Reducing Residual Mass of Domain-Wall Fermions using Machine Learning

Shunsuke Yasunaga[1,2]
Kenta Yoshimura[1,2], Akio Tomiya[2,3,4], and Yuki Nagai[5]
[1]Institute of Science Tokyo
[2]RIKEN R-CCS
[3]Tokyo Woman's Christian University
[4]Kyoto University
[5]The University of Tokyo

The 43rd International Symposium on Lattice Field Theory (Lattice 2026)
University of Maryland, 2026/7/27

Research overview

Scope of this talk

This talk is about **chiral symmetry in lattice QCD**:
domain-wall fermions are used as chirally symmetric discretization for QCD.

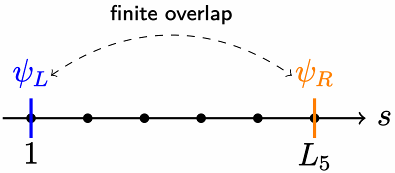
Main issue

At finite fifth-dimensional extent L_5, the overlap fermion is approximated, and a small residual chiral-symmetry violation remains.

We set the fifth-dimensional parameters with machine learning.

Domain-wall fermions at finite fifth dimension

Domain-wall fermions introduce an auxiliary fifth dimension.



- Left- and right-handed modes are localized on opposite boundaries.
- In the limit $L_5 \to \infty$, the two boundaries decouple.
- At finite L_5, the modes slightly mix.

Can we reduce violation of chiral symmetry at fixed L_5 without substantially increasing computational cost?

Domain-wall to overlap fermion

At finite L_5, the effective four-dimensional boundary operator is

$$D_4 = \frac{1+m_q}{2} + \frac{1-m_q}{2}\gamma_5\,\varepsilon_{L_5}(\{H_s\}),$$

where the finite-L_5 sign-function approximation is

$$\varepsilon_{L_5}(\{H_s\}) = \frac{1 - \prod_{s=1}^{L_5} T_s(H_s)}{1 + \prod_{s=1}^{L_5} T_s(H_s)}, \qquad T_s(H_s) = \frac{1-H_s}{1+H_s}, \qquad H_s = \gamma_5 \frac{(b_s + c_s)D_{\text{Wilson}}}{2 + (b_s - c_s)D_{\text{Wilson}}}.$$

$$L_5 \to \infty: \qquad \varepsilon_{L_5}(H) \to \operatorname{sgn}(H), \qquad D_4 \to D_{\text{ov}}.$$

Parameters b_s, c_s tune the chiral symmetry.

Previous choices of b_s and c_s

$$H_s = \gamma_5 \frac{(b_s + c_s) D_{\text{Wilson}}}{2 + (b_s - c_s) D_{\text{Wilson}}}.$$

Shamir

$$b_s = 1, \quad c_s = 0$$

Fixed coefficients.

Optimal DWF

$$b_s + c_s = \omega_s$$

ω_s is determined by Zolotarev approx.

Möbius DWF

$$b_s = b, \quad c_s = c$$

are independent on s.

(More generally, s-dependent coefficients are allowed.)

This work

$\{b_s, c_s\}$ are

trainable by machine learning.

Shamir, Nucl. Phys. B 406, 90 (1993).
Chen and Chiu, Phys. Rev. D 86, 094508 (2012).
Brower, Neff, and Orginos, Comput. Phys. Commun. 220, 1 (2017).

Machine learning as parameter optimization

We optimize parameters $\boldsymbol{\theta} = \{b_1, \dots, b_{L_5}, c_1, \dots, c_{L_5}\}$ by machine learning

Present method: gradient-based optimization

Starting from an initial parameter set, we update

$$\boldsymbol{\theta}^{(n+1)} = \boldsymbol{\theta}^{(n)} - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}.$$

- The gradient is calculated from the domain-wall operator.
- We use Adam to stabilize the stochastic updates.
- No neural network is used at this stage.

Loss function: residual mass

The training target is the residual mass evaluated with noise vectors,

$$m_{\text{res}}(\boldsymbol{\theta}) = \frac{\mathrm{Re}\langle \mathrm{Tr}\, D_4^{-1}(\boldsymbol{\theta})\rangle}{\langle \mathrm{Tr}(D_4^\dagger D_4(\boldsymbol{\theta}))^{-1}\rangle} - m_q.$$

Y.-C. Chen and T.-W. Chiu, Phys. Rev. D **86**, 094508 (2012).

