Gauge-equivariant multigrid neural networks for lattice QCD

based on Lehner, Wettig 2302.05419, 2304.10438

Christoph Lehner

(University of Regensburg)

May 15, 2023 – HU Berlin / NIC DESY Zeuthen joint lattice seminar

Preconditioning

- In lattice QCD, wall-clock time is often dominated by solution of Dirac equation

$$Du = b$$

  Usually done by an iterative solver (here, GMRES)

- Time to solution is determined by condition number of Dirac matrix
  - Condition number increases dramatically in physical quark-mass and continuum limit

- Can be addressed by Preconditioning
  - Find a preconditioner M such that $M \approx D^{-1}$
  - Define $v = M^{-1}u$ and use

$$DMM^{-1}u = (DM)v = b$$

    to solve for v with preconditioned matrix DM (smaller condition number)
  - Then $u = Mv$

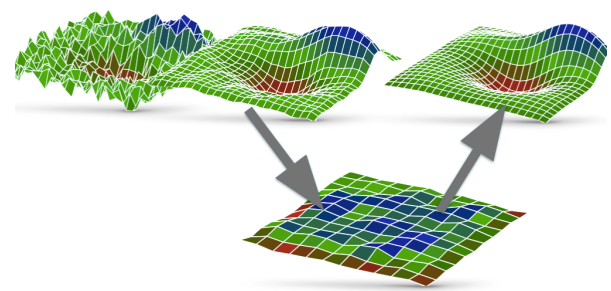
Low and high modes

- Consider the eigendecomposition of D

$$D = \sum_n \lambda_n |n\rangle\langle n|$$

  Preconditioner should approximate low-mode and high-mode components of D^{-1}

- State-of-the-art algorithms (multigrid) are designed to do this



- We will follow this paradigm, but here we learn the preconditioner

Source: `https://summerofhpc.prace-ri.eu/multithreading-the-multigrid-solver-for-lattice-qcd`

Gauge-equivariant layers

Parallel transport

- Consider a field $\phi(x)$ with $x \in S$ (space-time lattice, dim $= d$) and $\phi \in V_I = V_G \otimes V_{\bar{G}}$
  (gauge space: $V_G = \mathbb{C}^N$, non-gauge space: $V_{\bar{G}} = \mathbb{C}^{\bar{N}}$)

- Also consider an SU(N) gauge field $U_\mu(x)$ acting on V_G

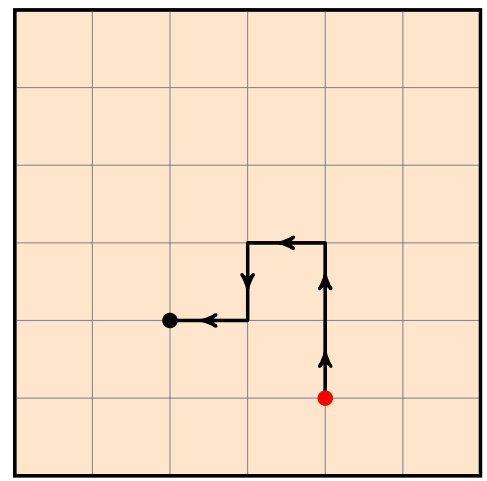
- Define the parallel-transport operator for a path $p = p_1, \ldots, p_{n_p}$ with $p_i \in \{\pm 1, \ldots, \pm d\}$

$$T_p = H_{p_{n_p}} \cdots H_{p_2} H_{p_1} \qquad \text{with} \qquad H_\mu \phi(x) = U_\mu^\dagger(x - \hat{\mu})\phi(x - \hat{\mu})$$

- H_μ transports information by a single hop in direction $\hat{\mu}$

- H_μ acts on field; new field $H_\mu \phi$ is evaluated at x

- Example: $T_p = H_{-1} H_{-2} H_{-1} H_2 H_2$



Gauge equivariance

- A gauge transformation by $\Omega(x) \in \mathrm{SU}(N)$ acts in the usual way

$$\phi(x) \to \Omega(x)\phi(x)$$
$$U_\mu(x) \to \Omega(x)U_\mu(x)\Omega^\dagger(x+\hat{\mu})$$

- Such gauge transformations commute with T_p for any path p

$$T_p\phi(x) \to \Omega(x)\,T_p\phi(x)$$

- This is an example of gauge equivariance:

An object (here: ϕ) and the transformed object (here: $T_p\phi$) transform in the same way under a gauge transformation.

Parallel-transport convolutions

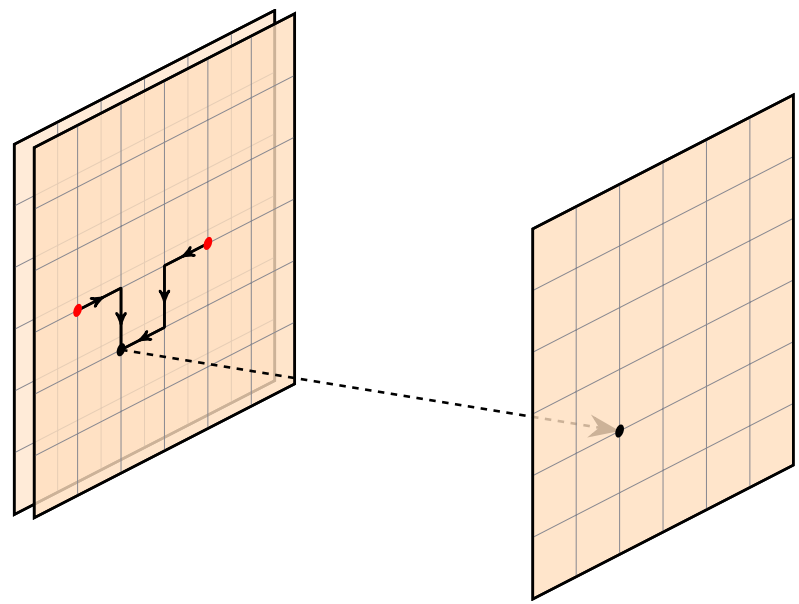
- Parallel-transport convolution layer and local parallel-transport convolution layer

$$\psi_a(x) \overset{\text{PTC}}{=} \sum_b \sum_{p \in P} W_a^{bp} T_p \phi_b(x) \qquad \psi_a(x) \overset{\text{LPTC}}{=} \sum_b \sum_{p \in P} W_a^{bp}(x) T_p \phi_b(x)$$

  - $a =$ output feature index
  - $b =$ input feature index
  - $P =$ set of paths
  - W_a^{bp} acts in $V_{\bar{G}}$ (here: 4×4 spin matrix)
  - Elements of W: "layer weights"

- Layers are gauge-equivariant

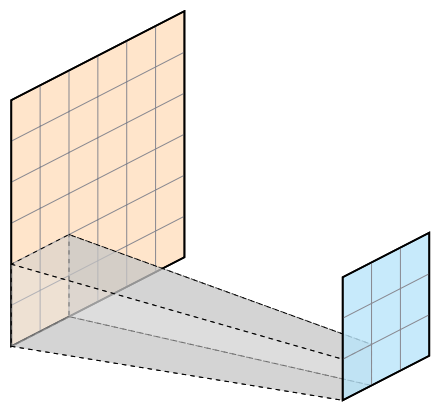
- No activation function since we want to learn a linear preconditioner

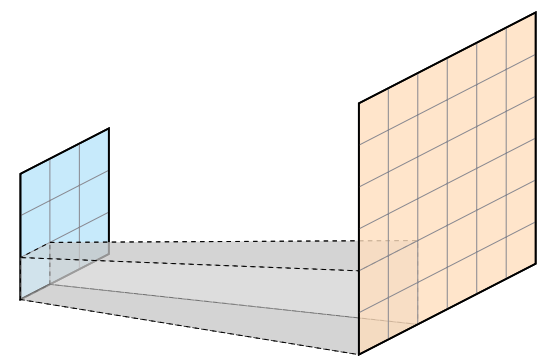


Restriction and prolongation layers

- On the coarse grid $\tilde{S}$ we have fields $\tilde{\phi}(y)$ with $y \in \tilde{S}$ and $\tilde{\phi} \in \tilde{V}_l$
- $\tilde{V}_l$ has no gauge degrees of freedom $\rightarrow$ no gauge transformations on $\tilde{V}_l$
- Restriction and prolongation layer (with $B =$ block map from $\tilde{S}$ to S)

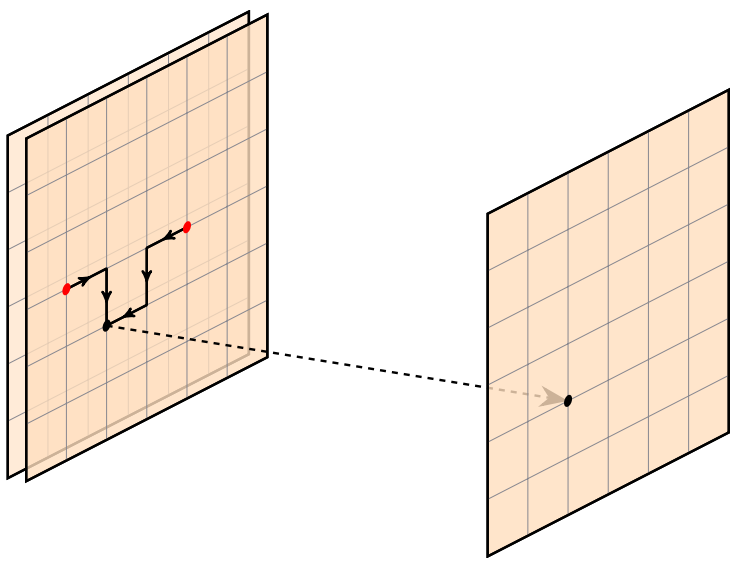
$$\tilde{\psi}(y) \overset{\mathrm{RL}}{=} \sum_{x \in B(y)} W(y,x)\phi(x) \qquad \psi(x) \overset{\mathrm{PL}}{=} W(y,x)^{\dagger}\tilde{\phi}(y)$$

