

A novel gauge-equivariant neural network architecture for preconditioners in lattice QCD

Daniel Knüttel, Christoph Lehner, **Simon Pfahler**, Tilo Wettig

University of Regensburg

November 3, 2025



Daniel Knüttel



Christoph Lehner

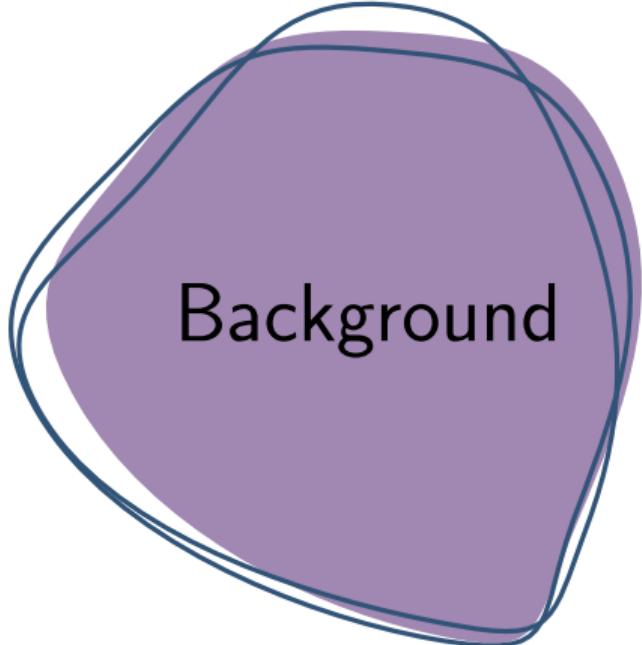


Tilo Wettig



Studienstiftung
des deutschen Volkes





Background

Problem statement

Bigger picture

- ▶ Lattice QCD calculations are expensive
- ▶ One of the most expensive parts is solving the Dirac equation
- ▶ Preconditioners can accelerate this

Preconditioners

- ▶ State of the art:
algebraic multigrid (AMG)
- ▶ But AMG needs costly setup
- ▶ Can this setup be avoided?
- ▶ **Goal:** Preconditioner competitive with AMG, but with very little (or even no) setup



Status quo

Previous work

- ▶ Exploratory studies with different kinds of Machine Learning models [1,2]
- ▶ Multigrid coarse grid solver and smoother were reformulated in a learnable way [3]
- ▶ Prolongation and restriction operators were translated into ML language [4]

- [1] S. Calì et al., PRD **107**, 034508 (2023)
- [2] Y. Sun et al., arXiv:2509.10378 [hep-lat] (2025)
- [3] C. Lehner and T. Wettig, PRD **108**, 034503 (2023)
- [4] C. Lehner and T. Wettig, PRD **110**, 034517 (2024)

Shortcomings

- ▶ Focus on preconditioning the high modes
- ▶ expensive AMG setup is still needed
- ▶ no generalization to unseen configurations



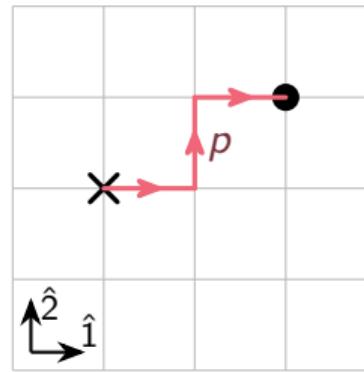
Gauge equivariance

- ▶ Gauge-equivariant transport of information between adjacent lattice sites:

hop operator

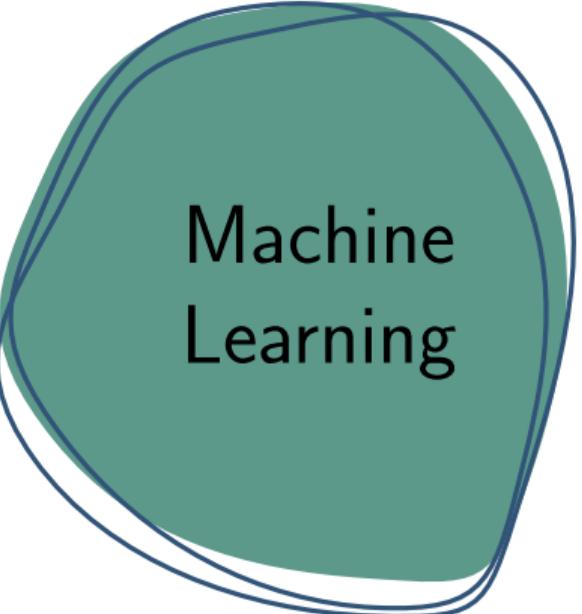
$$H_{p_i} \varphi(x) = U_{p_i}^\dagger (x - \hat{p}_i) \varphi(x - \hat{p}_i)$$

- ▶ Combining multiple hops leads to **parallel transports** T_p along a path p
- ▶ weighted linear combinations of parallel transports are gauge-equivariant



$$T_p = H_1 H_2 H_1$$





Machine
Learning

Network Architecture

Dense linear (L) layer

- ▶ n spin-color fields
 $\varphi \in \mathbb{C}^{L_x \times L_y \times L_z \times L_t \times N_c \times N_s}$
- ▶ build m linear combinations of the input fields, with spin matrices $W_{ij} \in \mathbb{C}^{N_s \times N_s}$ as weights:

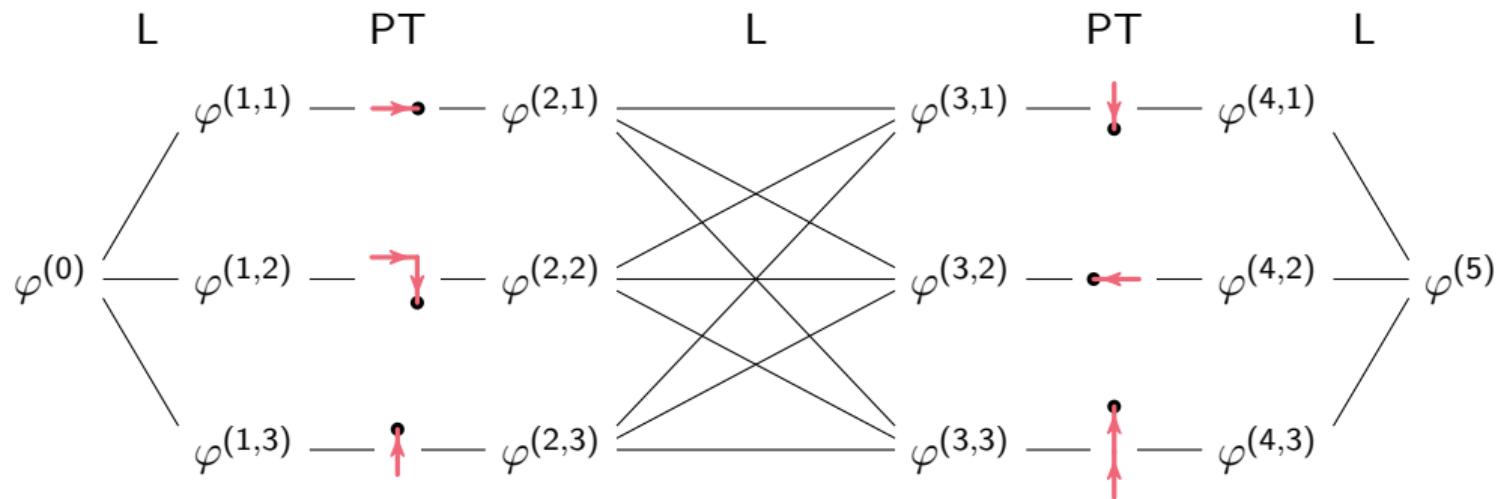
$$\psi_i(x) \stackrel{\text{L}}{=} \sum_{j=1}^n W_{ij} \varphi_j(x)$$

