

Updates on the Machine Learning Approach for Lattice Gauge Fixing

2026.07.29 @ Lattice 2026, University of Maryland



Ho Hsiao ^{シヤオ}蕭 ^{ホウ}禾, Benjamin J. Choi 崔 在敦,
Hiroshi Ohno 大野浩史



京都大学



Akio Tomiya 富谷昭夫

Outline

- Lattice gauge fixing
 - Traditional iterative approaches
- Recap: a machine learning approach for gauge fixing
 - Gauge transformation and Convolutional neural network construction
 - Preliminary results with a small model
- Toward deep CNN & Preliminary results
- ❖ Summary and Outlook

Lattice Gauge Fixing

- Gauge fixing is required for gauge-dependent quantities:

- gluon propagator & quark-gluon vertex

[J.E. Mandula, M. Ogilvie, Phys.Lett.B 185(1987)]

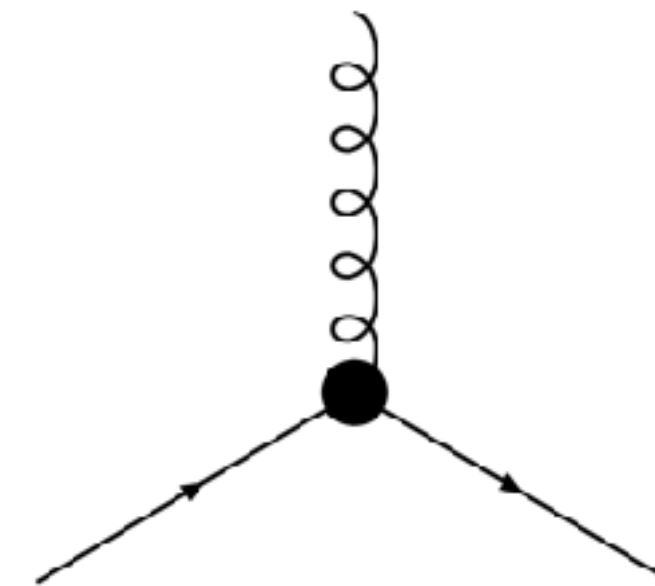
- renormalisation schemes

[L. Maiani, G. Martinelli, C.T. Sachrajda, Nucl.Phys.B 368 (1992)]

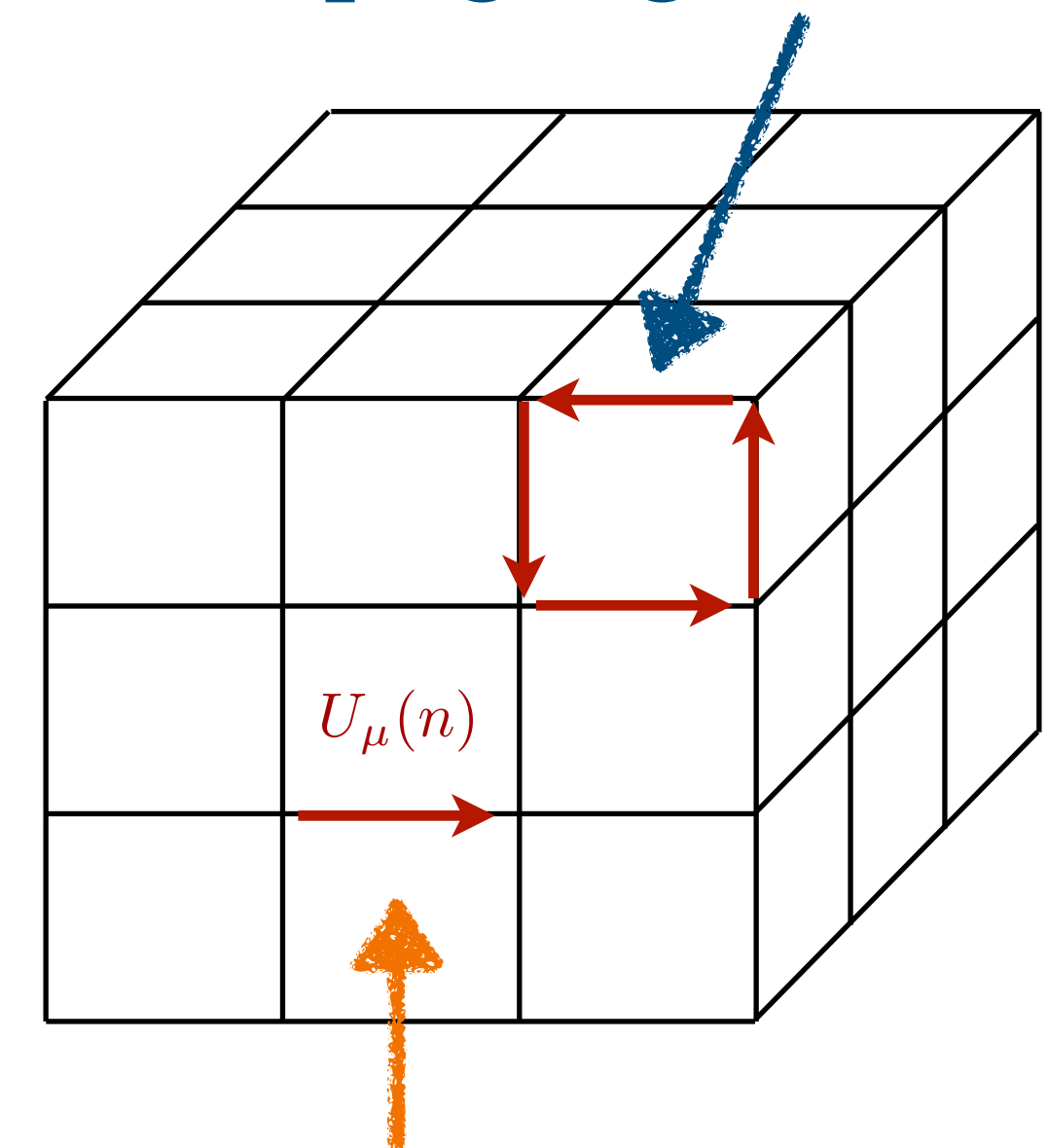
- fundamental QCD Green's functions

- Noise reduction

[CP-PACS Collaboration, S. Aoki, et al, Phys.Rev.D 67 (2003), 034503]