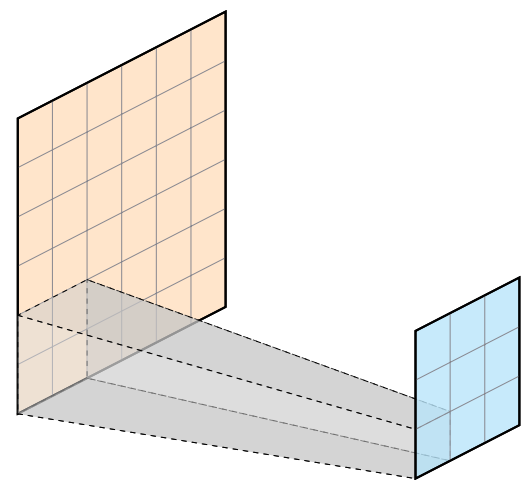




Graphical conventions for features and layers

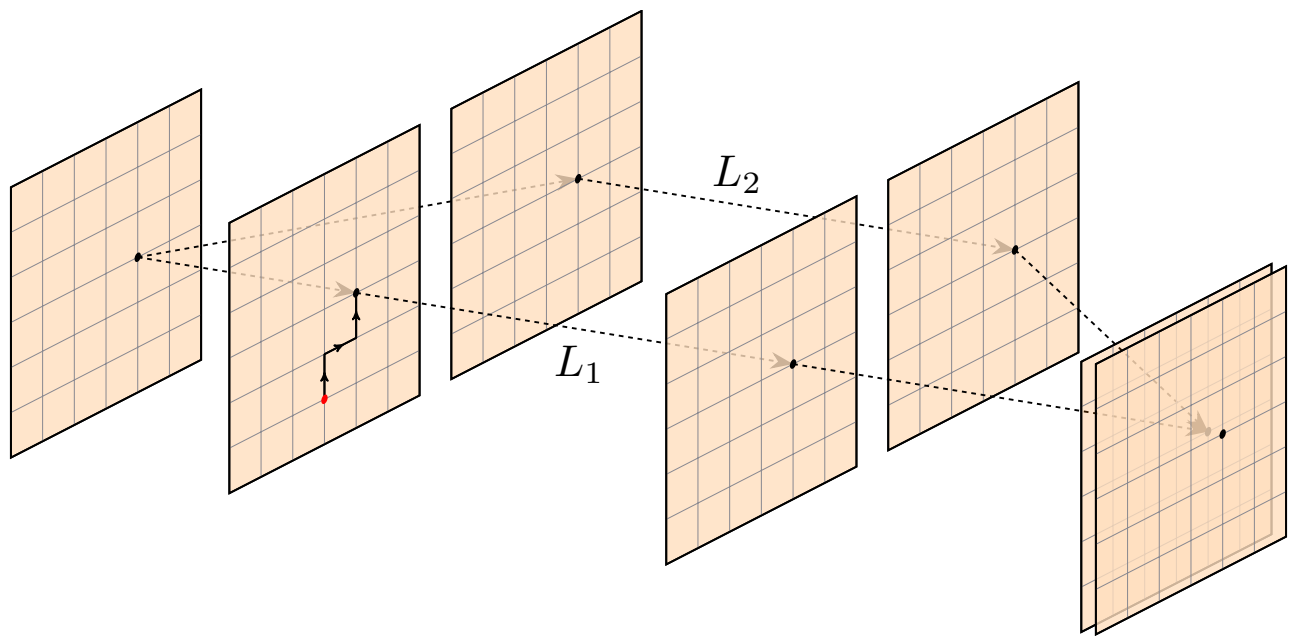
- A feature is represented by a plane
- A layer sits between planes and is represented by paths plus a dashed arrow
- For restriction/prolongation, a layer is represented by a square frustum





Parallel and identity layers

- Parallel layers act on the same input feature in parallel
- Identity layer: simple copy operation, i.e., output $=$ input
- Example: (All layers except L_1 are identity layers)



Communication avoidance

- On machines with many nodes, sub-volumes are assigned to different MPI processes

- We also consider models where no information is communicated between sub-volumes (by setting the links $U_\mu(x)$ connecting sub-volumes to zero)

- We find that the performance of these models is close to those with communication
  $\rightarrow$ Overall wall-clock time could be lower since no time is spent on communication

Wilson-clover Dirac operator

Dirac operator

- The Wilson Dirac operator can be written in terms of single hops:

$$D_{\mathrm{W}} = \frac{1}{2}\sum_{\mu=1}^{4}\gamma_\mu(H_{-\mu} - H_{+\mu}) - \frac{1}{2}\sum_{\mu=1}^{4}(H_{-\mu} + H_{+\mu} - 2) + m$$

- For Wilson-clover, consider closed paths with four hops and define

$$\begin{aligned} Q_{\mu\nu} = {} & H_{-\mu}H_{-\nu}H_{+\mu}H_{+\nu} + H_{-\nu}H_{+\mu}H_{+\nu}H_{-\mu} \\ & + H_{+\nu}H_{-\mu}H_{-\nu}H_{+\mu} + H_{+\mu}H_{+\nu}H_{-\mu}H_{-\nu}\,. \end{aligned}$$

  Then

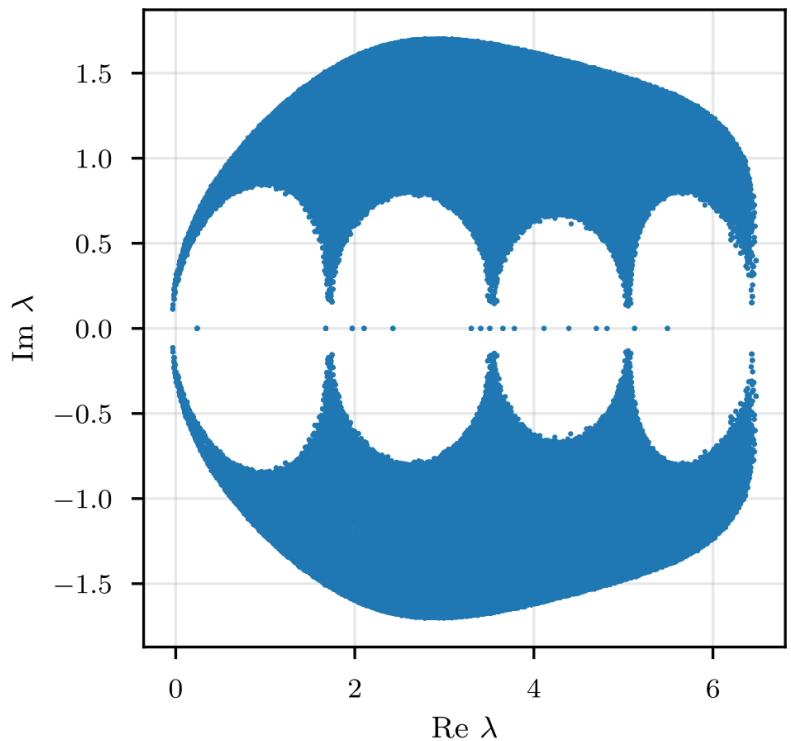
$$D_{\mathrm{WC}} = D_{\mathrm{W}} - \frac{c_{\mathrm{sw}}}{4}\sum_{\mu,\nu=1}^{4}\sigma_{\mu\nu}F_{\mu\nu}$$

  with

$$F_{\mu\nu} = \frac{1}{8}(Q_{\mu\nu} - Q_{\nu\mu}) \qquad \sigma_{\mu\nu} = \frac{1}{2}(\gamma_\mu\gamma_\nu - \gamma_\nu\gamma_\mu)$$

Numerical details and eigenvalue spectrum

- $V = 8^3 \times 16$, $\beta = 6.0$ (pure gauge), $c_{\mathrm{SW}} = 1$, periodic boundary conditions for all fields
- $m = -0.6$ is chosen so that D_{WC} is tuned to near criticality (i.e., real part of smallest eigenvalue ≈ 0) $\rightarrow$ solution of $Du = b$ is challenging problem



High-mode preconditioners

Model setup and training strategy

- High-mode part of Dirac spectrum is related to short-distance behavior
  $\rightarrow$ Expect one or two layers with small number of hops to show gain in iteration count
- Consider a linear model M mapping a vector x to Mx
- Supervised learning approach with training step as follows:
  - Pick random vector v from Gaussian distribution (mean zero, standard deviation 1)
  - Compute training tuple $(D_{\mathrm{WC}} v, v)$ and optimize cost function

$$C = |M D_{\mathrm{WC}} v - v|^2$$

    $\rightarrow$ Model learns to map $D_{\mathrm{WC}} v$ to v (and hence $M \approx D_{\mathrm{WC}}^{-1}$)
  - Optimizer is Adam Kingma & Ba, arXiv:1412.6980 [cs.LG]
  - Derivatives w.r.t. model weights computed using backpropagation
- Training data set is unbounded in size $\rightarrow$ no need to add a regulator
- Cost function is dominated by high modes

Models chosen for high-mode preconditioner

- One layer, one hop (i.e., 9 paths)

$$T_0 = \mathbb{1}\,,\ T_1 = H_1\,,\ T_2 = H_2\,,\ T_3 = H_3\,,\ T_4 = H_4\,,$$
$$T_5 = H_{-1}\,,\ T_6 = H_{-2}\,,\ T_7 = H_{-3}\,,\ T_8 = H_{-4}$$

