Matrix-free Neural Preconditioners for the Dirac Equations in Lattice Gauge Theory

Yixuan Sun

Srinivas Eswar, Yin Lin, William Detmold, Phiala Shanahan, Sherry Li, Yang Liu, and Prasanna Balaprakash

Argonne National Laboratory
Massachusetts Institute of Technology
Lawrence Berkeley National Laboratory
Oak Ridge National Laboratory

U.S. DEPARTMENT of ENERGY

Background

Calculations in Lattice quantum field theory requires solving large linear systems

$$D^{\dagger}Dx = b$$

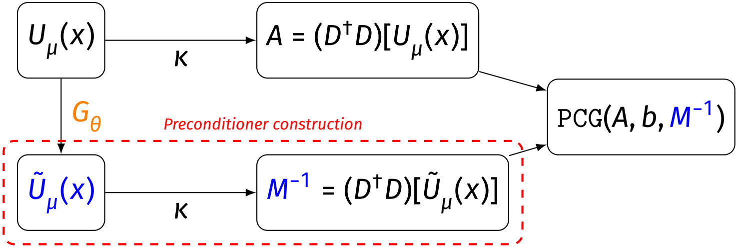
D is the Wilson-Dirac operator ($D^{\dagger}$ is its conjugate transpose).

- Large in size, could potentially go to millions
- Ill-conditioned, causing slow convergence of iteratie solvers

Existing preconditioners

- Require deep understanding of the structure of the linear systems
- Potentially have high setup cost, especially for large systems
- Current neural preconditioners rely on explicit matrices

Matrix-free Neural Preconditioners



- Create another “gauge field configuration”
- Geneate a linear opeartor that approximates the inverse of the original one

Operator Learners

Assumption: mapping is general for gauge fields $U_\mu(x)$ regardless of the size of the underlying physical system.

$$G : \{U_\mu(x)\} \mapsto \{\tilde{U}_\mu(x)\}$$

where $\{U_\mu(x)\}$ and $\{\tilde{U}_\mu(x)\}$ represent sets of lattice gauge field configurations.

$$G_\theta(U_\mu(x)) = \tilde{U}_\mu(x), \quad U_\mu(x), \tilde{U}_\mu(x) \in U(1)^{X \times T \times d} \subset \mathbb{C}^{X \times T \times d}$$

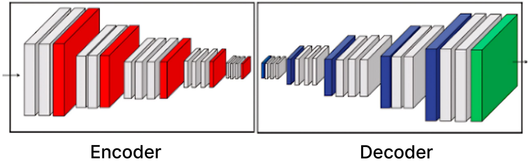
with operator learners of the structure

$$(G_\theta U_\mu)(x) = \left(K_\theta^{(N-1)} + B_\theta^{(N-1)}\right) \circ \sigma \circ \cdots \circ \sigma \circ \left(K_\theta^{(0)} + B_\theta^{(0)}\right)(U_\mu)(x)$$

Operator Learners: FCN

$$(K_\theta u)(x)_{\mathrm{FCN}} = \sum_{\delta \in S} k_\theta(\delta) u(x - \delta)$$

- Fully Convolutional Network
- Handles variable sized inputs
- Kernel integral is spatially *local*

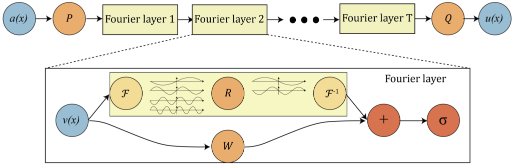


Source: analyticsvidhya.com

Operator Learners: FNO

$$(K_\theta u)(x)_{\mathrm{FNO}} = \sum_{|f|\leq m} k_\theta(f)\hat{u}(f)e^{2\pi i f x}$$

- Fourier Neural Operator
- Also handles variable sized inputs
- Kernel integral is spatially *global*



Source: Li et al., "Fourier Neural Operator for Parametric Partial Differential Equations"

Training the model

Aim to produce a linear operator approximating the inverse of $D^\dagger D$

$$L(\theta) = \frac{1}{N \cdot K} \sum_{i=0}^{N-1} \sum_{j=0}^{K-1} \| M_i(\theta)^{-1} D_i^\dagger D_i v_j - v_j \|_2,$$