Parallel Transport (PT) layer

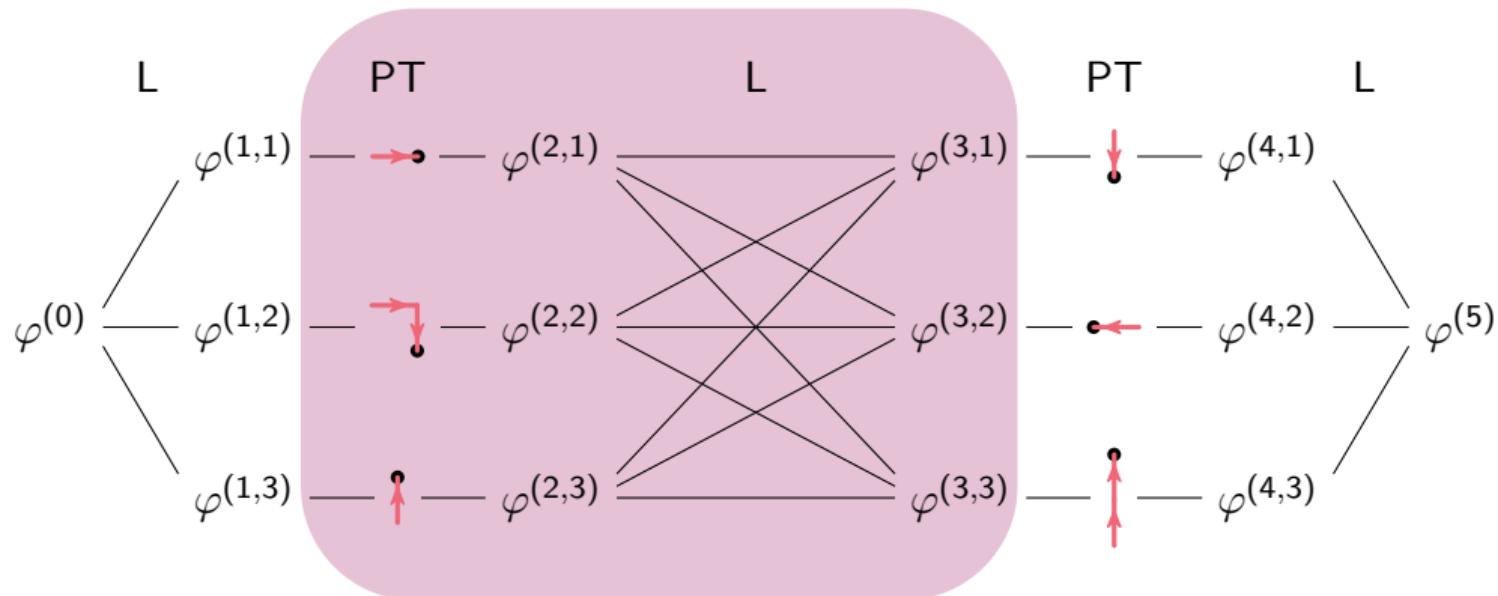
- ▶ n spin-color fields
 $\varphi \in \mathbb{C}^{L_x \times L_y \times L_z \times L_t \times N_c \times N_s}$
- ▶ apply a different parallel transport $p^{(i)}$ to every input field:

$$\psi_i(x) \stackrel{\text{PT}}{=} T_{p^{(i)}} \varphi_i(x)$$

Network Architecture



Network Architecture



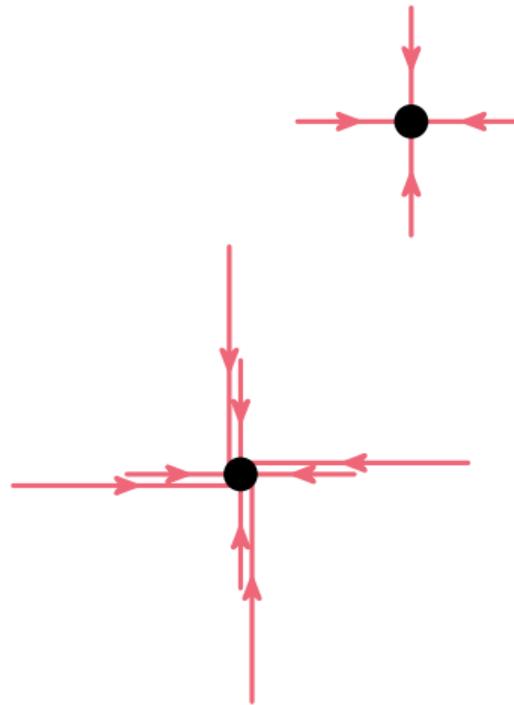
Choice of PT paths

- ▶ 1-hop paths:

$$P = \{H_p | p \in \{\pm 1, \pm 2, \pm 3, \pm 4\}\}$$

- ▶ long straight paths:

$$P = \{H_p^n | p \in \{\pm 1, \pm 2, \pm 3, \pm 4\}, n \in \{2^0, 2^1, \dots, L_p/2\}\}$$



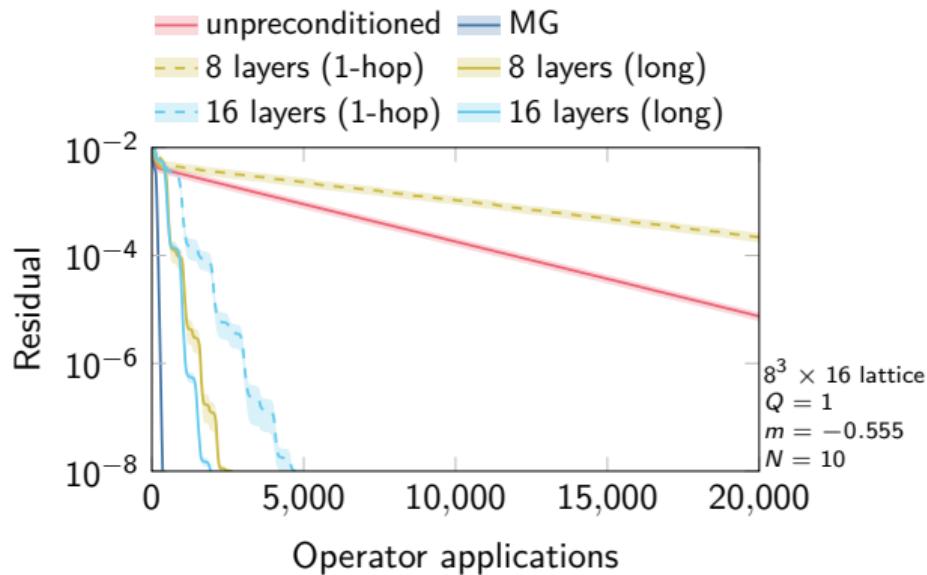
Choice of PT paths

- ▶ **1-hop paths:**

$$P = \{H_p | p \in \{\pm 1, \pm 2, \pm 3, \pm 4\}\}$$

- ▶ **long straight paths:**

$$P = \{H_p^n | p \in \{\pm 1, \pm 2, \pm 3, \pm 4\}, n \in \{2^0, 2^1, \dots, L_p/2\}\}$$



Score function

- Standard choice:

$$C = \|MD\varphi - \varphi\|^2$$

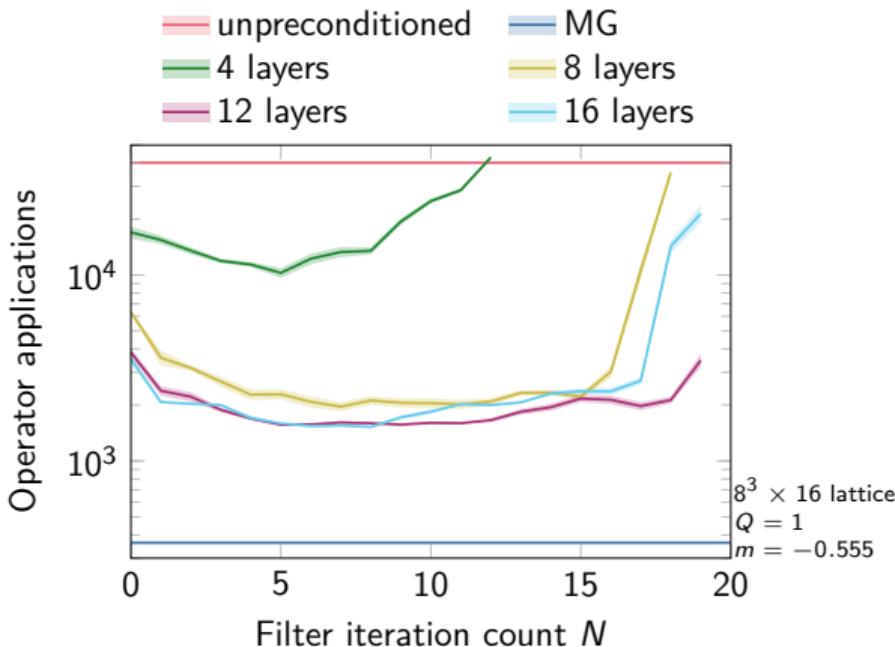
with random field φ

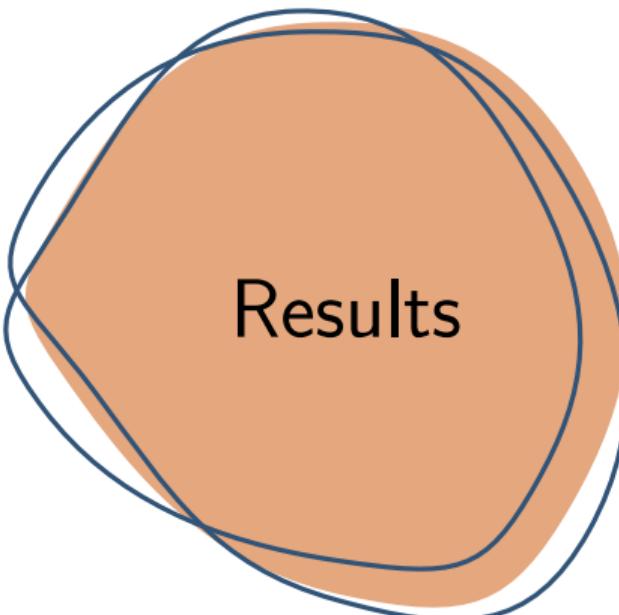
- Filtered cost function:

$$C = \|MD\tilde{\varphi} - \tilde{\varphi}\|^2$$

where $\tilde{\varphi}$ is obtained through N iterations of GMRES for $D\varphi = 0$ with random initial guess

⇒ Stronger emphasis on the low modes!