- One layer, two hops: extend the above by 56 two-hop paths

$$H_a H_b \qquad \text{with} \qquad a, b \in \{-4, -3, -2, -1, 1, 2, 3, 4\} \qquad (a \neq -b)$$

- "Deep" network of two one-hop layers:
  - $1 \to 1 \to 1$: Two successive layers with one hop each
  - $1 \to 2 \to 1$: Two output features in first layer, two input features in second layer

- PTC (layer weights constant) and LPTC (layer weights depend on x)

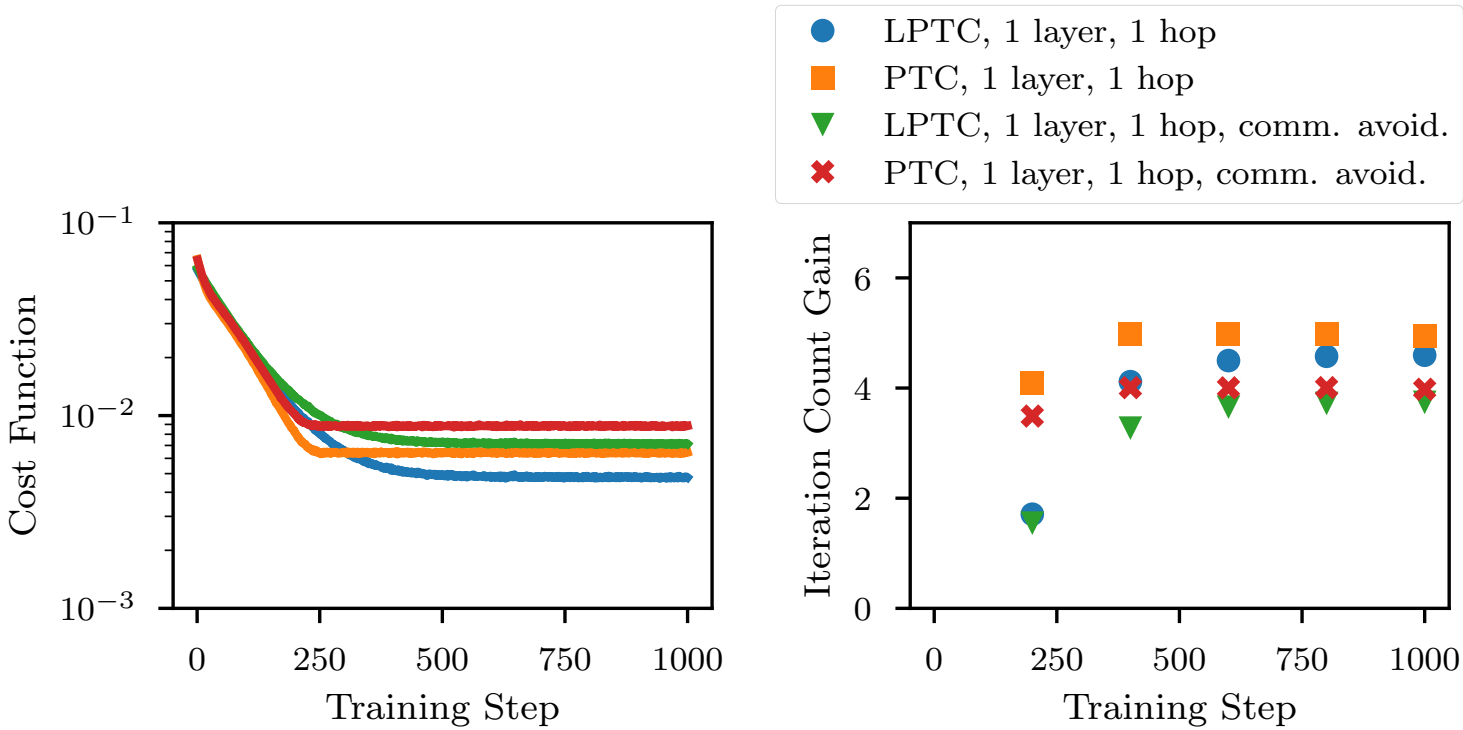
- Communication avoidance: $U_\mu(x) \equiv 0$ between subvolumes of size $4^3 \times 8$

Measure of performance

$$\text{Iteration count gain} = \frac{\text{Iteration count without preconditioner}}{\text{Iteration count with preconditioner}}$$

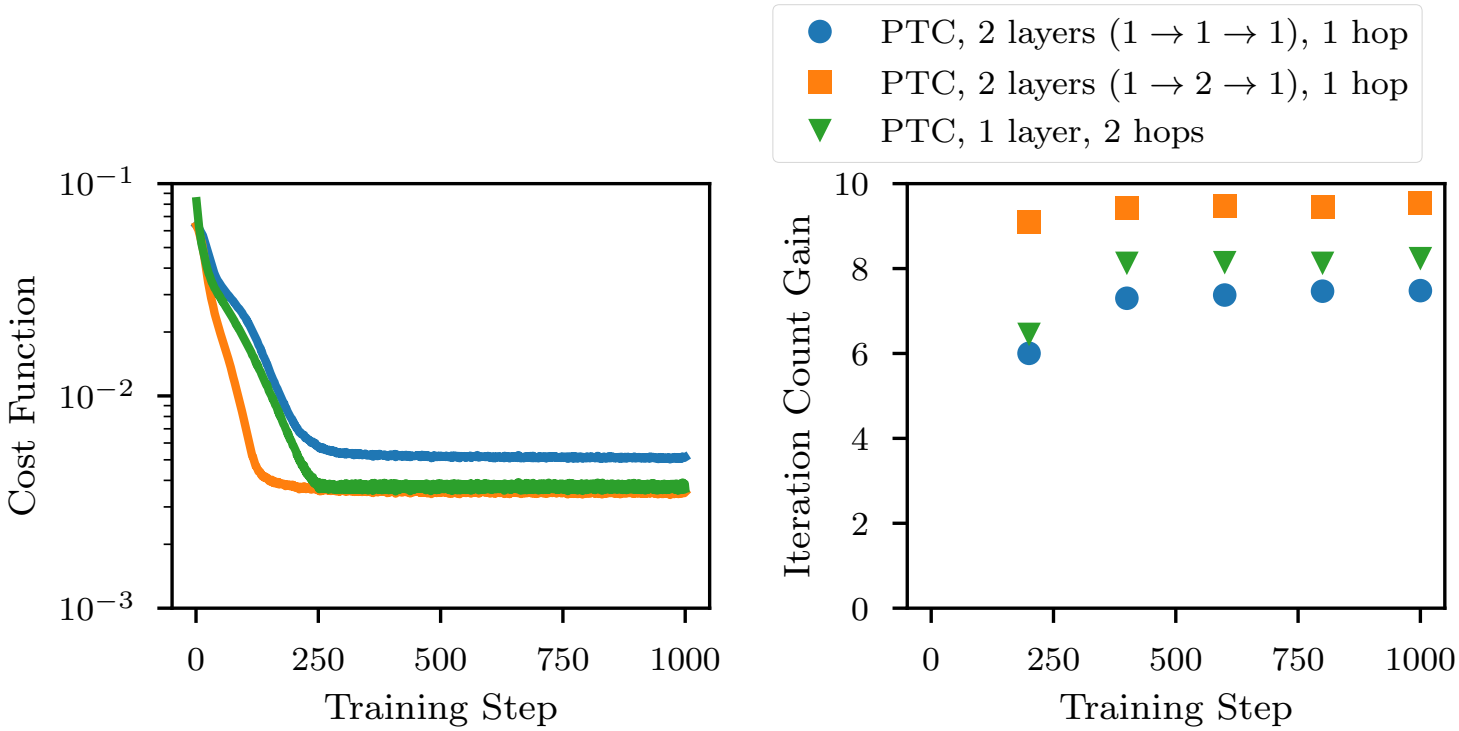
- Iteration count refers to outer solver (here, GMRES)

Results for high-mode preconditioner (one layer, one hop)



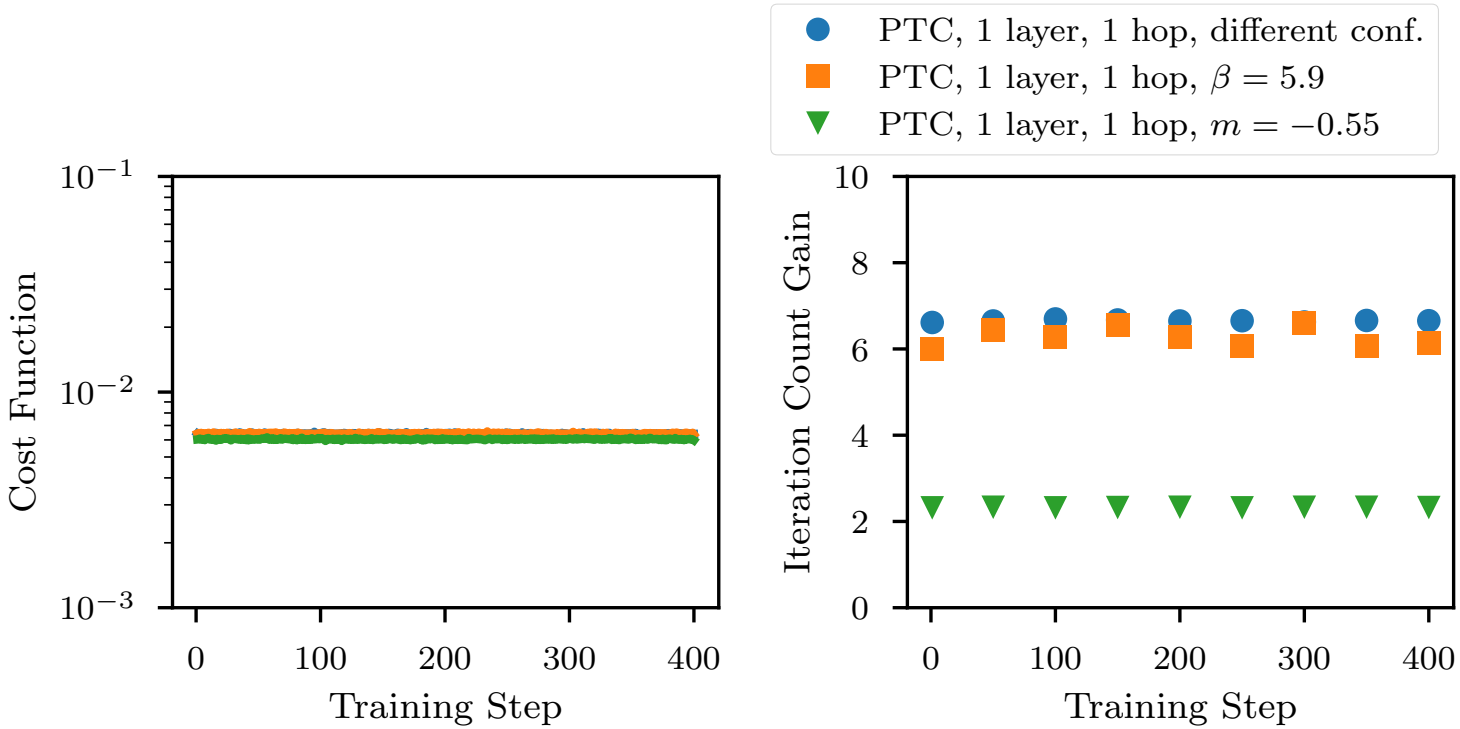
- No gain from LPTC (and they require more training)
- Communication-avoiding version only slightly worse (could be amortized)