closed loop: gauge-invariant



open loop: gauge-dependent

Lattice Gauge Fixing

- Gauge fixing condition:

$$F[g] = \mathbb{R} \text{Tr} \sum_x \sum_{\mu}^{d_{\text{fix}}} U_{\mu}^g(x), \quad \text{with} \quad \frac{\delta F}{\delta g} = 0$$

where d_{fix} is 3 for the Coulomb and 4 for the Landau gauge.

- Iterative approach:

- Los Alamos (LA) method: SU(2) subgroup update

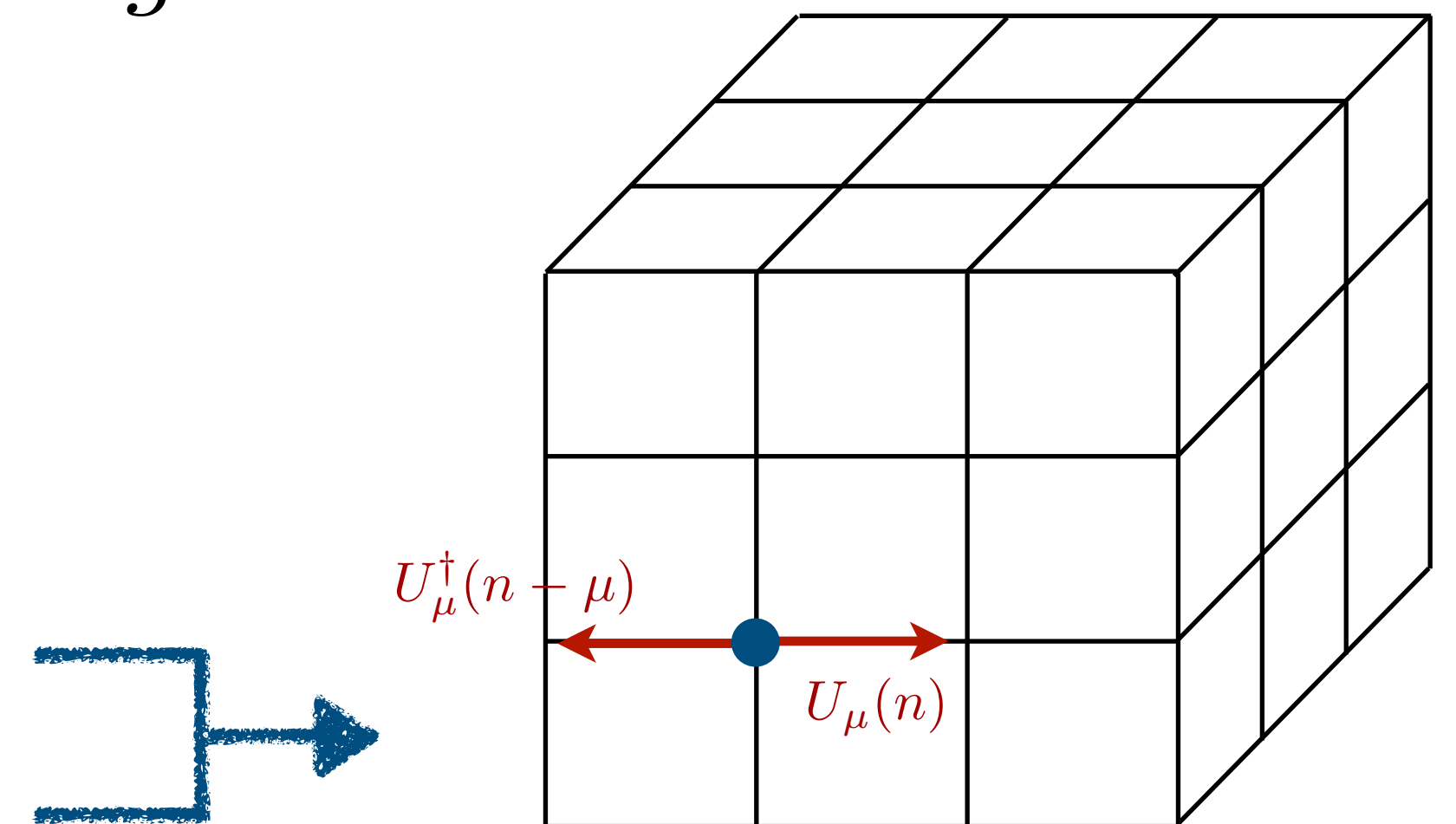
[R.Gupta, G.Guralnik, G.Kilcup, A.Patel, S. R. Sharpe, and T.Warnock, Phys.Rev. D36 (1987) 2813.]

- Cornell method: Steepest descent (SD)

[C.T.H.Davies, et al., Phys.Rev D37 (1988) 1581]

- Advanced techniques: Fourier acceleration, conjugate gradient, and global optimization...

