_R\gamma_\mu d_L)\Big], \tag{260}$$

$$\mathcal{O}_{SS+PP} = 2\cdot\Big[(\bar{s}_L d_R)(\bar{s}_L d_R) + (\bar{s}_R d_L)(\bar{s}_R d_L)\Big] \tag{261}$$

and they are combinations of the $(\bar{\mathbf{6}},\mathbf{6})$ and $(\mathbf{6},\bar{\mathbf{6}})$ representation.

[27] $(0,2)\equiv\bar{\mathbf{6}}$ to dinstinguish it from $(2,0)$. Similarly $(0,1)\equiv\bar{\mathbf{3}}$, $(0,3)\equiv\overline{\mathbf{10}}$.