Results for high-mode preconditioner (deep network/multiple hops)



- $1 \to 2 \to 1$ model performs best (and gives $\sim$ twice the gain of 1 layer/1 hop model)
- Since layers are linear, deep models are not more expressive than shallow models with same number of hops (but easier to train b/o smaller number of weights) $\to$ 2-hop model should reach similar performance with improved training procedure

Transfer learning



- No retraining required for (i) different configuration from same ensemble, (ii) configuration with different β, (iii) different mass
- Performance varies slightly between configurations
- $m = -0.55$ is not tuned to criticality $\rightarrow$ Easier initial problem $\rightarrow$ Smaller gain

Low-mode preconditioners

Possible approaches

- Low-mode part of Dirac spectrum is related to long-distance behavior
  $\rightarrow$ Need deep network of (L)PTC layers to propagate information over long distances

- Alternative: use multigrid paradigm

  - Define coarse version of the lattice

  - Define restriction and prolongation operations (= layers)

  - Preserve low-mode part of Dirac spectrum
    Lüscher arXiv:0706.2298 [hep-lat]

Model setup

- Reminder: coarse lattice $= \tilde{S}$, internal vector space $= \tilde{V}_l$ with $s = \dim(\tilde{V}_l)$
- Find s vectors in the near-null space of D

$$Du_i \approx 0 \qquad (i = 1, \ldots, s)$$

  - Apply GMRES for D with source vector $= 0$ and random initial guess (solve to 10^{-8})
  - This removes high-mode components and leaves linear combination of low-modes

- Block the u_i
  - One site $y \in \tilde{S}$ corresponds to a set of sites (or block) $B(y) \in S$
  - Blocked vector u_i^y lives on the sites of $B(y)$
- Orthonormalize the u_i^y within each block $\rightarrow \bar{u}_i^y$
- Then the prolongation map (see slide 6) is defined by

$$W(y,x)^\dagger = \sum_{i=1}^{s} \bar{u}_i^y(x)\hat{e}_i^\dagger$$

with $x \in B(y)$ and $\hat{e}_i$ the canonical unit vectors of $\tilde{V}_l$

Training strategy

- Coarse-grid operator is defined as

$$\tilde{D} = R D_{\mathrm{WC}} P$$

with R and P defined according to restriction and prolongation layers (slide 6)

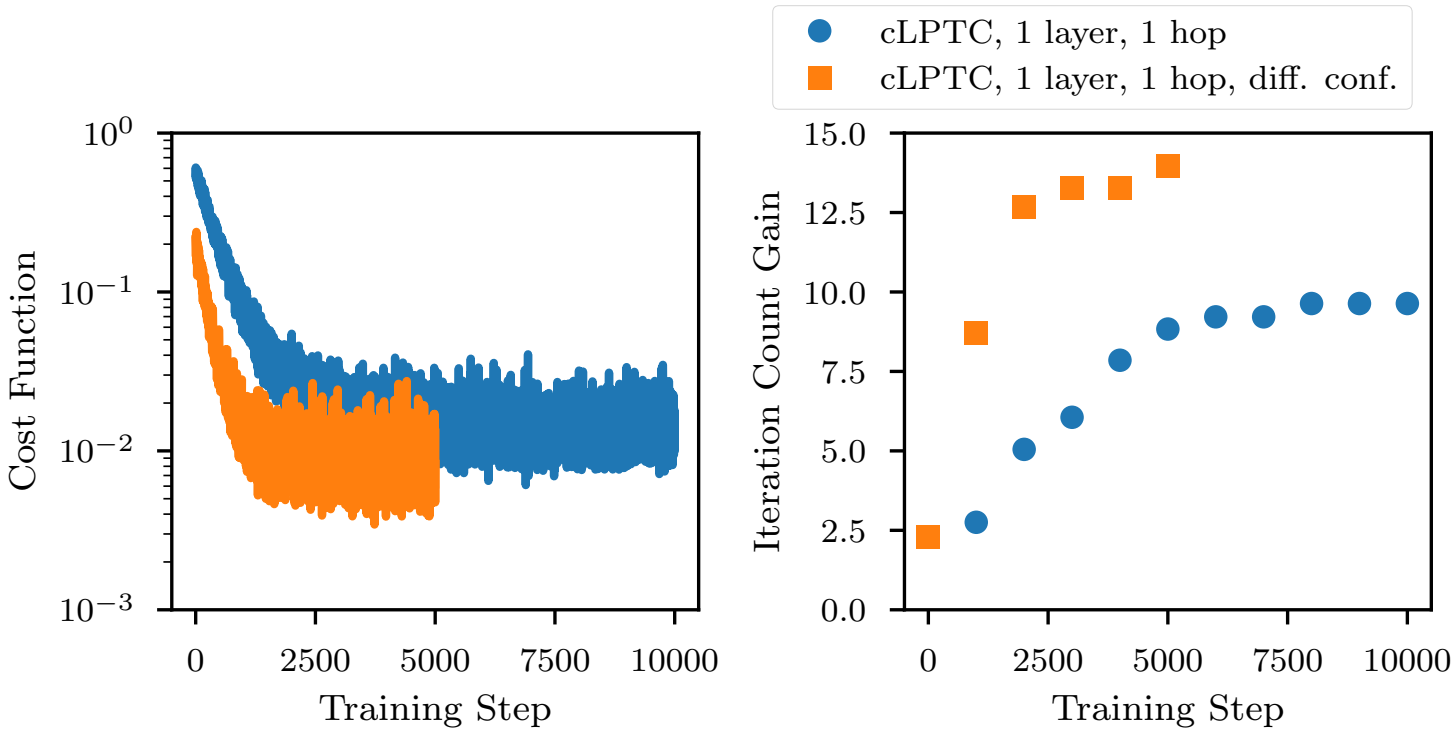
- Coarse-grid model $\tilde{M}$ contains single LPTC layer with zero- and one-hop paths and gauge fields replaced with $\mathbb{1}$ (layer is denoted by cLPTC)

- Same training strategy as before, with cost function

$$C = |\tilde{M}\tilde{D}v - v|^2$$

(this avoids costly inversion of $\tilde{D}^{-1}$)

Results for low-mode preconditioner (cLPTC layer)



- Iteration count gain refers to inversion of $\tilde{D}$ (we use $\tilde{S} = 2^3 \times 4$ and $s = 12$)
- Longer training period compared to high-mode preconditioner
- Transfer learning works with moderate retraining

Multi-grid preconditioners

Model setup

- Combine the high- and low-mode models to learn a model M that approximates the short- and long-distance features of D^{-1}
- First create a short-distance model that accepts a second input feature (initial guess)
  - Model plays role of smoother in multigrid paradigm
  - Initial guess from long-distance model acting on coarse grid

Smoother model setup and training strategy

- Find a sequence of u_k that approximately solve $Du = b$ (exact solution for $k \to \infty$)
  - Assume we have a high-mode model M_h that approximates D^{-1}
  - Smoother maps the tuple (u_k, b) to u_{k+1}

$$\begin{aligned} u_{k+1} &= (\mathbb{1} - M_\mathrm{h} D) u_k + M_\mathrm{h} b \\ &= u_k + M_\mathrm{h}(b - D u_k) \qquad (*) \end{aligned}$$