The traces are estimated stochastically,

$$\mathrm{Tr}\,\mathcal{O} \simeq \frac{1}{N_\eta}\sum_{k=1}^{N_\eta} \eta_k^\dagger \mathcal{O} \eta_k.$$

Loss function

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2}\left[m_{\text{res}}(\boldsymbol{\theta})\right]^2.$$

Gradient of the loss function

For $r_s = b_s$ or c_s,

$$\frac{\partial \mathcal{L}}{\partial r_s} = m_{\mathrm{res}} \frac{\partial m_{\mathrm{res}}}{\partial r_s}.$$

The required inverse-operator derivative is

$$\frac{\partial D_5^{-1}}{\partial r_s} = -D_5^{-1} \frac{\partial D_5}{\partial r_s} D_5^{-1}.$$

For the fifth-dimensional coefficients,

$$\frac{\partial (D_5)_{t,u}}{\partial b_s} = \delta_{s,t}\delta_{t,u} D_{\mathsf{Wilson}}, \qquad \frac{\partial (D_5)_{t,u}}{\partial c_s} = \delta_{s,t} F_{t,u} D_{\mathsf{Wilson}}.$$

Optimization workflow

1. Prepare gauge configuration.
2. Perform one standard HMC trajectory.
3. Estimate gradient of the loss function.
4. Update $\{b_s, c_s\}$ using Adam.
5. Repeat 2 – 4.
6. Measure the plateau value of $R(t)$:

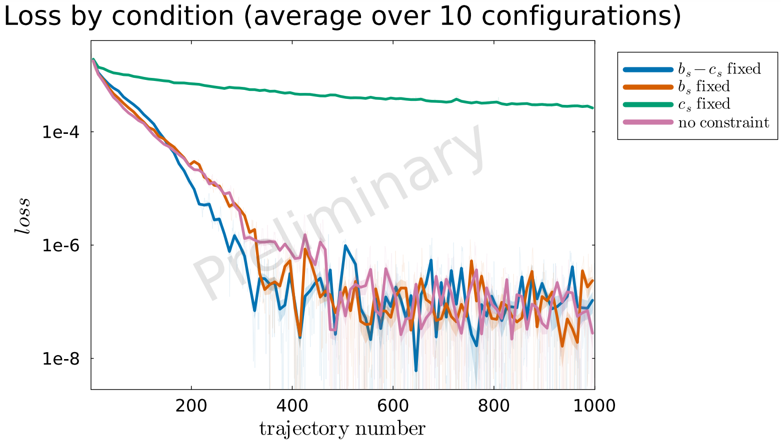
$$R(t) = \frac{\sum_{\boldsymbol{x}} \langle J_{\text{mid}}(\boldsymbol{x}, t) J_{\text{edge}}(0) \rangle}{\sum_{\boldsymbol{x}} \langle J_{\text{edge}}(\boldsymbol{x}, t) J_{\text{edge}}(0) \rangle}.$$

S. Aoki et al., Phys. Rev. D **62**, 094502 (2000).

Numerical setup

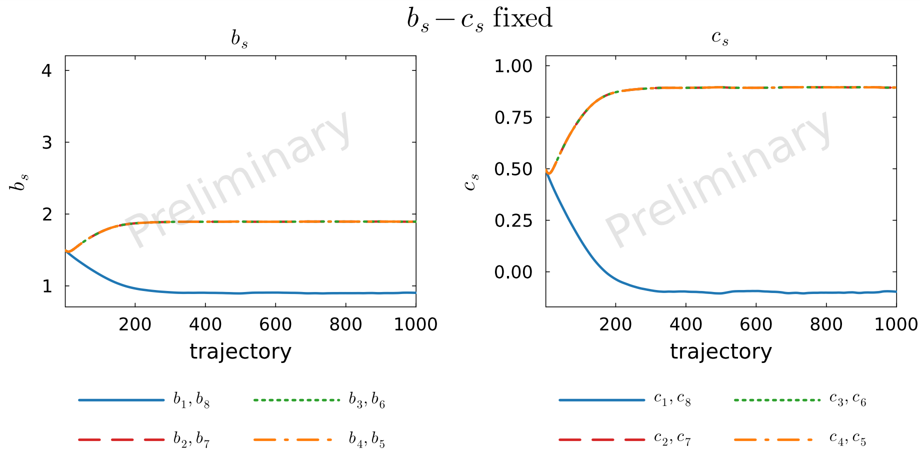
- Lattice: 16^4, $L_5 = 8$
- Number of flavors: $N_f = 2$
- Coupling: $\beta = 5.7$
- Bare mass: $m_q a = 0.1$
- Domain-wall height: $M_5 = 1.5$
- Lattice spacing: $a = 0.1457(6)$ fm
- Initial value: $b_s = 1.5$, $c_s = 0.5$
- `JuliaQCD`
  `https://github.com/JuliaQCD`