Results

Gauge configurations

$8^3 \times 16$ ensemble¹:

- ▶ Wilson action
- ▶ quenched, heat bath
- ▶ Wilson-clover Dirac operator
- ▶ configurations with topological charge $|Q| \in \{0, 1, 2\}$

$16^3 \times 32$ ensemble²:

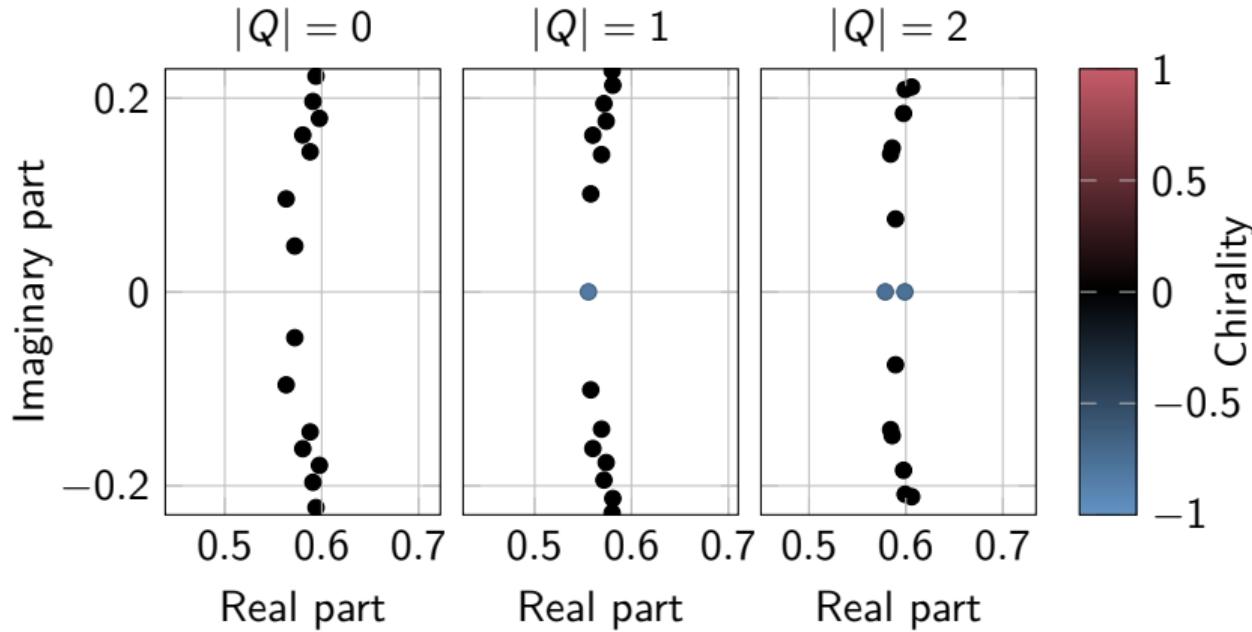
- ▶ Wilson action
- ▶ quenched, HMC
- ▶ Wilson-clover Dirac operator
- ▶ configurations with topological charge $|Q| \in \{0, 4\}$