[A. Cucchieri, T. Mendes, PRD 57 (1998) 3822] [R.J. Hudspith, Comput. Phys. Commun. 187 (2015) 115] [O.Oliveira and P.J. Silva, Comput. Phys. Commun. 158 (2004) 73]

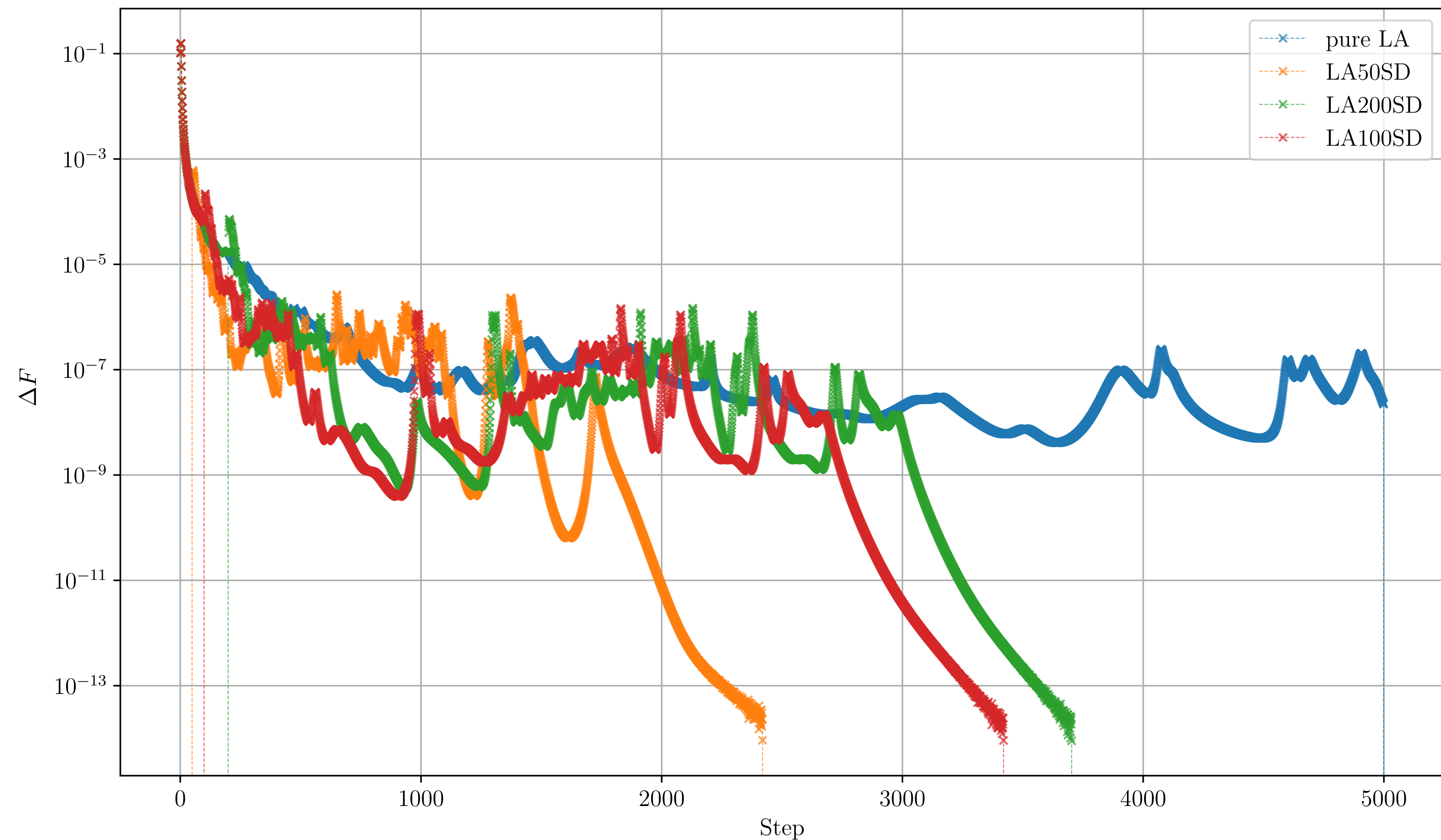


Lattice Gauge Fixing

Issues of iterative approaches

- Volume scaling bottleneck
- Critical slowing down:

Applying the Los Alamos method first, then steepest descent until reaching the tolerance 10^{-14} for a single configuration with the size of $12^3 \times 24$.