- v is a random vector sampled from isotropic Gaussain
- K the number of random vectors, treated as a hyperparameter
- Proxy of the Frobenius norm of the difference matrix between the preconditioned and identity

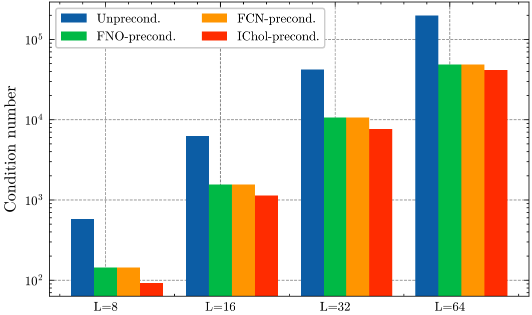
Experiments

Models	κ	β	#train	#val	#test
$N_{L8}, N_{L16}, N_{L32}, N_{L64}$	0.276	2.0	1280	320	200

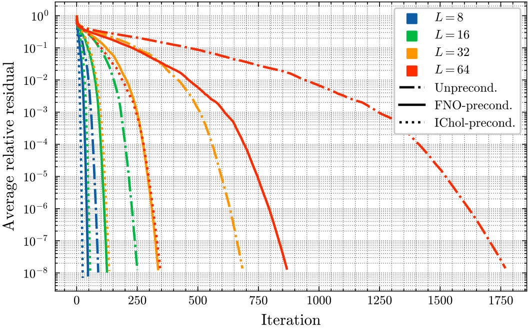
- Action-specific models – trained on fixed action parameters, L, κ, β
- Volume transfer – applying a pretrained model to problems from various sizes and action parameters
- Examine the learned mapping

Action-specific Models

Condition numbers



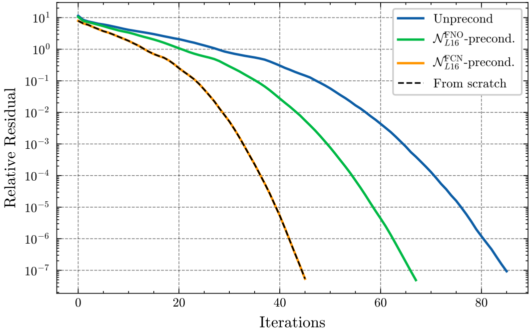
CG iterations



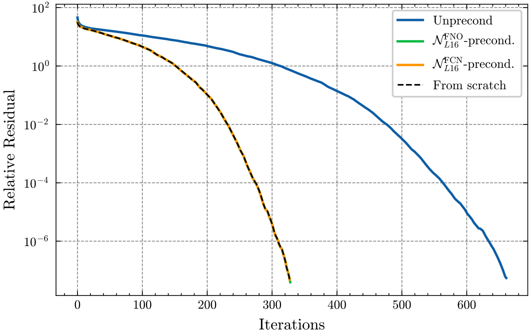
- Both FCN and FNO preconditioners substantially reduce the condition numbers
- Neural network-preconditioned CG takes approxmiately *half* the iterations as the unpreconditioned case