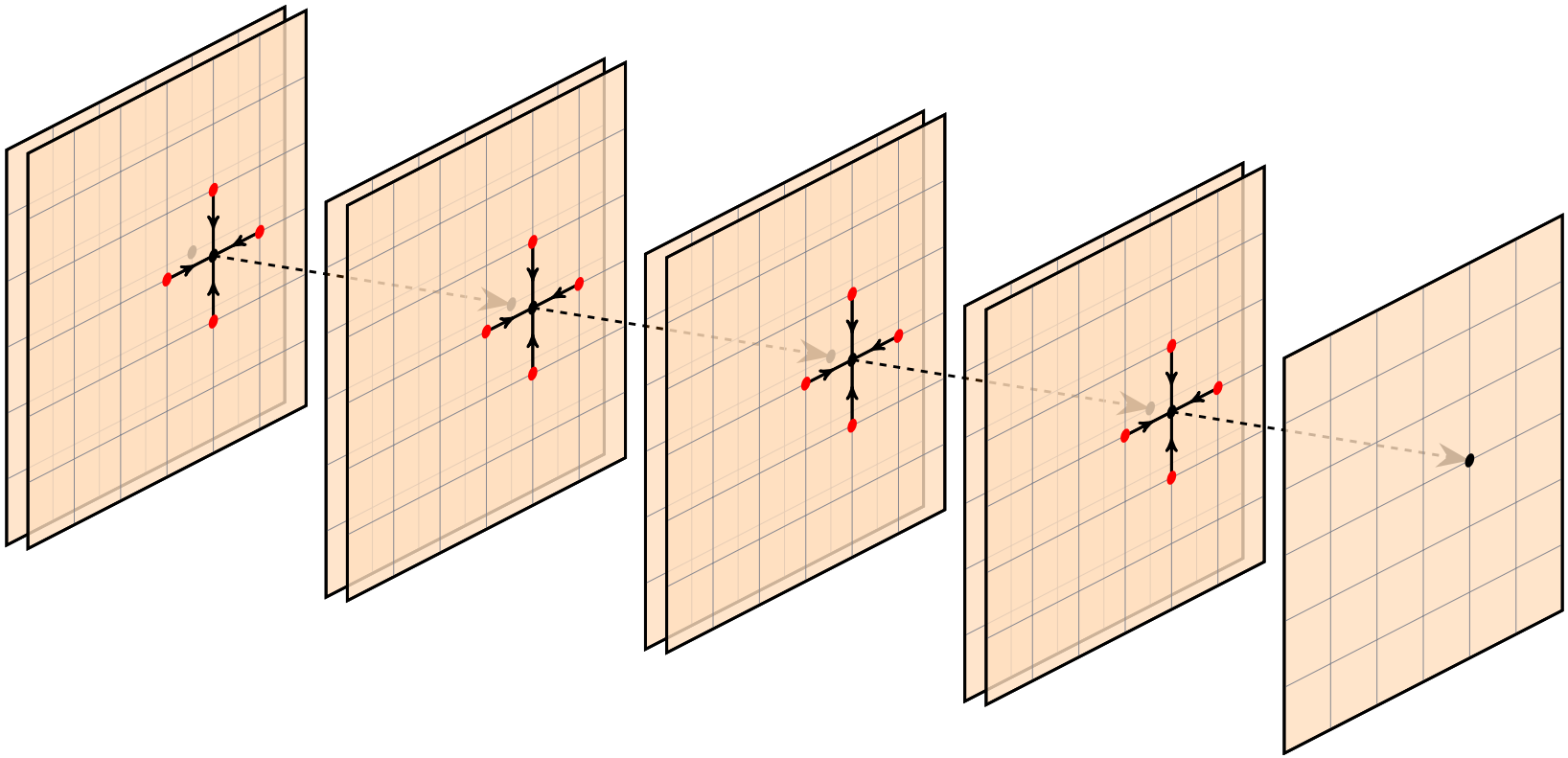
  ("iterative relaxation approach" or "defect correction" with defect $b - Du$)

- Both D and high-mode model M_h can be represented by (L)PTC layers
  $\to$ Train a model M_s to map (u_k, b) to a u_{k+r} (with $r \in \mathbb{N}^+$)
  - Model must have two input features and one output feature
  - Every iteration of $(*)$ corresponds to two (L)PTC layers
    $\to$ Construct M_s using $2r$ successive layers (with up to one hop each)
  - We use $r = 2$ since it performed better than $r = 1$ in full multigrid model

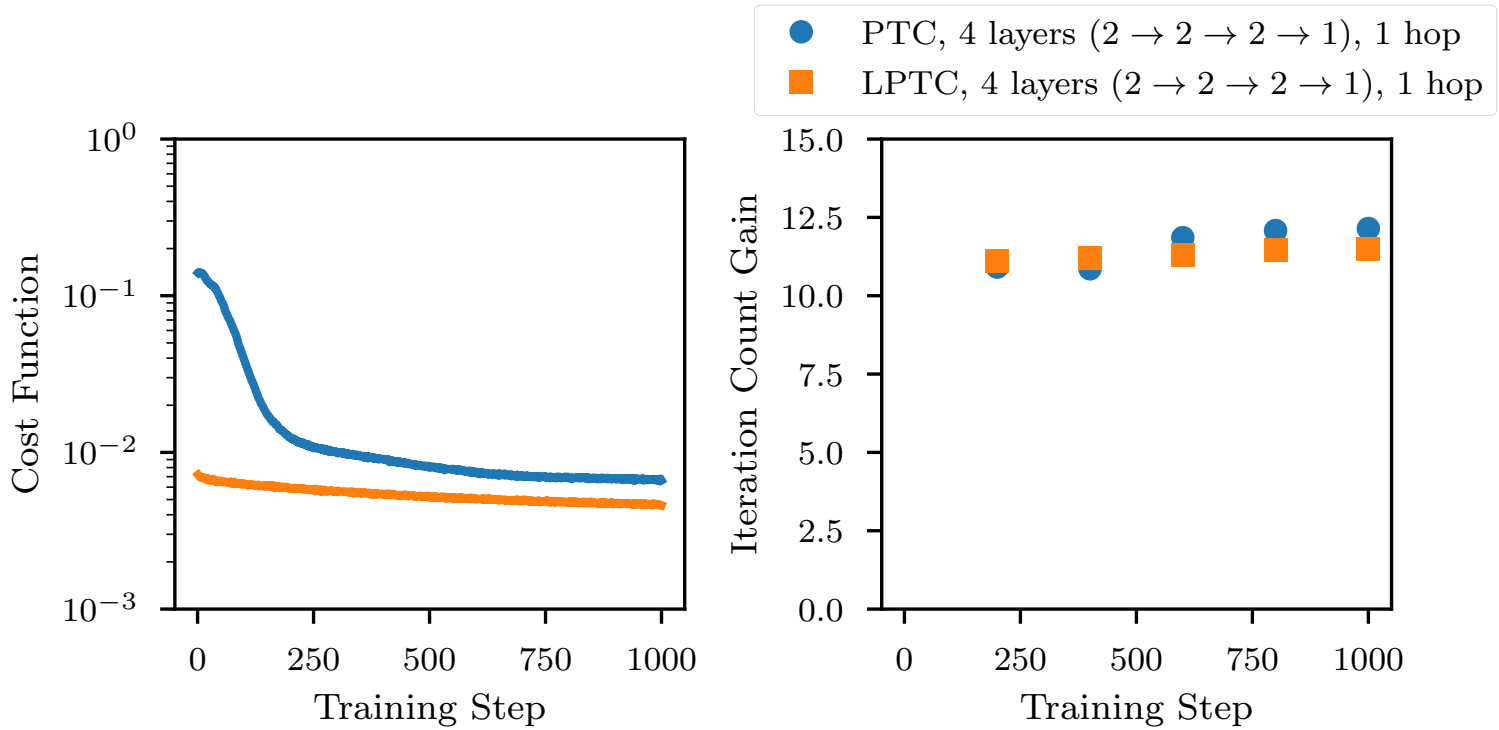
- Cost function (with random vectors u_k, b)

$$C = |M_\mathrm{s}(u_k, b) - u_{k+r}|^2$$

Smoother model



Results for smoother

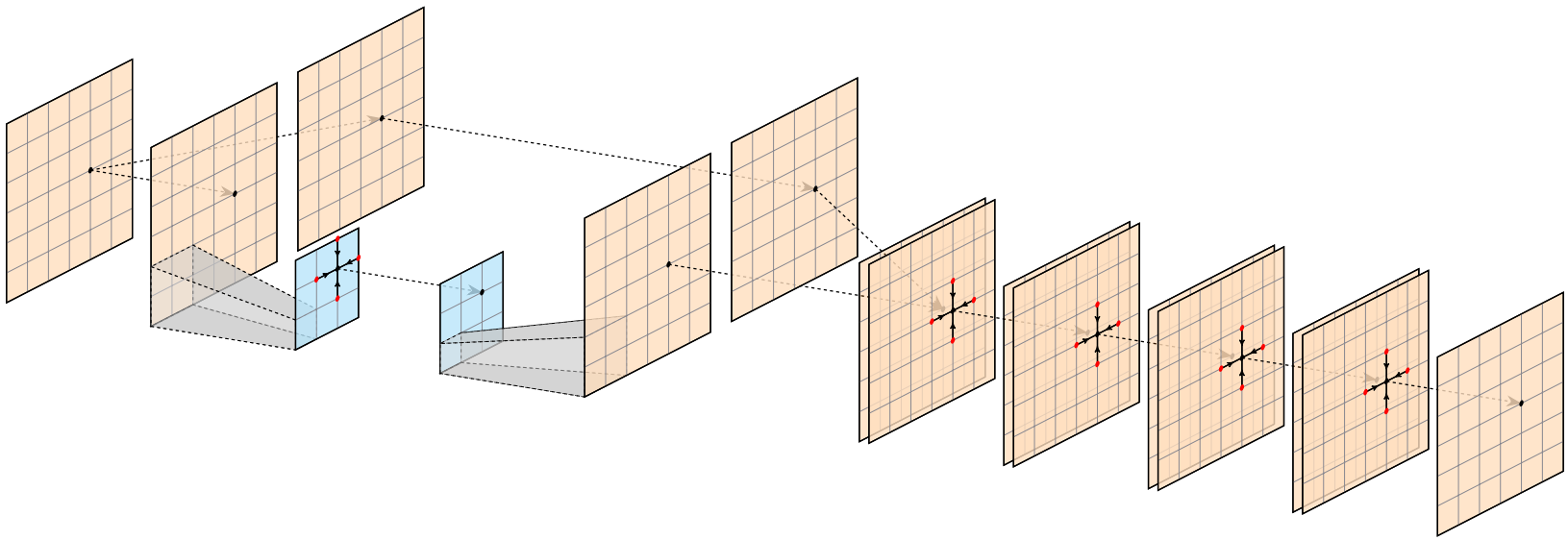


- Iteration count gain from using M_s as preconditioner for $Du = b$ with initial guess zero
- Performance is $\sim$ twice that of M_h with 1 layer/1 hop (since $r = 2$)
- Trained PTC model is used as initial weights for LPTC model (but no benefit from LPTC)