¹ doi.org/10.5281/zenodo.15018324

² doi.org/10.5281/zenodo.17494829

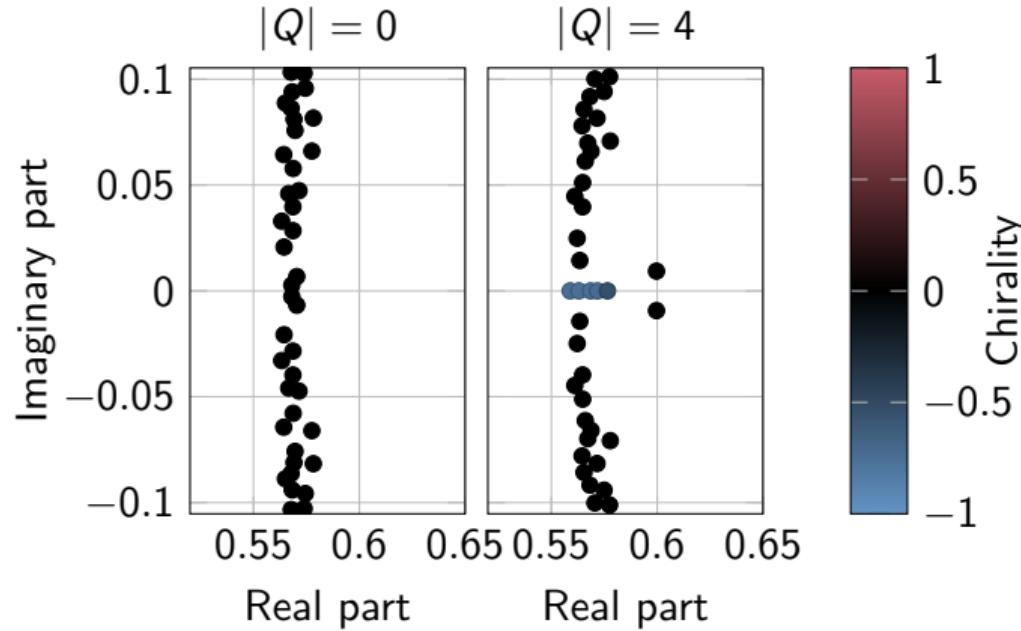
Gauge configurations

$8^3 \times 16$ lattice

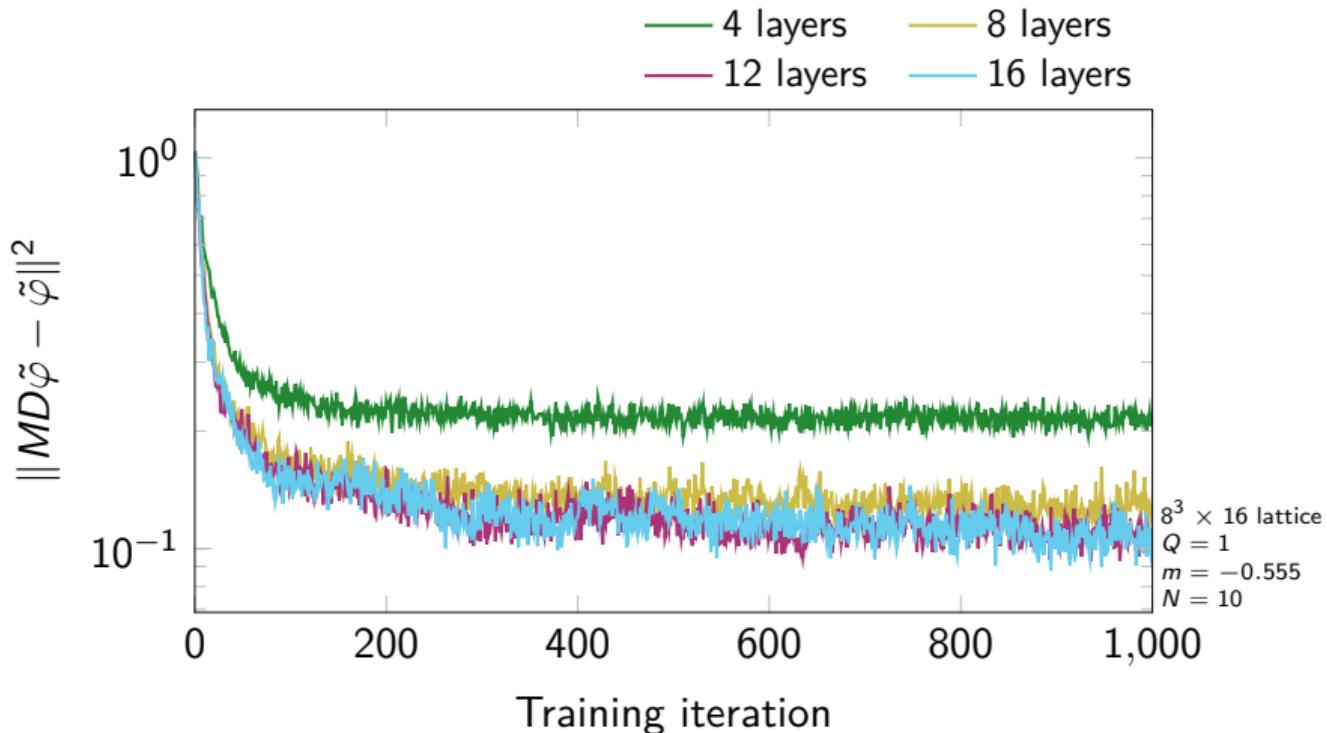


Gauge configurations

$16^3 \times 32$ lattice



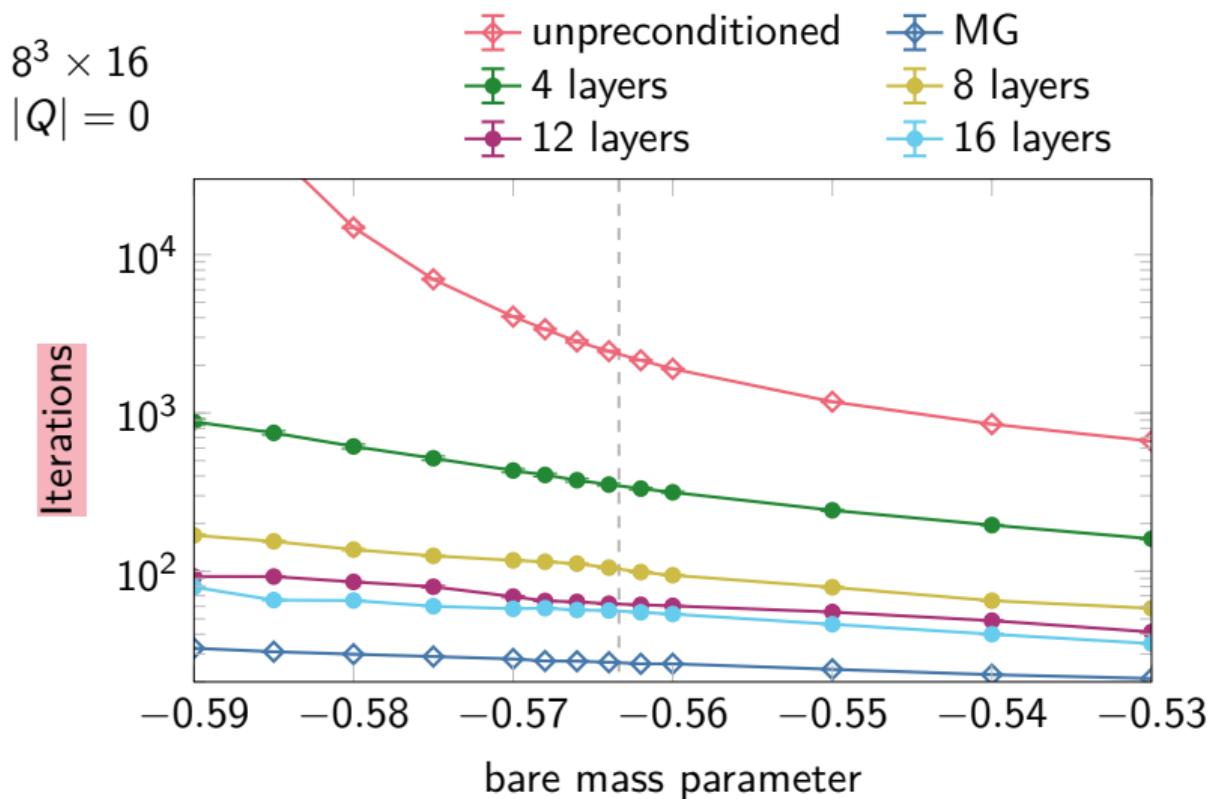
Training cost



Iteration

count

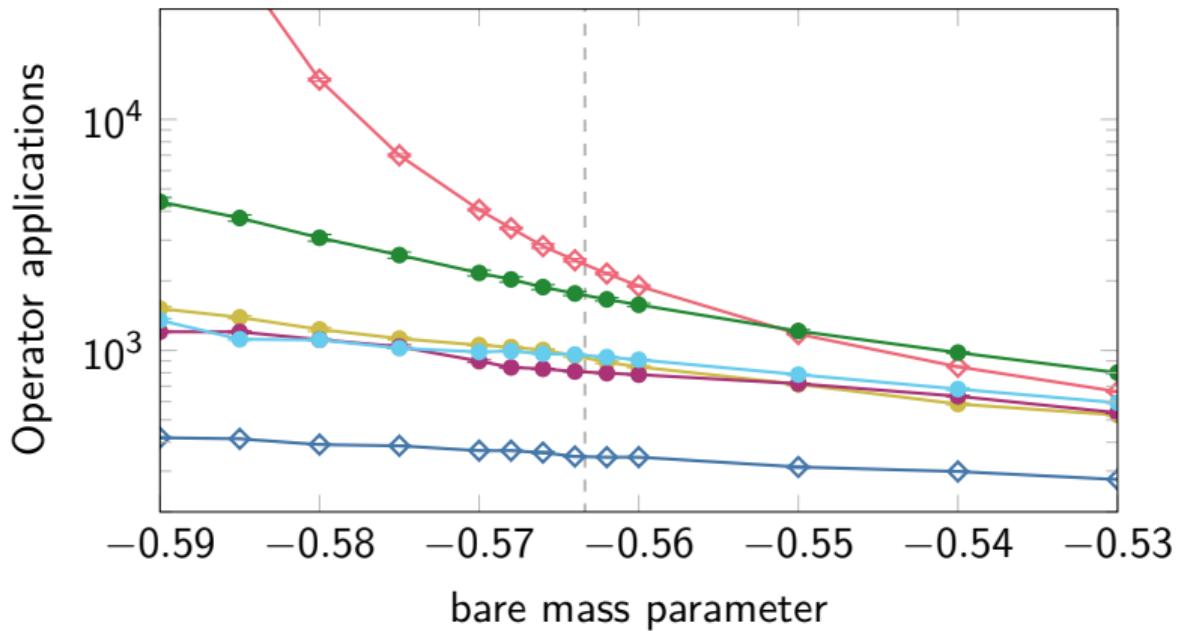
gain?