Loss history: noise-vector optimization

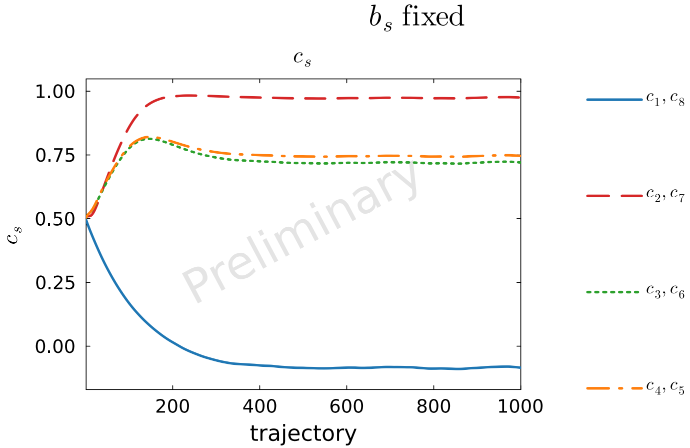


All cases except c_s fixed reach a common stochastic noise floor, while c_s fixed converges much more slowly.

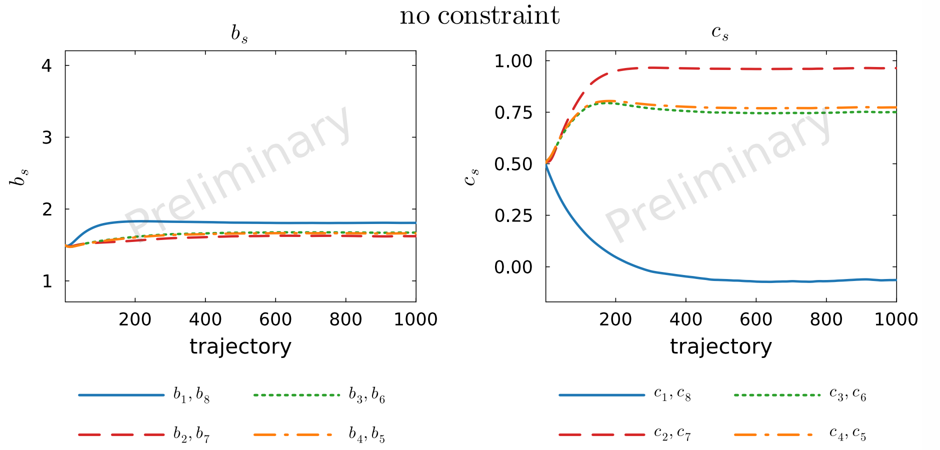
History of coefficients



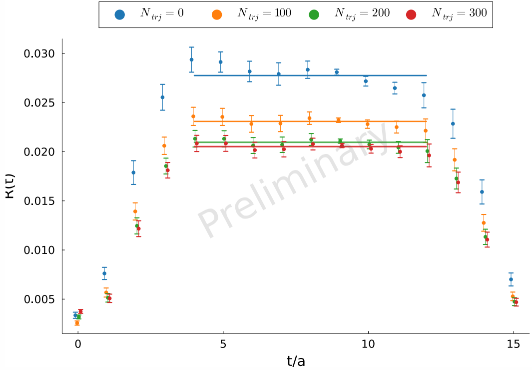
History of coefficients



History of coefficients



Plateau measurement during optimization



Plateau values during the $b_s - c_s$ fixed optimization.