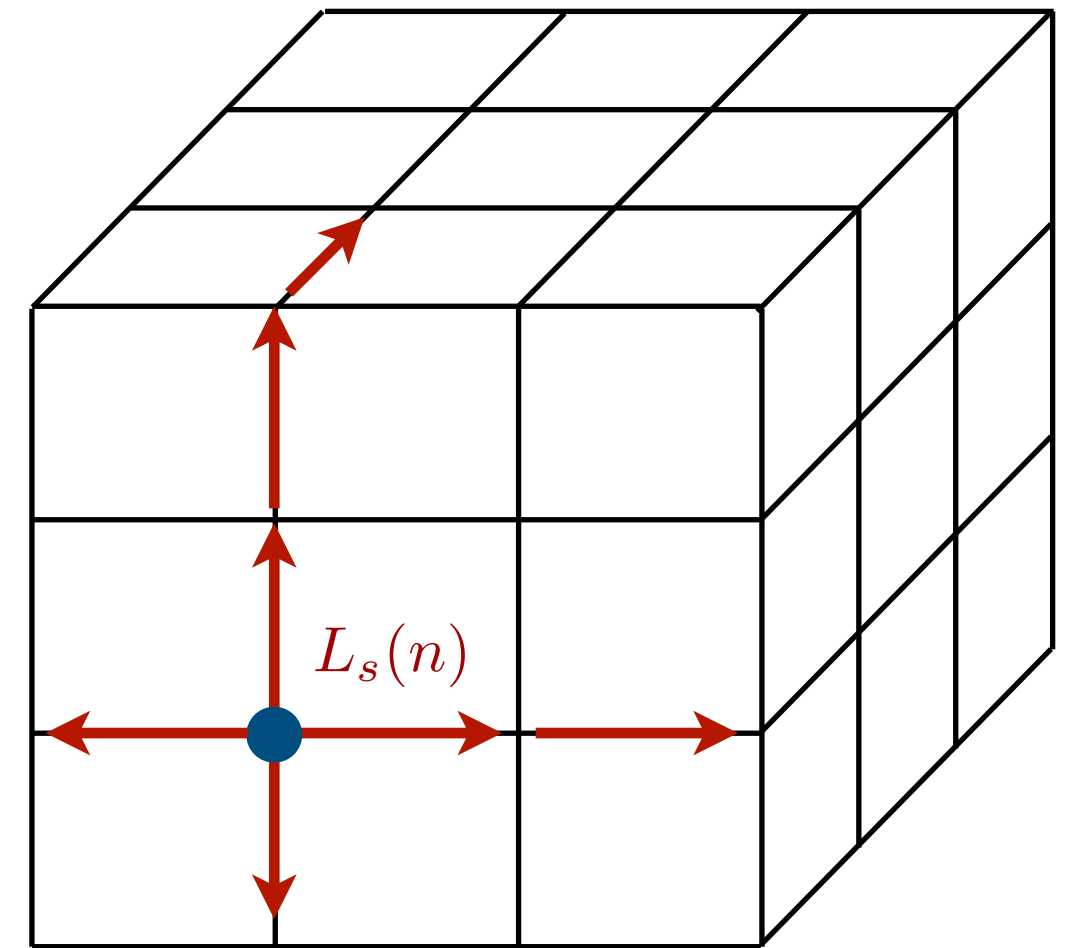


Our model: Machine learning approach

Basic idea

Try to construct a gauge transformation that can replace/reduce the iterative procedure of the gauge fixing by

- Extending from a single link to Wilson lines with multiple lengths $L_s(n)$.
- Regarding the projection of the SU(3) group as a non-linear operation—activation function.
- Constructing complicated Wilson lines via a Convolutional Neural Network (condensing the information into layers).
- Optimising the tunable parameters by machine learning.



Our model: Machine learning approach

Network construction

- The WLB gauge transform matrix is defined as

$$g^{(\ell)}(n) = \exp \left[Q^{(\ell)}(n) \right] \in SU(N_c = 3)$$

$$Q^{(\ell)}(n) = \left[\Omega^{(\ell)}(n) \right]_{\text{TA}} = \frac{1}{2} \left(\Omega^{(\ell)\dagger}(n) - \Omega^{(\ell)}(n) \right) - \frac{1}{2N_c} \text{Tr} \left(\Omega^{(\ell)\dagger}(n) - \Omega^{(\ell)}(n) \right) I$$

$$\Omega^{(\ell)}(n) = \sum_{r=1}^s \theta_r^{(\ell)} \underline{L_r \left(n; \left\{ U^{(\ell-1)} \right\} \right)}$$

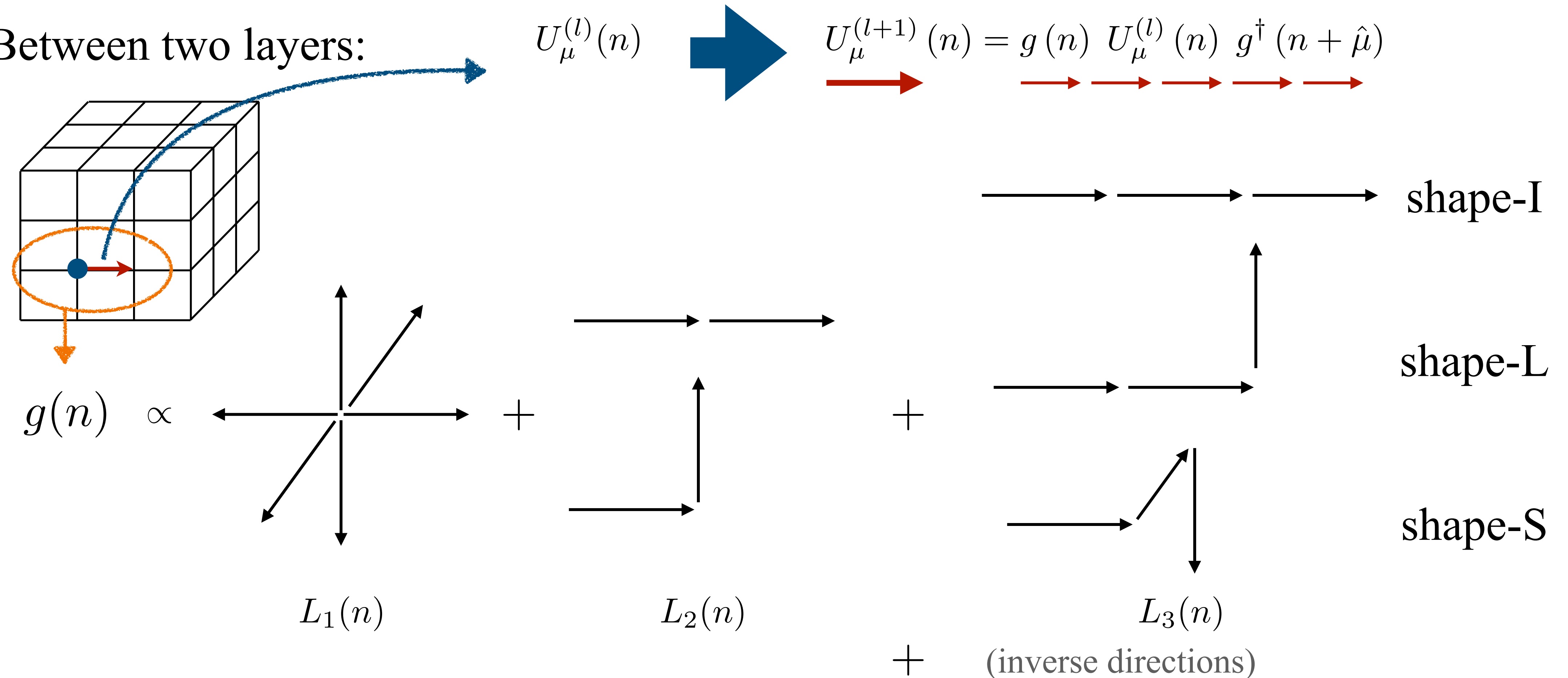
Wilson Line Bundles (WLB)