Critical slowing down

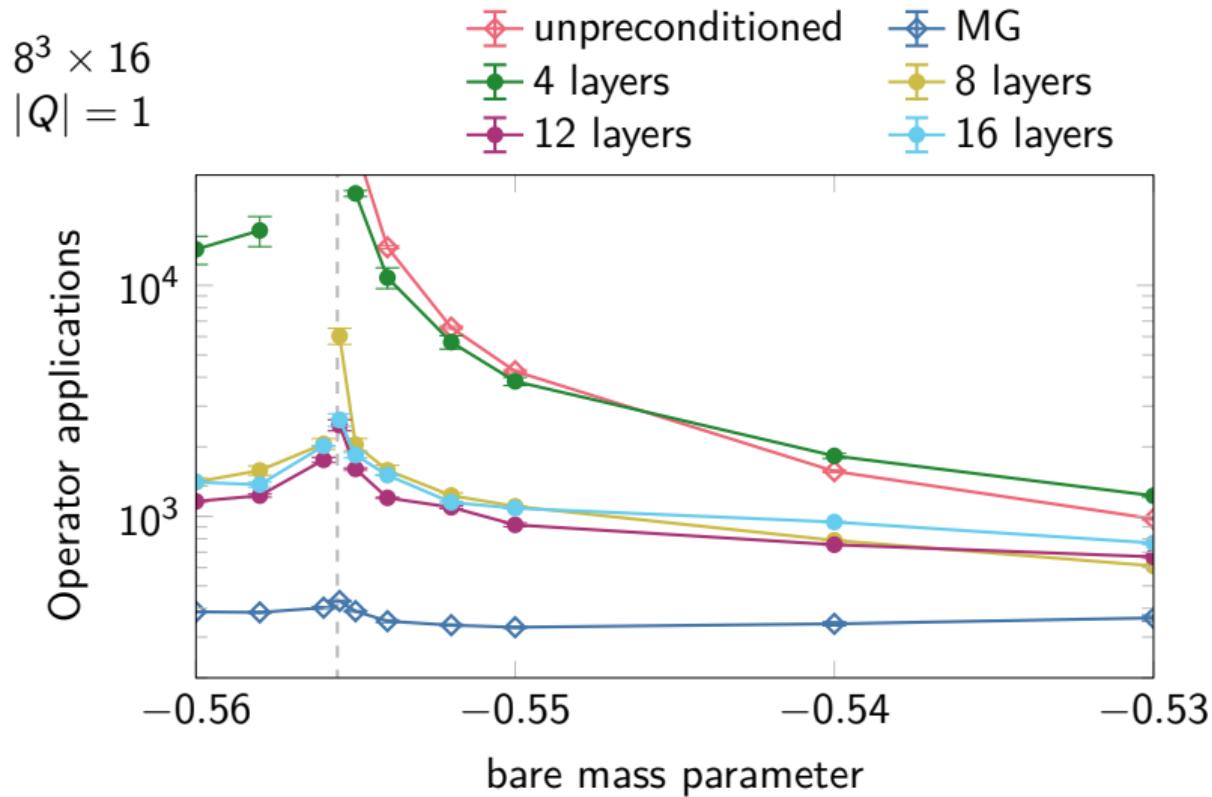
$8^3 \times 16$
 $|Q| = 0$

unpreconditioned
4 layers
12 layers
MG
8 layers
16 layers

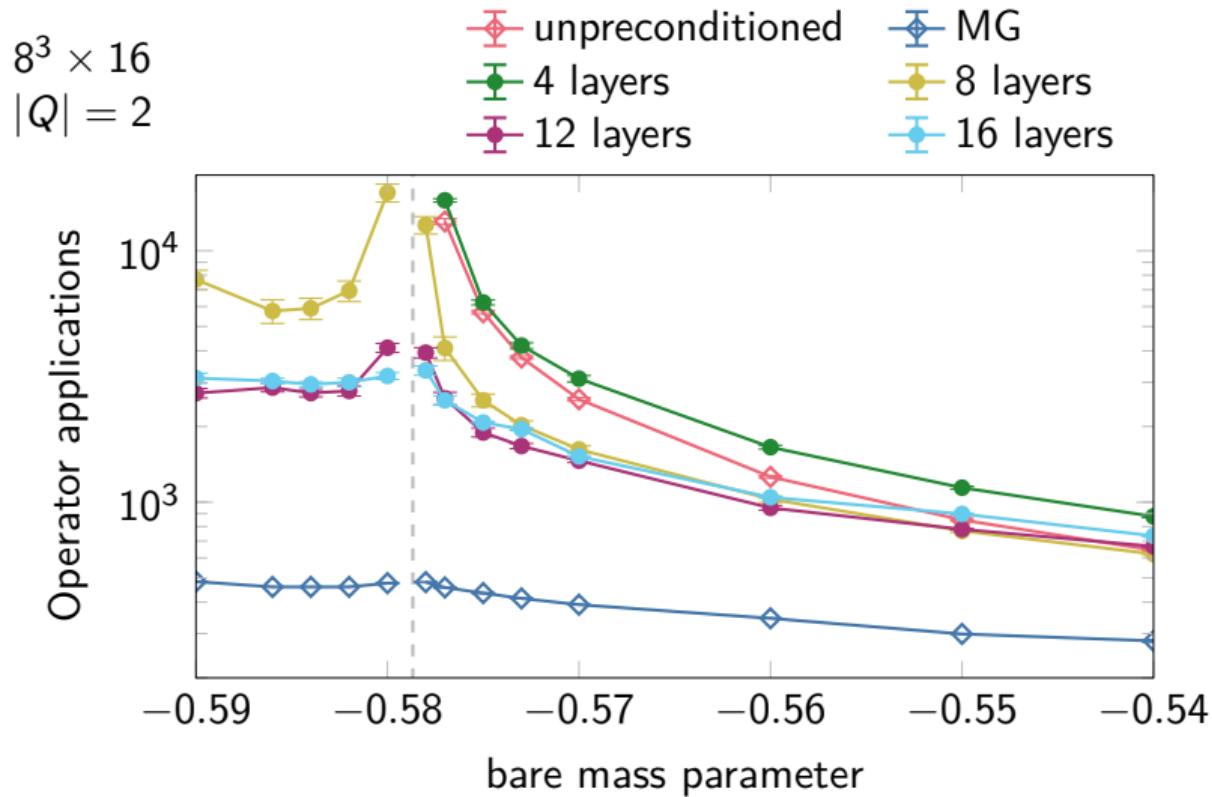


applications of:
► D_{fine}
► PT layer

Critical slowing down



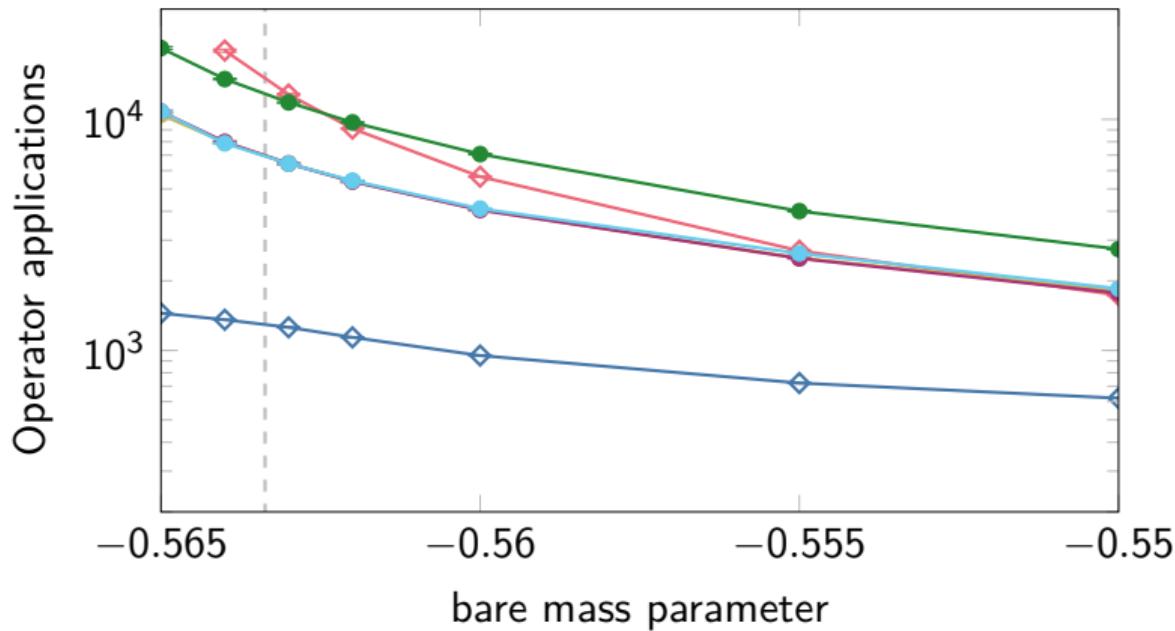
Critical slowing down



Critical slowing down

$16^3 \times 32$
 $|Q| = 0$

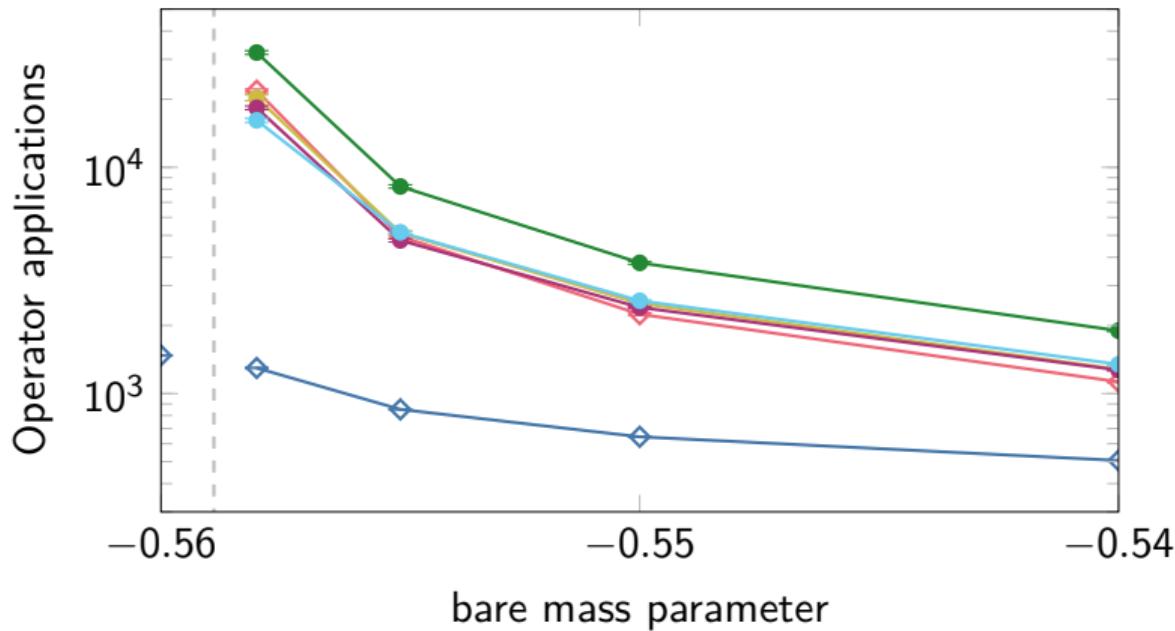
unpreconditioned
4 layers
12 layers
MG
8 layers
16 layers