Volume Transfer

N_{L16} on $L = 8$



N_{L16} on $L = 32$

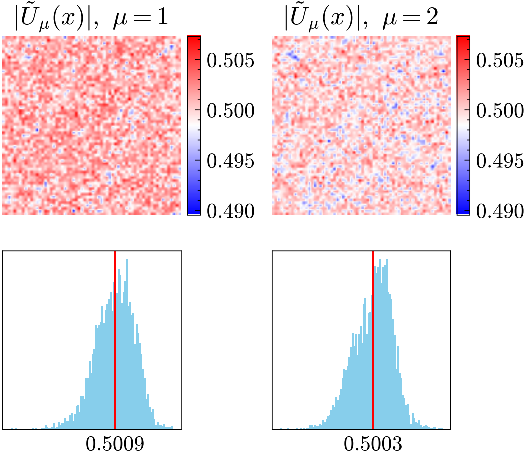


Volume Transfer

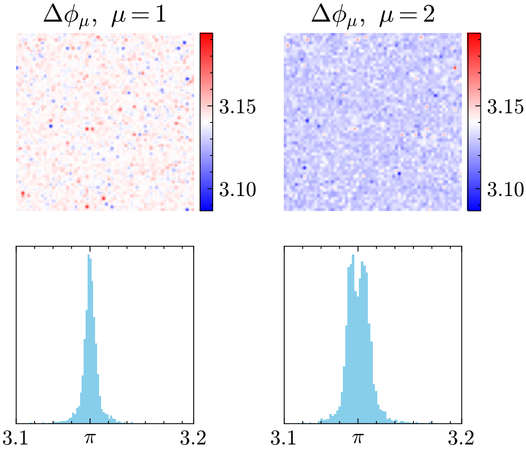
Configuration	Unprecond.	IChol Precond.	FNO_{16}	FCN_{16}
$L = 8,\ \kappa = 0.276,\ \beta = 2.0$	78 ± 4	22 ± 1	60 ± 3	40 ± 2
$L = 8,\ \kappa = 0.276,\ \beta = 1.843$	80 ± 4	23 ± 2	62 ± 3	42 ± 3
$L = 8,\ \kappa = 0.260,\ \beta = 2.0$	76 ± 2	21 ± 1	59 ± 2	40 ± 1
$L = 16,\ \kappa = 0.276,\ \beta = 2.0$	201 ± 14	44 ± 4	99 ± 7	99 ± 7
$L = 16,\ \kappa = 0.276,\ \beta = 3.124$	166 ± 10	33 ± 2	78 ± 5	78 ± 5
$L = 32,\ \kappa = 0.276,\ \beta = 2.0$	548 ± 41	111 ± 9	267 ± 21	267 ± 21
$L = 32,\ \kappa = 0.276,\ \beta = 5.555$	260 ± 19	44 ± 3	117 ± 9	117 ± 9
$L = 64,\ \kappa = 0.276,\ \beta = 2.0$	1540 ± 91	300 ± 17	719 ± 47	719 ± 47

Learned Mappings

Magnitude of $\tilde{U}_\mu(x)$



Phase difference $\Delta\phi \in [0, 2\pi)$



Learned Mappings

$$\tilde{U}_\mu^{\text{simple}}(x) = -\frac{1}{2}U_\mu(x)$$

	$L = 8$	$L = 16$	$L = 32$
Unprecond.	78 ± 4	201 ± 14	548 ± 41
$U_\mu^{\text{simple}}(x)$-precond.	43 ± 2	104 ± 7	278 ± 20
N_L^{FNO}-precond.	40 ± 2	99 ± 7	267 ± 21

$$D^\dagger D[-0.5U_\mu(x)] = |1 + \kappa H|^2 \underset{\kappa\to 0}{\simeq} |1 + 2\kappa H| \simeq |1 - 2\kappa H|^{-1} \simeq (D^\dagger D[U_\mu(x)])^{-1}$$

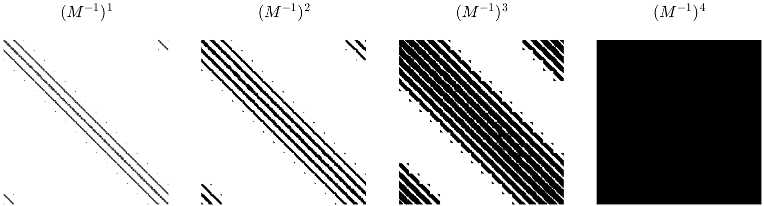
Remarks

The trained neural preconditioners *avoid setup time and a linear solve* per iteration, thus have an advantage when

- Given problem has many unique linear operators but fewer right hand sides
- Problem size becomes over costly or prohibitive for IChol preconditioner construction

Limitation

Structure limits the inverse approximation



- Same sparstiy prevents close inverse approximation
- Higher powers density the operator, showing improvement, but still limited

Conclusion and Future work

Our framework

- Provides effective preconditioners for solving Wilson-Dirac normal equations
- Is completely *matrix-free* during both training and application
- Uses effective and efficient loss in model training, avoiding expensive condition number computation
- Learns a general mapping effective for *unseen* lattice ensembles of different parameters and sizes, with no performance degradation

Future work:

- Develop a more theorectically grounded training objective to improve preconditioners
- Extend this framework to higher dimensional problems in $SU(2)$ and $SU(3)$

Thank you!