Training parameters

Our model: Machine learning approach

Network construction

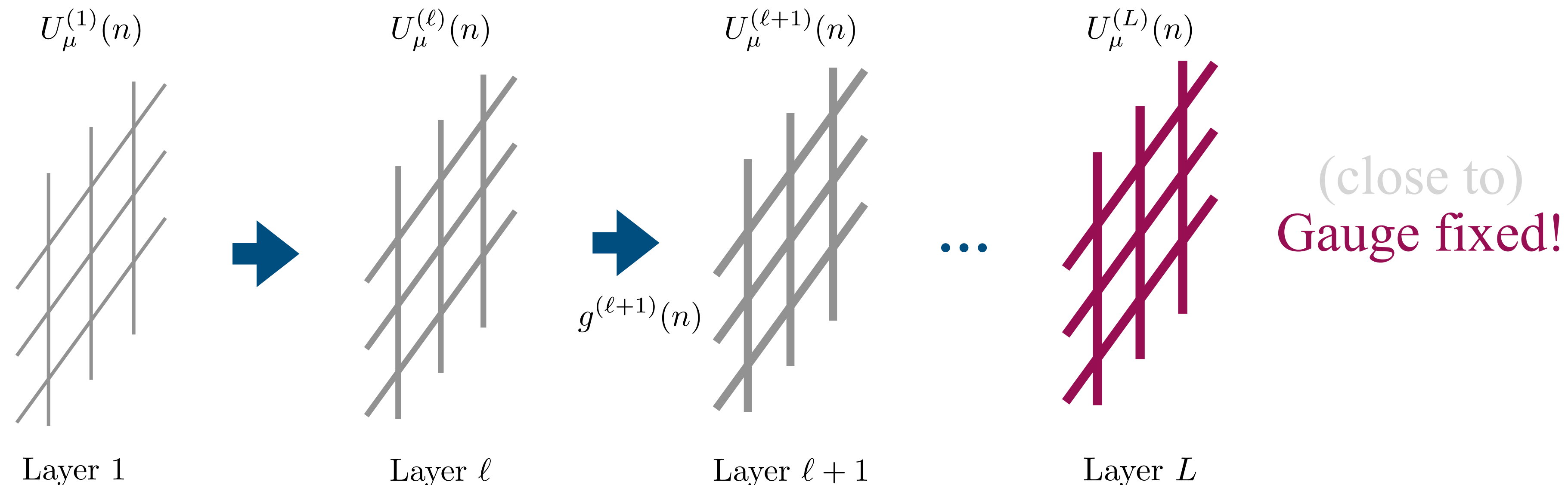
- Between two layers:



Our model: Machine learning approach

Network construction

- Turning iterations into finite layers of a CNN
- Constructing complicated Wilson lines via a CNN