Critical slowing down

$16^3 \times 32$
 $|Q| = 4$

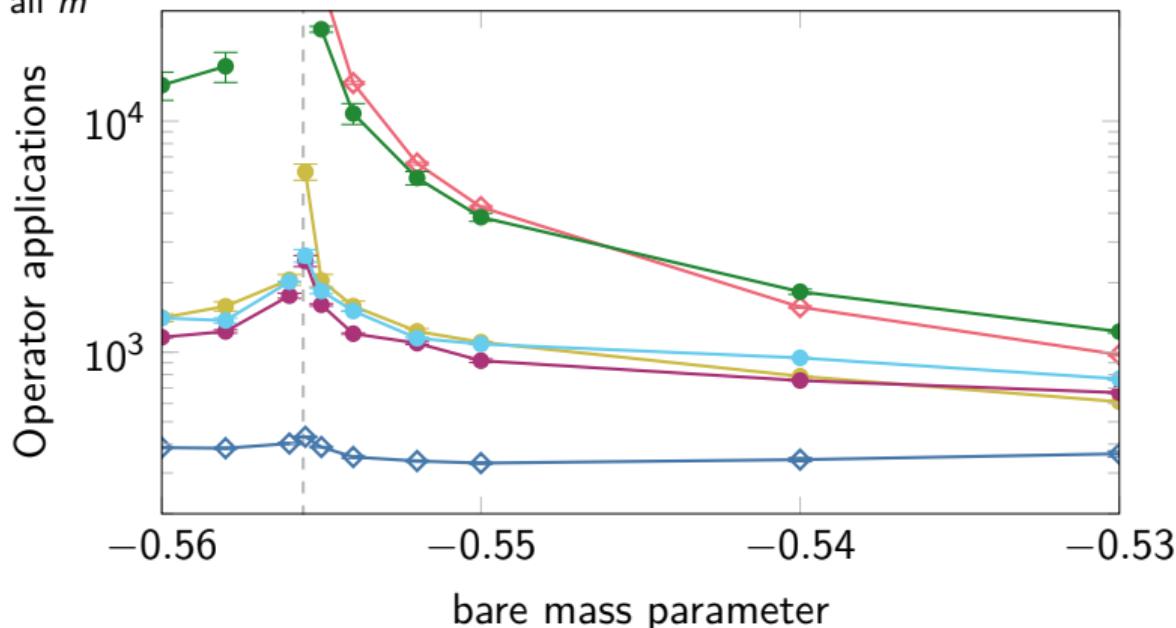
unpreconditioned
4 layers
12 layers
MG
8 layers
16 layers



Transfer

trained and applied on
 $8^3 \times 16$
 $|Q| = 1$
all m

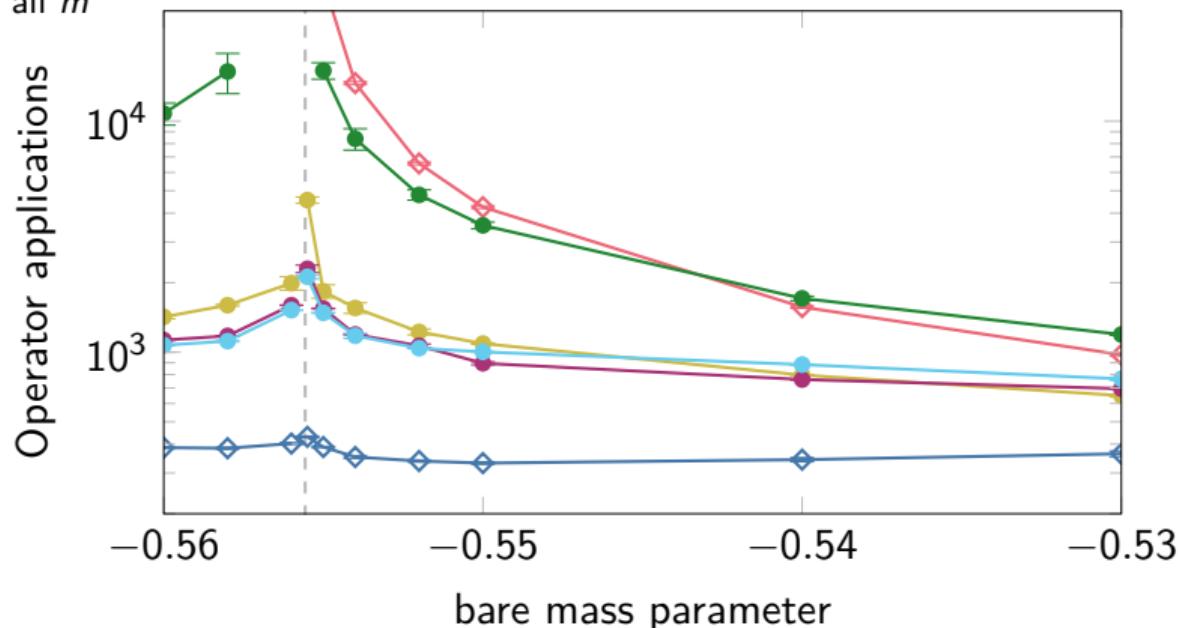
unpreconditioned
4 layers
12 layers
MG
8 layers
16 layers



Transfer

trained on applied on
 $8^3 \times 16$ $8^3 \times 16$
 $|Q| = 0$ $|Q| = 1$
 $m = -0.56$ all m

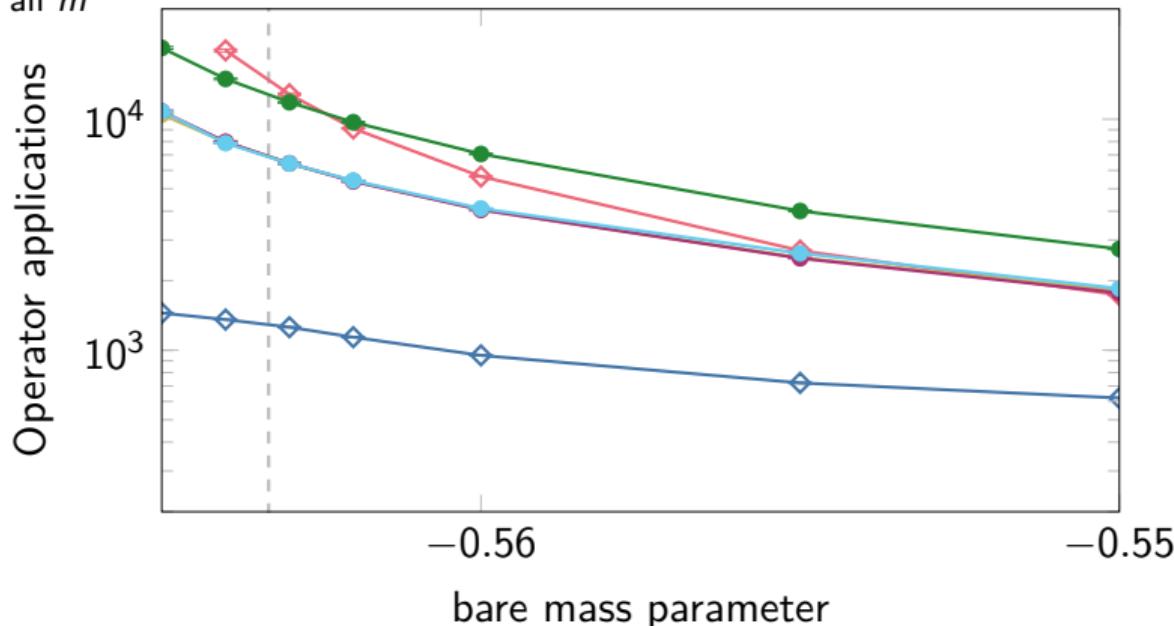
unpreconditioned MG
4 layers 8 layers
12 layers 16 layers



Transfer

trained and applied on
 $16^3 \times 32$
 $|Q| = 0$
all m

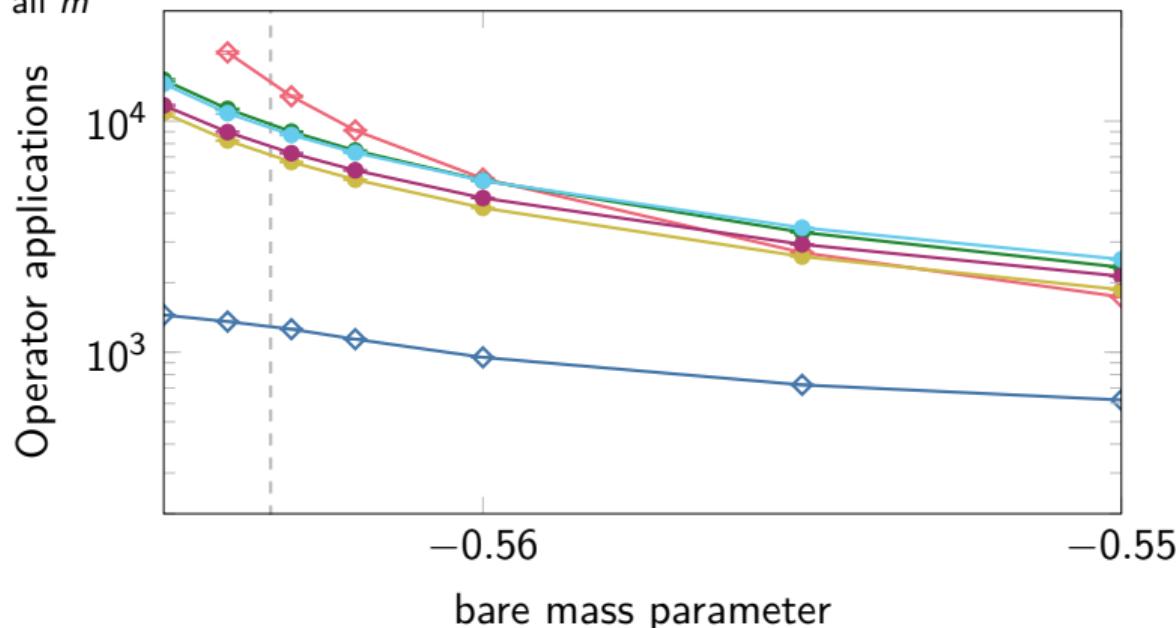
unpreconditioned
4 layers
12 layers
MG
8 layers
16 layers