Multigrid model setup

- Duplicate the input feature and preserve one copy for smoother
- Restrict other copy to coarse grid and apply our coarse-grid model
- Prolongate result to fine grid
- Combine copy of initial feature and result of coarse-grid model to two input features for last four layers (= smoother)

Combined two-level multigrid model



- Allows for efficient transport of information over both short and long-distances
- Additional levels: Recursively replace coarse-grid layer by entire model

Training strategy for multigrid model

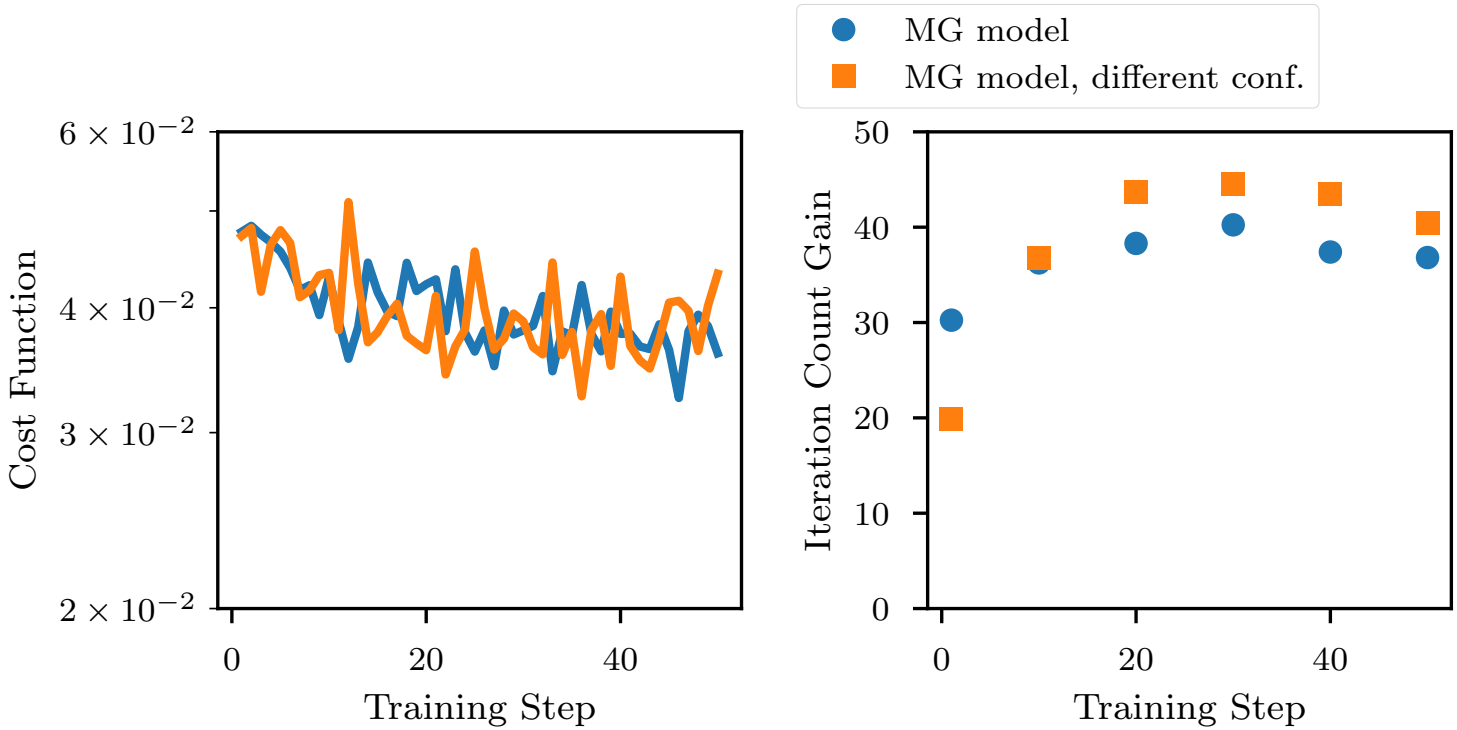
- In principle, model should work well with layer weights from individual models
- Performance can be further improved by continued training with cost function

$$C = |Mb_h - u_h|^2 + |Mb_\ell - u_\ell|^2 \qquad (*)$$

  - $b_h = D_{\mathrm{WC}} v_1$, $u_h = v_1$, $b_\ell = v_2$, $u_\ell = D_{\mathrm{WC}}^{-1} v_2$
  - v_1 and v_2 are random vectors with $|b_h| = |b_\ell| = 1$

- Focus on high- or low modes could be shifted by relative prefactor in $(*)$

Results for full multigrid model



- Performance greatly improved over individual high-/low-mode models
- Continued training converges very quickly
- Transfer learning works again after brief retraining

Summary of important points so far

- We reformulate the problem of constructing a (multigrid) preconditioner in the language of gauge-equivariant neural networks.
- We find that such networks can learn the general paradigms of multigrid and significantly reduce the iteration count of the outer solver.
- Transfer learning: If we change the configuration or parameters such as κ and β, only very little or no extra training is needed.
- We can implement communication avoidance naturally.
- We provide a flexible implementation interface (GPT, http://github.com/lehner/gpt) for experimentation and further studies.

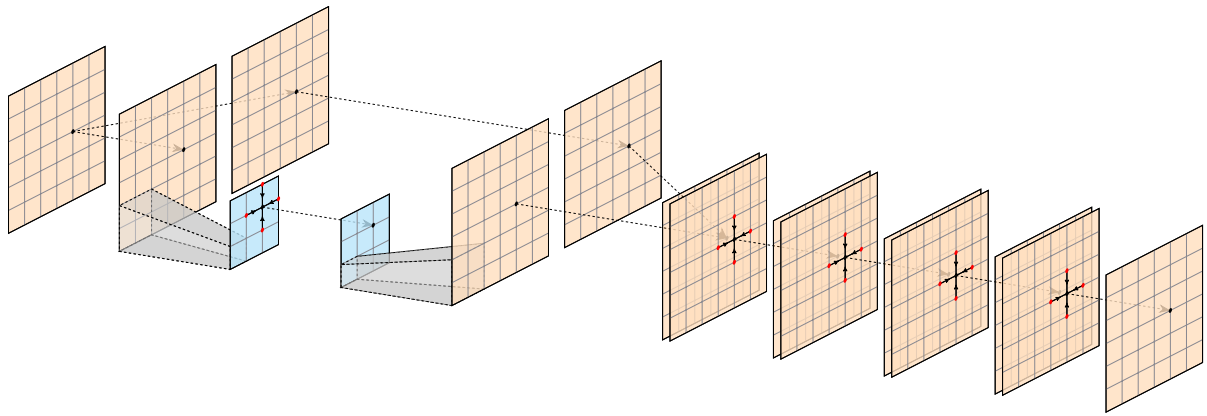
A broader look

Motivation of research program

- Goal: approximate propagators D^{-1}, $\det(D)$, and hadronic correlation functions
- Deep networks work (cf. Krylov solvers)
- Multigrid paradigm makes much shallower models perform as well as deep ones (cf. multigrid solvers)
- By casting it in language of neural networks it is easy to investigate non-Krylov methods.
- Focus on explicitly gauge-equivariant models such that gauge-equivariance does not have to be learned. Helps with transfer learning.

The gauge-equivariant multigrid neural network research program

- Status after arXiv:2302.05419:
  - We had a multigrid preconditioner model



  - with conventional construction of restriction and prolongation layers (RL/PL)
  - and without explicit gauge degree of freedom on coarse grid
  - which approximates a Dirac propagator D^{-1} well both at short and long distances.

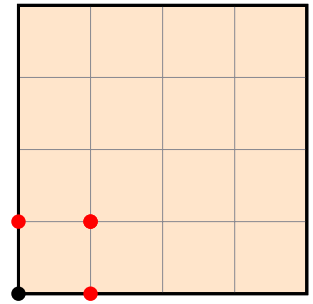
The gauge-equivariant multigrid neural network research program

- In second paper arXiv:2304.10438:
  - Keep explicit gauge degree of freedom on coarse grid and
  - use a gauge-equivariant layer on coarse grid (reduced number of weights).
  - Parametrize RL and PL with gauge-invariant weights (spin matrices).
  - Show absence of critical slowing down in preconditioned solve even for $Q \neq 0$.

The gauge-equivariant multigrid neural network research program

- Future work:
  - Relate RL/PL spin matrices to energy density, topology density, Wilson loops via gauge-invariant models. This would eliminate most of the typical multigrid setup cost. Useful for ensemble generation.
  - Address fermions with more complex spectrum (such as DWF)
  - Do not just approximate D^{-1} but directly complex hadronic correlation functions to be used in AMA.