The residual mass is extracted from the axial Ward–Takahashi identity:

$$R(t) = \frac{\sum_{\boldsymbol{x}} \langle J_{\text{mid}}(\boldsymbol{x}, t) J_{\text{edge}}(\mathbf{0}, 0) \rangle}{\sum_{\boldsymbol{x}} \langle J_{\text{edge}}(\boldsymbol{x}, t) J_{\text{edge}}(\mathbf{0}, 0) \rangle}.$$

Evaluation points

$$N_{\text{trj}} = 0,\ 100,\ 200,\ 300$$

Summary and outlook

Goal: Reduce residual chiral-symmetry breaking at fixed L_5.

Method: Optimize b_s, c_s using a stochastic estimate of m_{res} and Adam updates.

Result: The optimized parameters reduce the plateau value of $R(t)$.

Outlook: Jointly optimize chiral symmetry, solver iterations, and computational cost.

Toward domain-wall fermions with improved chiral symmetry at controlled computational cost.

Backup: Code implementation

JuliaQCD

https://github.com/JuliaQCD

Bridge++

https://bridge.kek.jp/Lattice-code/

Implemented features

- Generalized domain-wall operator
- Global residual-mass (JuliaQCD)
- Plateau residual-mass (Bridge++)
- Adam update with `Optimisers.jl`

Computing environment

- Current test size: $16^4 \times 8$
- GPU calculation on Miyabi-G

  (at JCAHPC, The University of Tokyo)
- Heian computer

  (at YITP, Kyoto University)

Backup: Fifth dimensional matrix

$$S_5 = \sum_{s,t} \bar{\Psi}_s \left(D_5\right)_{st} \Psi_t$$

$$(D_5)_{st} = \begin{pmatrix} D_+^{(1)} & D_-^{(1)} P_- & 0 & \cdots & 0 & 0 & -m_q D_-^{(1)} P_+ \\ D_-^{(2)} P_+ & D_+^{(2)} & D_-^{(2)} P_- & \cdots & 0 & 0 & 0 \\ 0 & D_-^{(3)} P_+ & D_+^{(3)} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & D_+^{(L_5-2)} & D_-^{(L_5-2)} P_- & 0 \\ 0 & 0 & 0 & \cdots & D_-^{(L_5-1)} P_+ & D_+^{(L_5-1)} & D_-^{(L_5-1)} P_- \\ -m_q D_-^{(L_5)} P_- & 0 & 0 & \cdots & 0 & D_-^{(L_5)} P_+ & D_+^{(L_5)} \end{pmatrix}_{st}$$

$$D_+^{(s)} = b_s D_{\mathsf{Wilson}}(-M) + 1$$

$$D_-^{(s)} = c_s D_{\mathsf{Wilson}}(-M) - 1$$

Backup: Is fifth-dimensional reflection symmetry mandatory?

For the rational approximation itself

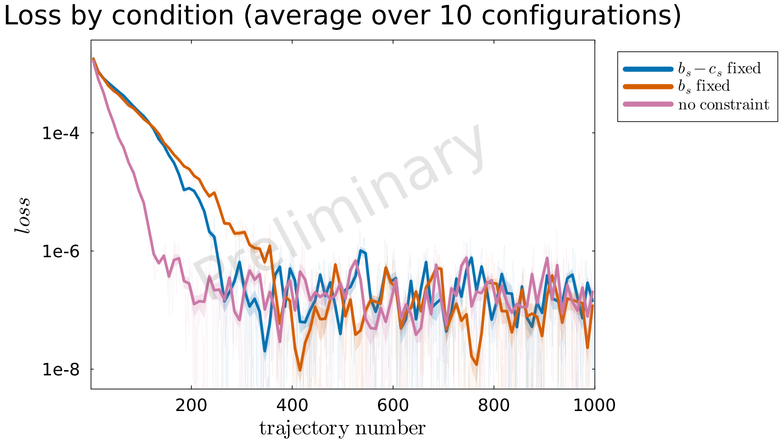
A product of transfer matrices can be defined without imposing $b_s = b_{L_5+1-s}$ and $c_s = c_{L_5+1-s}$.

For the standard dynamical five-dimensional formulation

Reflection symmetry is a practical structural constraint because it preserves the usual $\gamma_5 R_5$-Hermiticity and the corresponding convenient HMC formulation.

In the present optimization, we therefore impose the symmetry rather than treating all layers as independent.

Backup: Loss history without reflection symmetry



The c_s-fixed case is not shown because the HMC became unstable.