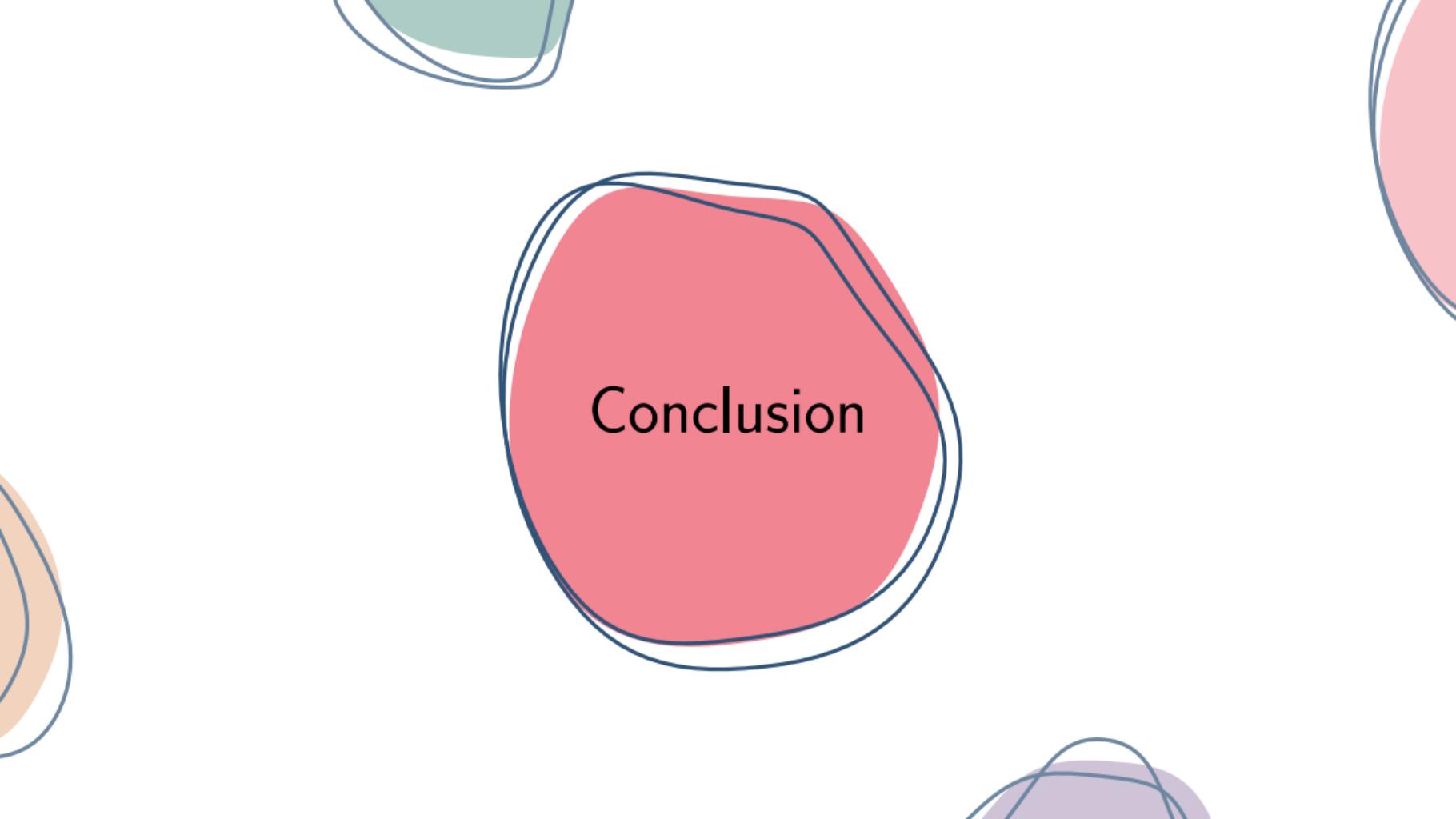


Transfer

trained on applied on
 $8^3 \times 16$ $16^3 \times 32$
 $|Q| = 0$ $|Q| = 0$
 $m = -0.56$ all m

unpreconditioned MG
4 layers 8 layers
12 layers 16 layers





Conclusion

Conclusion

Goals

- Build a preconditioner using ML
that mitigates critical slowing
down
- Transfer to unseen configurations
without retraining
→ No costly setup
- Handle topological modes nicely



Slides and code:
[simon-pfahler.github.io](https://github.com/simon-pfahler)

Conclusion

Goals

- Build a preconditioner using ML
that mitigates critical slowing
down
- Transfer to unseen configurations
without retraining
→ No costly setup
- Handle topological modes nicely



Slides and code:
simon-pfahler.github.io

Conclusion

Goals

- Build a preconditioner using ML
that mitigates critical slowing
down
- Transfer to unseen configurations
without retraining
→ No costly setup
- Handle topological modes nicely



Slides and code:
simon-pfahler.github.io

Conclusion

Goals

- Build a preconditioner using ML
that mitigates critical slowing
down
- Transfer to unseen configurations
without retraining
→ No costly setup
- Handle topological modes nicely



Slides and code:
simon-pfahler.github.io

Conclusion

Goals

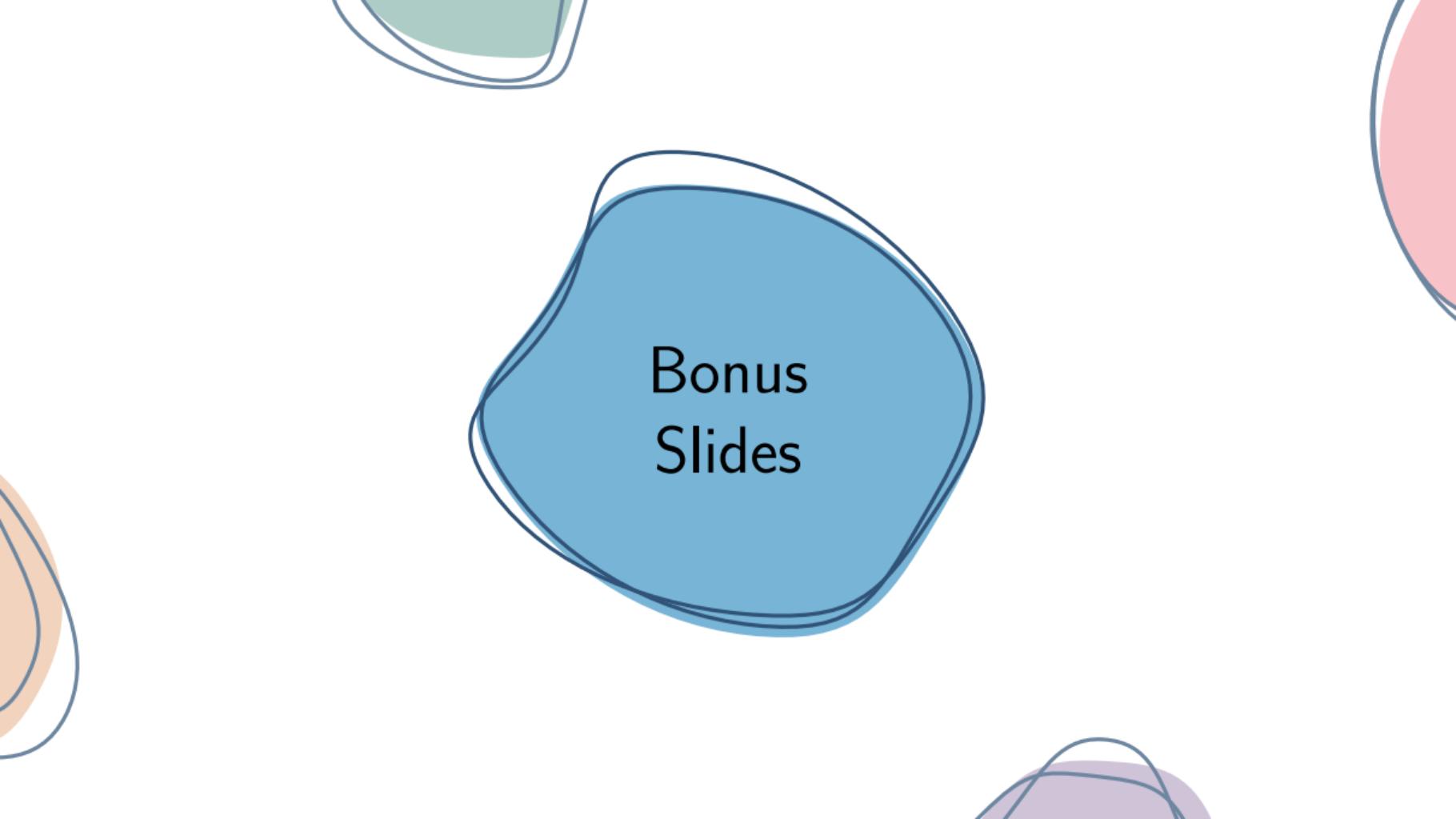
- ✓ Build a preconditioner using ML that mitigates critical slowing down
- ✓ Transfer to unseen configurations without retraining
→ No costly setup
- ✗ Handle topological modes nicely