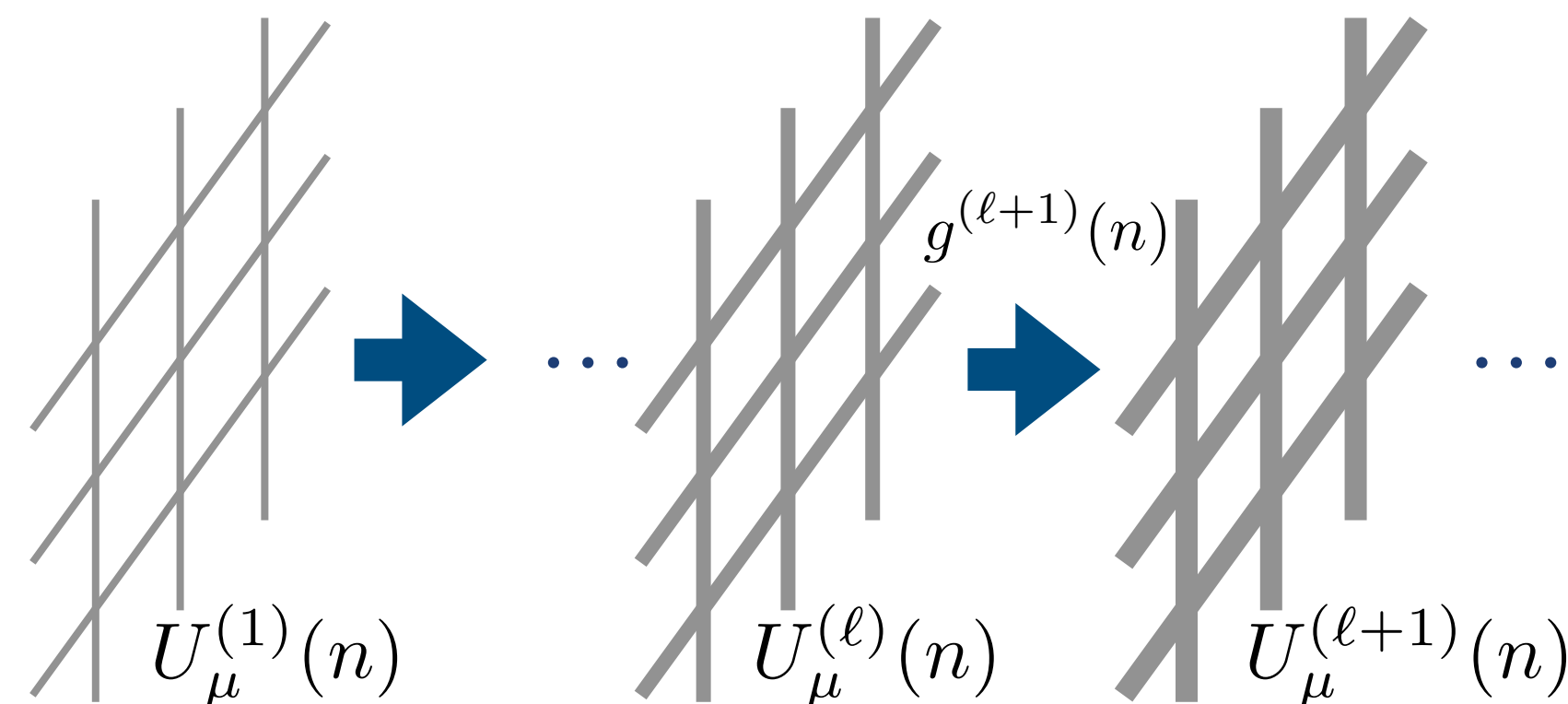


Our model: Machine learning approach

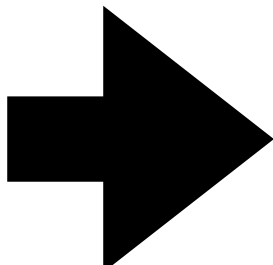
backpropagation

- The objective function (optimum functional):

$$\mathcal{F}(\theta) = \frac{1}{d_{\text{fix}} N_c N_{\text{vol}}} \text{Re} \sum_n \sum_{\mu=1}^{d_{\text{fix}}} \text{Tr} U_{\mu}^{(L)}(n)$$



where N_c is the number of color, and N_{vol} is number of the lattice sites

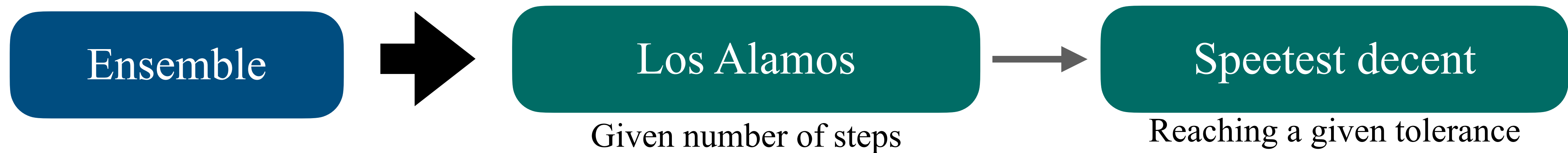
- Training gradient  **Mini-batch training with Adam**

$$\frac{d\mathcal{F}(\theta)}{d\theta_r^{(\ell)}} = 2 \sum_{\mu=1}^{d_{\text{fix}}} \sum_n \text{Re} \text{Tr} \left[\Lambda_{\mu}^{H^{(\ell)}}(n) L_r \left(n; \left\{ U^{(\ell-1)} \right\} \right) + \Lambda_{\mu}^{S^{(\ell)}}(n + \hat{\mu}) L_r \left(n + \hat{\mu}; \left\{ U^{(\ell-1)} \right\} \right) \right]$$

Gauge fixing procedure

Workflow

- Pure traditional iterative approach:



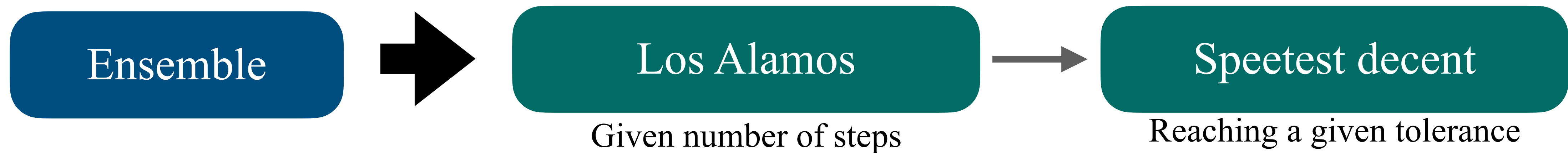
- Machine learning hybrid approach:



Gauge fixing procedure

Workflow

- Pure traditional iterative approach:



- Machine learning hybrid approach:



- JLDG public ensembles:

Ensemble	Lattice	β	κ_{ud}	κ_{s}	$a[\text{fm}]$	F_{fixed}	size[GB]
RC12x24	$12^3 \times 24$	1.8	0.14090	-	0.2150(22)	0.8213(5)	0.023
RC16x32	$16^3 \times 32$	1.83	0.13655	0.13710	0.1209(16)	0.8529(4)	0.072
RC20x40	$20^2 \times 40$	1.9	0.13580	0.13580	0.0982(19)	0.8656(2)	0.176
RC32x48	$32^3 \times 48$	1.9	0.1373316	0.1367526	0.0907(13)	0.8656(2)	0.844
RC48x48	$48^3 \times 48$	1.9	0.1373316	0.1367526	0.0907(13)	0.86966(8)	2.848
RC64x64	$64^3 \times 64$	1.9	0.1373316	0.1367526	0.0907(13)	0.86978(4)	9

- PACS10_c: 128^4 (144 GB)

- HPC Machine:

- Pegasus (Intel Xeon Platinum-NVIDIA H100)
- Miyabi (NVIDIA GH200 Grace Hopper)
- Fugaku (CPUs)

Preliminary results

Preliminary results

- Defination of observables in gauge fixing section:

$$F[g] = \frac{1}{d_{\text{fix}} N_c N_{\text{vol}}} \text{Re} \sum_n \sum_{\mu}^{d_{\text{fix}}} \text{Tr} U_{\mu}^g(n) \rightarrow \text{finite value, } F_{\text{fixed}}$$

$$\sigma[g] = \frac{1}{d_{\text{fix}} N_c N_{\text{vol}}} \sum_n \text{Tr} \Delta(n) \Delta^{\dagger}(n) \rightarrow \text{Zero}$$

$$\Delta(n) = \sum_{\mu}^{d_{\text{fix}}} [A_{\mu}(n) - A_{\mu}(x - \hat{\mu})] , \text{ with } A_{\mu}(n) = \left[\frac{U_{\mu}(n) - U_{\mu}^{\dagger}(n)}{2ag_0} \right]_{\text{TA}}$$

Preliminary results

- Definition of observables for training:

- Objective function:

$$\mathcal{F}(\theta) = \frac{1}{d_{\text{fix}} N_c N_{\text{vol}}} \text{Re} \sum_n \sum_{\mu=1}^{d_{\text{fix}}} \text{Tr} U_{\mu}^{(L)}(n)$$

At final layer



- Change of objective function:

$$\Delta \mathcal{F}(\theta) = \mathcal{F}(\theta_i) - \mathcal{F}(\theta_{i-1})$$

- Shifted objective function:

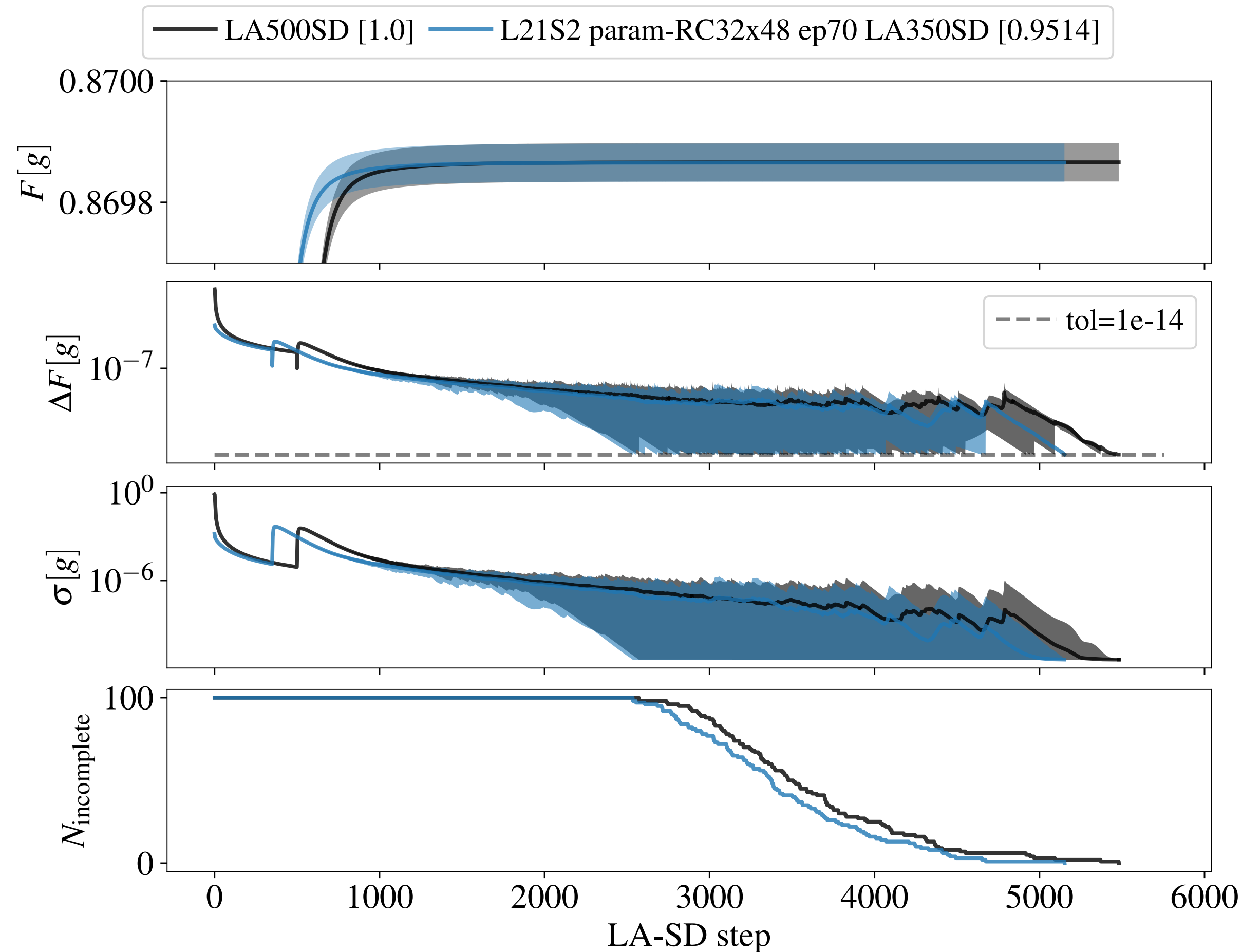
$$\mathcal{L}(\theta) = F_{\text{fixed}} - \mathcal{F}(\theta)$$