Explicit gauge degree of freedom on coarse grid

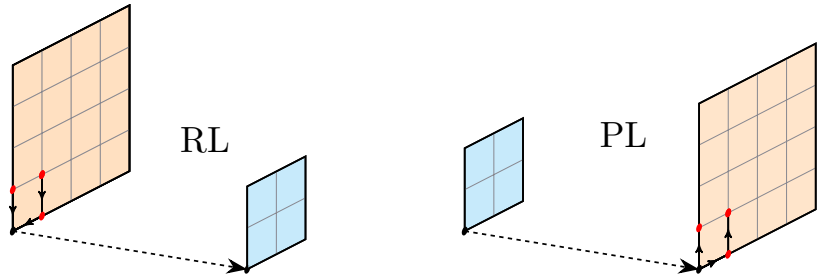


$B(y) = \{\bullet, \bullet\}$

$B_r(y) = \bullet$

- Field on fine grid: $\phi : S \to V_G \otimes V_{\bar{G}}, x \mapsto \phi(x)$ with local gauge space (V_G), non-gauge space ($V_{\bar{G}}$), and set of fine-grid sites S
- Gauge transformation: $\phi(x) \to \Omega(x)\phi(x)$
- Set of coarse sites $\tilde{S}$ and block map $B : \tilde{S} \to \mathcal{P}(S), y \mapsto B(y)$ (sites $B(y)$ on fine grid correspond to y on coarse grid)
- A reference site $B_r : \tilde{S} \to S, y \mapsto B_r(y)$ such that $B_r(y) \subset B(y)$
- Field on coarse grid: $\tilde{\phi} : \tilde{S} \to V_G \otimes \tilde{V}_{\bar{G}}, y \mapsto \tilde{\phi}(y)$ (note: same local gauge space as on fine grid)
- Find restriction and prolongation layers such that $\tilde{\phi}(y) \to \tilde{\Omega}(y)\tilde{\phi}(y)$ under gauge transformation Ω with

$$\tilde{\Omega}(y) = \Omega(B_r(y)) \,.$$



- Define RL/PL by pooling and subsampling layers:

$$\mathrm{RL} = \mathrm{SubSample} \circ \mathrm{Pool}\,, \tag{1}$$

$$\mathrm{PL} = \mathrm{Pool}^\dagger \circ \mathrm{SubSample}^\dagger\,. \tag{2}$$

  (weights in RL and PL can differ, so not necessarily $\mathrm{RL}^\dagger = \mathrm{PL}$)

- The pooling layer Pool: $\mathcal{F}_\phi \to \mathcal{F}_\phi$, $\phi \mapsto \mathrm{Pool}\phi$ is given by

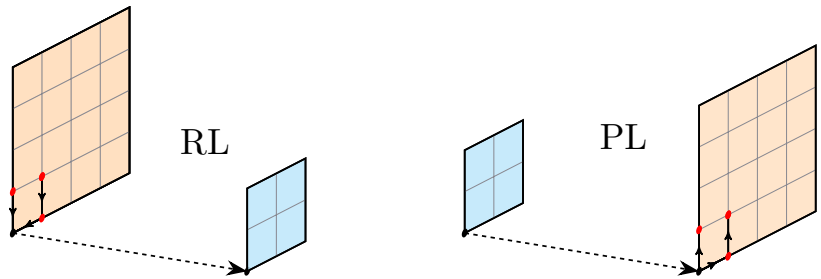
$$\mathrm{Pool}\phi(x) = \sum_{q \in Q} W_q(x)\, T_q \phi(x) \tag{3}$$

  with $q = (p, \bar{U})$, path p, gauge field $\bar{U}$, and $T_q = T_p(\bar{U})$. Weights $W_q(x)$ are spin matrices, separated gauge DOF.

- The subsampling layer is given by

$$\mathrm{SubSample}\phi(y) = \phi(B_r(y))\,. \tag{4}$$

More details on the pooling layer



- Gauge field $\bar{U}$ in $T_p(\bar{U})$ needs to satisfy

$$\bar{U}_\mu(x) \to \Omega(x)\bar{U}_\mu(x)\Omega^\dagger(x+\hat{\mu})\,. \tag{5}$$

  In practice, we use a variety of differently smeared links.

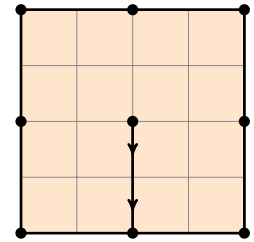
- Complete set of paths P transports every element of $B(y)$ exactly once to $B_r(y)$ $\Rightarrow |P| = |B(y)|$

- Efficient implementation for each complete set of path possible: GPT

- $\tilde{\phi} = \mathrm{RL}\phi$ yields $\tilde{\phi}(y) \to \tilde{\Omega}(y)\tilde{\phi}(y)$ under gauge transformations $\phi(x) \to \Omega(x)\phi(x)$

Explicit gauge-equivariant coarse layers need coarse gauge field

- Plain coarse gauge field construction:



$$B_r(y') - B_r(y) = b\hat{\mu}$$

with unit vector $\hat{\mu}$ in direction μ and $b \in \mathbb{N}^+$. The coarse-grid gauge field $\tilde{U}_\mu(y)$ corresponding to this pair of reference points is then simply

$$\tilde{U}_\mu(y) = U_\mu(B_r(y)) \cdots U_\mu(B_r(y) + (b-1)\hat{\mu})\,. \tag{6}$$

- Galerkin coarse gauge field construction:

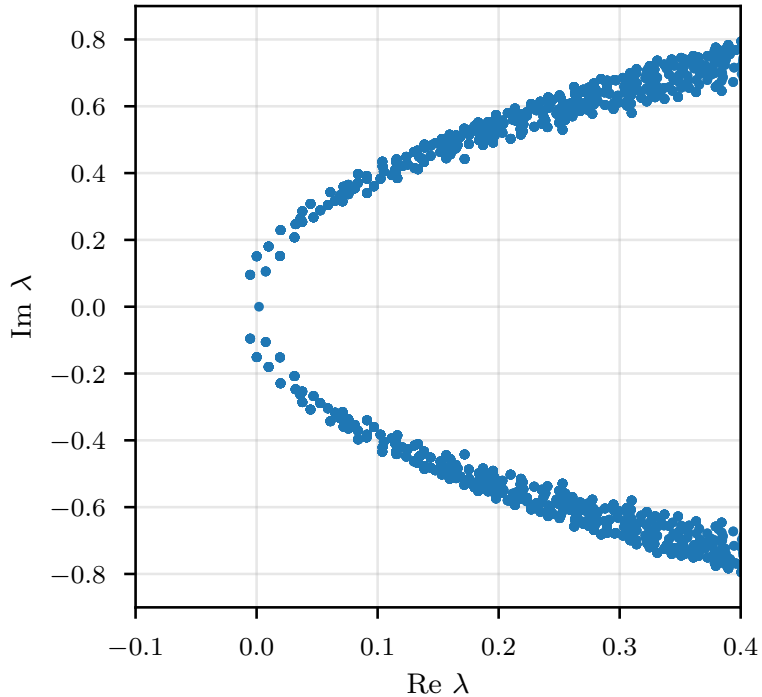
$$\tilde{U}_\mu(y) = \tilde{D}(y, y+\hat{\mu}) \tag{7}$$

with

$$\tilde{D} = \mathrm{RL} \circ D \circ \mathrm{PL} \tag{8}$$

for Wilson-clover D.

Spectrum of Wilson-clover Dirac operator



- $\beta = 6$ pure Wilson gauge field with topological charge $Q = 1$
- $8^3 \times 16$ lattice sites
- Wilson-clover operator with $m = -0.5645$ and $c_{\mathrm{sw}} = 1$

Model details I

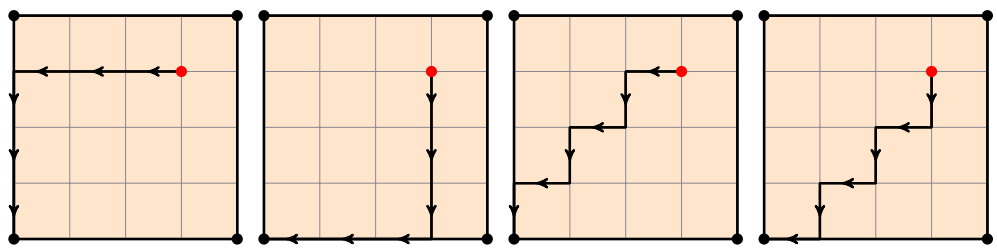
- Need prescription for q in

$$\mathrm{Pool}\phi(x) = \sum_{q \in Q} W_q(x)\, T_q \phi(x)$$

with $q = (p, \bar{U})$, path p, gauge field $\bar{U}$, and $T_q = T_p(\bar{U})$.

- For fixed i, we define paths $p^{(ij)}$ that connect all elements of $B(y)$, enumerated by $j = 1, \ldots, |B(y)|$, to the reference site $B_r(y)$. For different i we use different prescriptions for the paths $p^{(ij)}$, and then use the couples $q_{ij} = (p^{(ij)}, \bar{U}^{(i)})$.

- We define four different prescriptions $\hat{p}_1, \ldots, \hat{p}_4$ (depth first/breadth first lexicographic/reverse lexicographic)



and set $p^{(ij)} = \hat{p}^{(j)}_{i \bmod 4}$.

Model details II

- Concretely, we use 9 different gauge fields $\bar{U}^{(i)}$ with $i = 1, \ldots, 9$. We construct the $\bar{U}^{(i)}$ by applying $i(i-1)/2$ steps of $\rho = 0.1$ stout smearing to the unsmeared gauge fields U. Smearing radius proportional to $\sqrt{i(i-1)}$.

- So we have 9 different spin-matrix fields $W_1(x), \ldots, W_9(x)$.

- In practice, sufficient to use same weights in PL and RL such that $\mathrm{PL} = \mathrm{RL}^\dagger$. Found no benefits from general case.

- Coarse-grid size $2^3 \times 4$

Training setup – How to train RL/PL?

- Obvious approach: train

$$\mathrm{PL} \circ \mathrm{RL} \tag{9}$$

as an autoencoder with training vectors from the near-null space.