Next steps

- ▶ Understand why the architecture fails for topological modes
- ▶ First study easier systems like Laplace operator or Schwinger model
- ▶ Identify what is problematic about topological modes
- ▶ Explore models to generate representative near-null space vectors economically

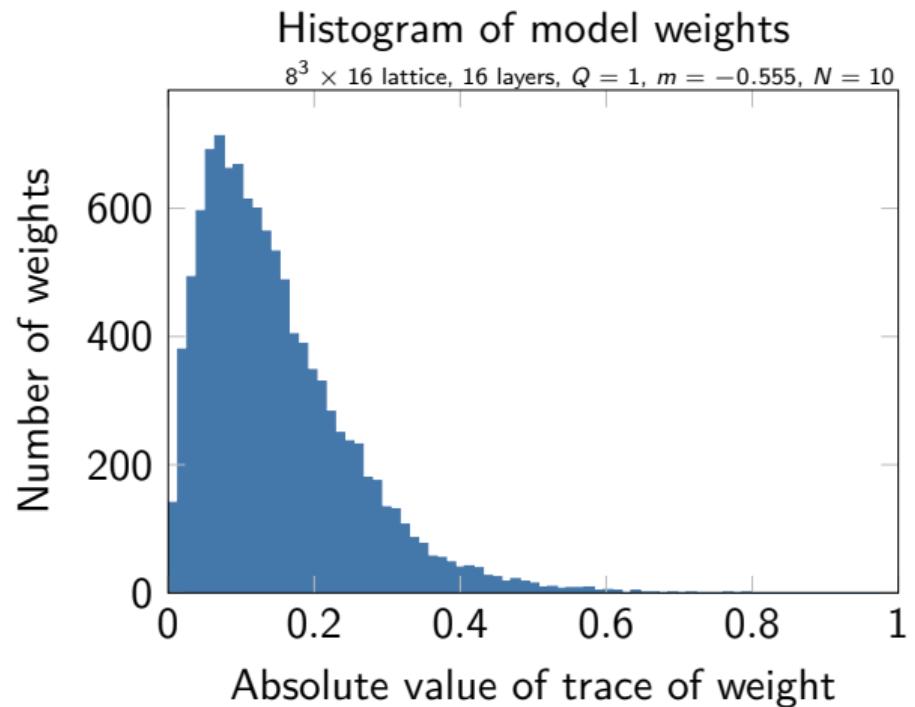


Slides and code:
simon-pfahler.github.io

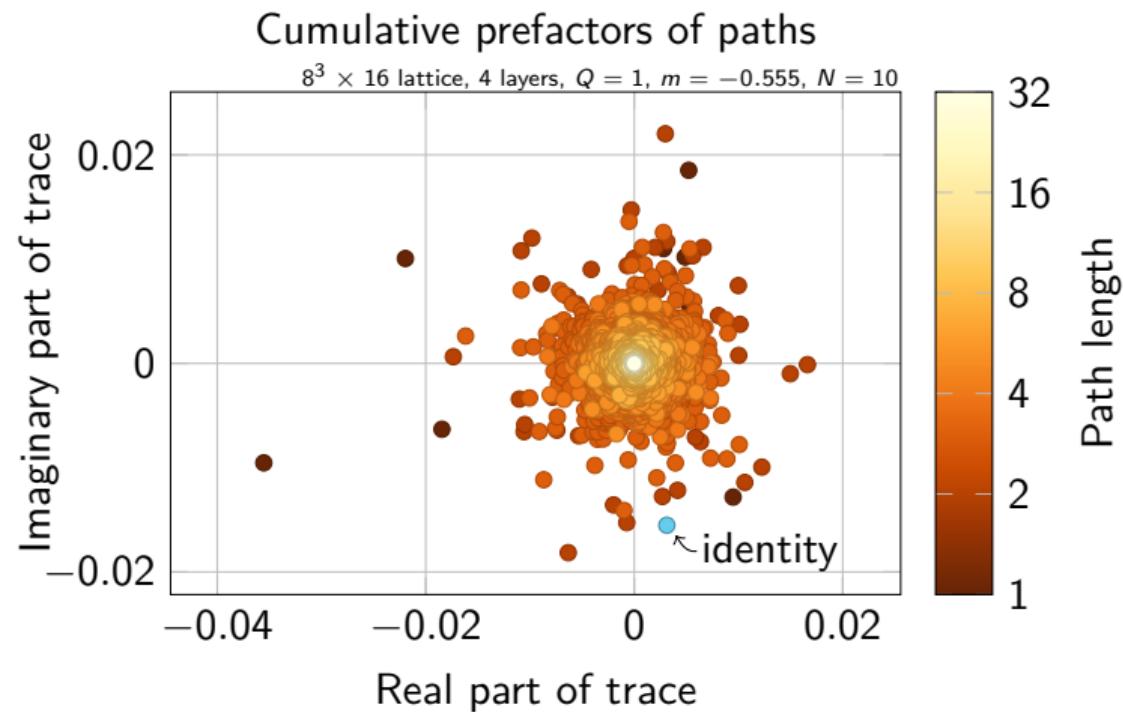


Bonus
Slides

Weights



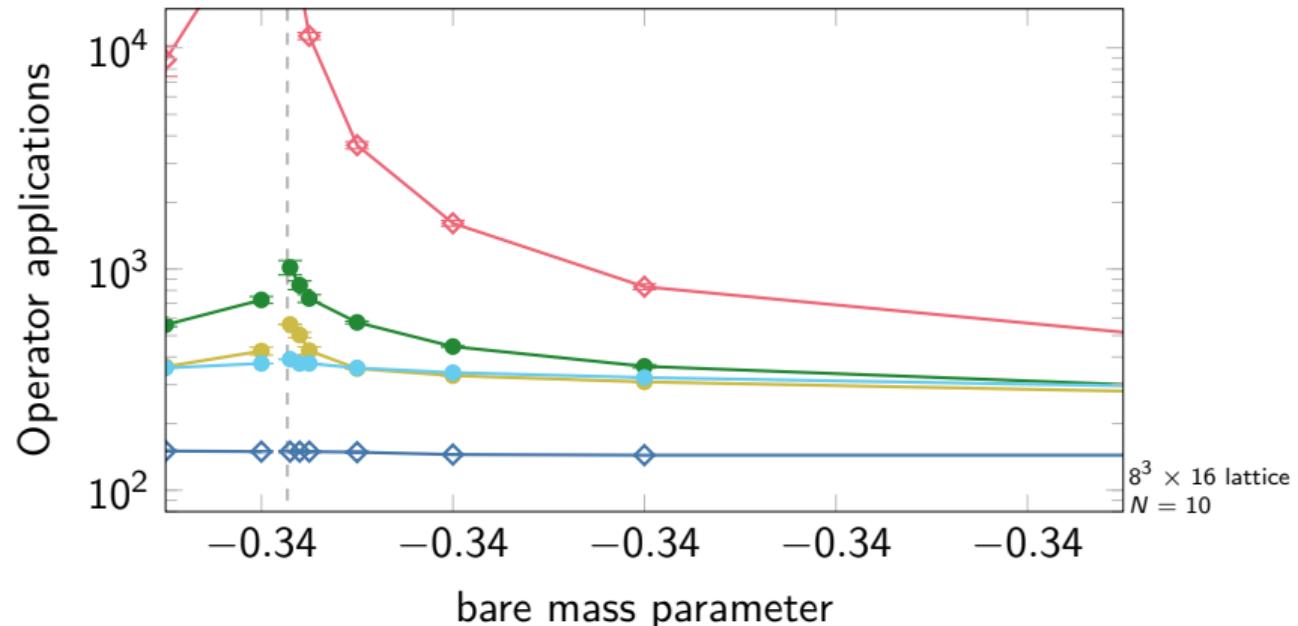
Weights



Laplace operator

Laplace operator
random U(1) links

unpreconditioned MG
4 layers 8 layers
16 layers



Laplace operator