- This could be done with a cost function

$$C = |\mathrm{PL} \circ \mathrm{RL} v_\ell - v_\ell|^2 \tag{10}$$

with fine-grid vectors v_ℓ. For each training step we select a random element of $v_\ell \in \{u_1, \ldots, u_s\}$ of the near-null space vectors u_i defined above.

- Use Adam optimizer.

- Result: did not perform well in MG preconditioner!

Training setup – How to train RL/PL?

- What was missing: PL ∘ RL should also project high eigenmodes to zero (if not could overload smoother layers)
- Found also additional benefit from encouraging RL ∘ PL = $\mathbb{1}$ such that we have a proper projection operator $P = \text{PL} \circ \text{RL}$ with $P^2 = P$.
- We implement this strategy by using the cost function

$$C = |\text{PL} \circ \text{RL} v_\ell - v_\ell|^2 + |\text{PL} \circ \text{RL} v_h - P_\ell v_h|^2 + |\text{RL} \circ \text{PL} v_c - v_c|^2 \qquad (11)$$

with additional fine-grid vector v_h and coarse-grid vector v_c. For each training step v_h and v_c are random vectors with elements normally distributed about zero.
P_ℓ is the blocked low-mode projector

$$P_\ell = W^\dagger W\,, \qquad W(y,x)^\dagger = \sum_{i=1}^{s} \bar{u}_i^y(x)\hat{e}_i^\dagger \qquad (12)$$

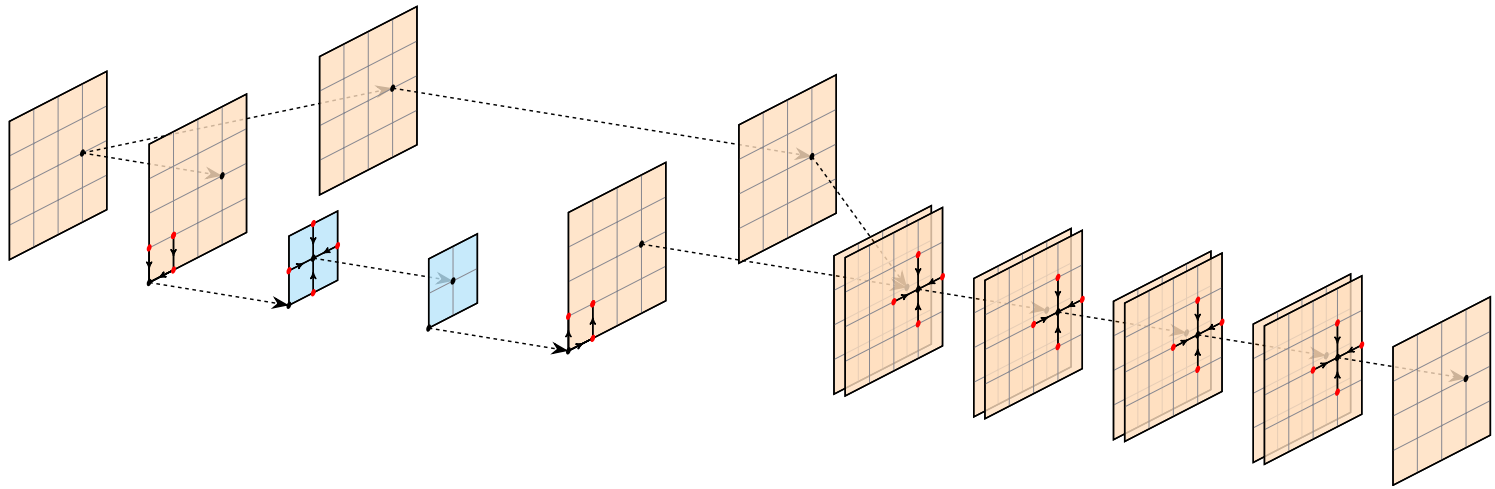
with block-orthonormalized $\bar{u}_i$ from u_i.

- All vectors v_ℓ, v_h, and v_c are normalized to unit length before being used in the cost function.

Training setup – How to train RL/PL?

- In first paper $s = 12$, here $s = 4$ was sufficient.
- Training converged after O(1000) steps.
- Yields $W_1(x), \ldots, W_9(x)$ but still costly since we first need near-null space vectors.
- In future work: obtain $W_i(x)$ as output of gauge-invariant models based on energy density $E(x)$, topology density $Q(x)$, plaquette $P(x)$ and other Wilson loops. At this point the u_i are no longer needed. (In a sense we generate training data for the next step in this work.)

Training setup – combined preconditioner model



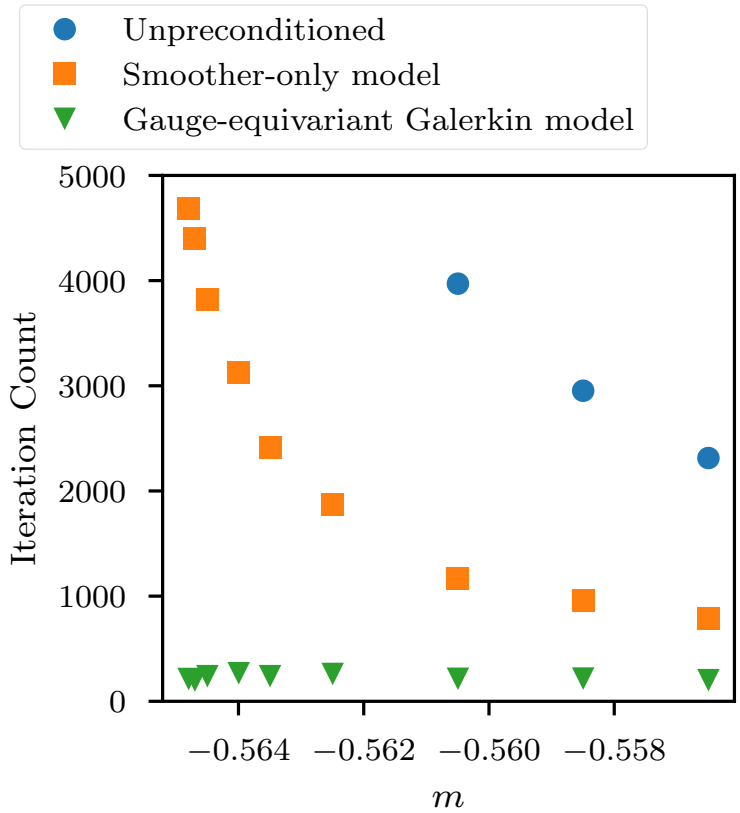
- First train RL/PL as described above.
- Then train combined model with frozen RL/PL using cost function

$$C = |Mb_h - u_h|^2 + |Mb_\ell - u_\ell|^2 \tag{13}$$

with $b_h = Dv_1$, $u_h = v_1$, $b_\ell = v_2$, and $u_\ell = D^{-1}v_2$.

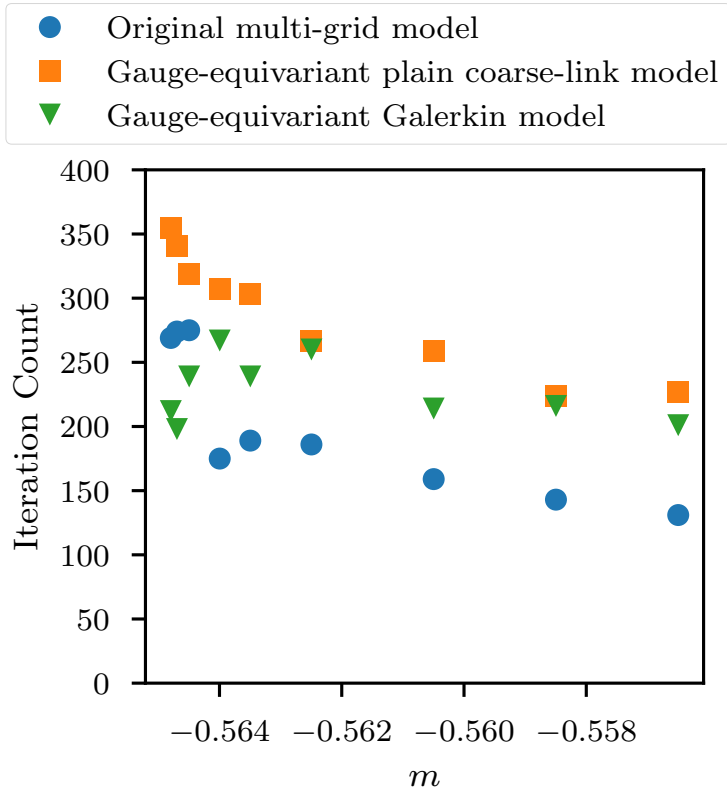
- Further training with unfrozen RL/PL leads to no notable improvement.

Results – critical slowing down



- Show outer iteration count in GMRES to 10^{-8} precision with and without model as preconditioner.
- Model with Galerkin gauge fields removes critical slowing down.

Results – critical slowing down



- Original multi-grid model also removes critical slowing down.
- Model with plain gauge fields shows small remnants of critical slowing down.

Summary and Outlook

- We now have an explicitly gauge-equivariant model on coarse and fine grid.
- The multigrid model removes critical slowing down when used as a preconditioner.
- We factored the gauge degree of freedom in RL and PL and learned the resulting spin-matrix weights.
- In future work, these spin-matrices will be outputs of gauge-invariant models themselves, which then removes the typical setup cost.
- Complex hadronic correlation functions can be approximated by intermediate approximations to propagators. However, it may be more efficient to use our architectures to directly approximate the correlation functions for use in AMA.
- Good approximations of propagators may be also be useful in HMC and other generative models.
- Investigate cases with more challenging spectrum (such as DWF).