- Validation function:

$$\mathcal{V}(\theta) = \frac{1}{d_{\text{fix}} N_c N_{\text{vol}}} \sum_n \text{Tr} \Delta^{(L)}(n) \Delta^{(L),\dagger}(n)$$

Preliminary results

RC64x64: using parameters from RC32x48



An exemplified application of using the training parameters of L12S3, training with the ensemble RC32x48, which shares the same lattice parameters as those of RC64x64.

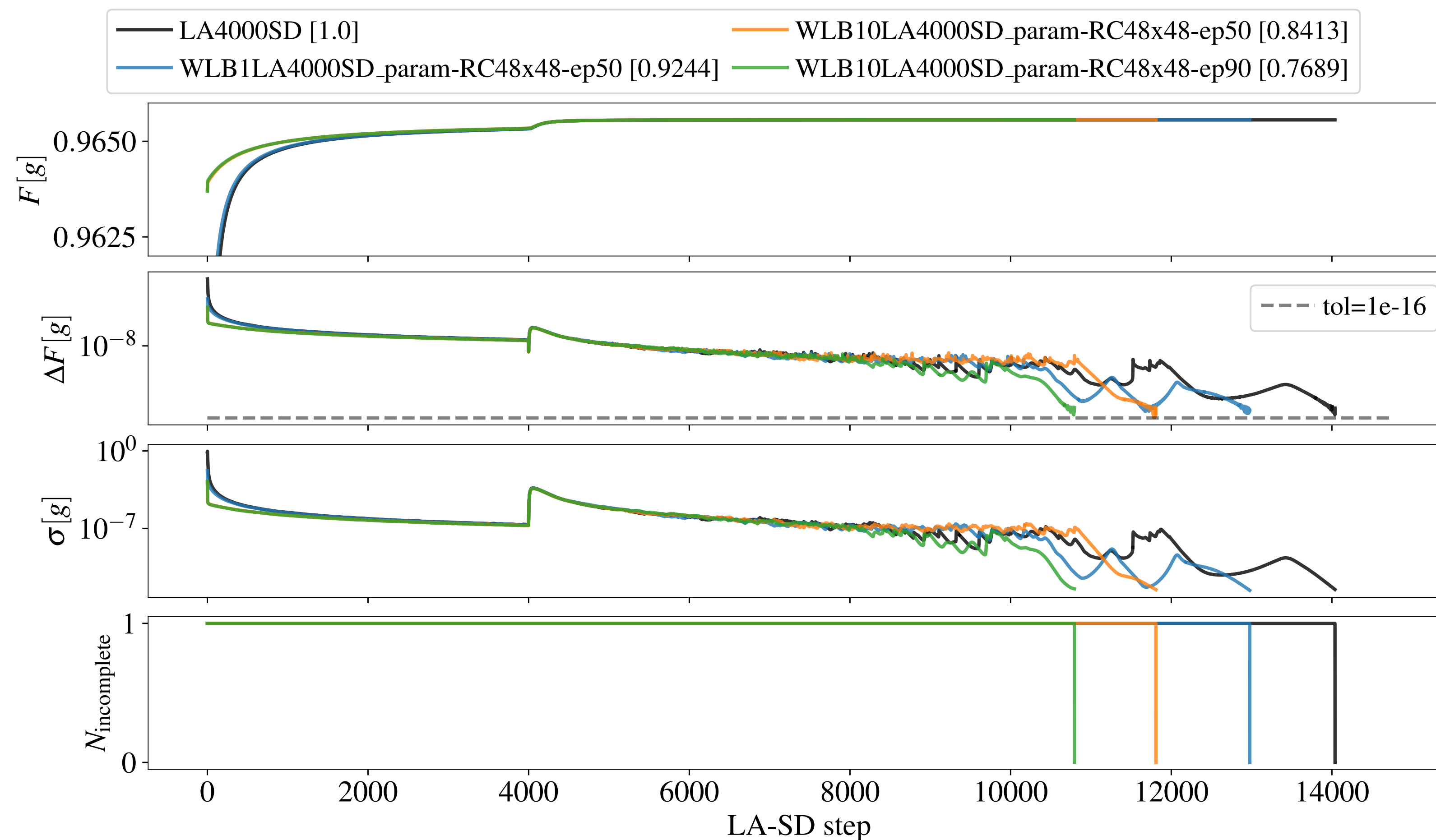
► Training with a smaller lattice for a larger one might be possible.

◆ Timing

- Loading: 3.3%
- WLB transformation: 1.6%
- Iterative gauge fixing: 95%

Preliminary results

Single large configuration test with parameters trained on RC48x48



- Lattice parameters (PACS10_c):

- 128^4
- $\beta = 1.68$
- $\kappa_{\text{ud}} = 0.1269105$
- $\kappa_{\text{s}} = 0.125715$
- $\kappa_{\text{c}} = 0.111150$
- $a = 0.084[\text{fm}]$

► 5 hours with 128 GPUs

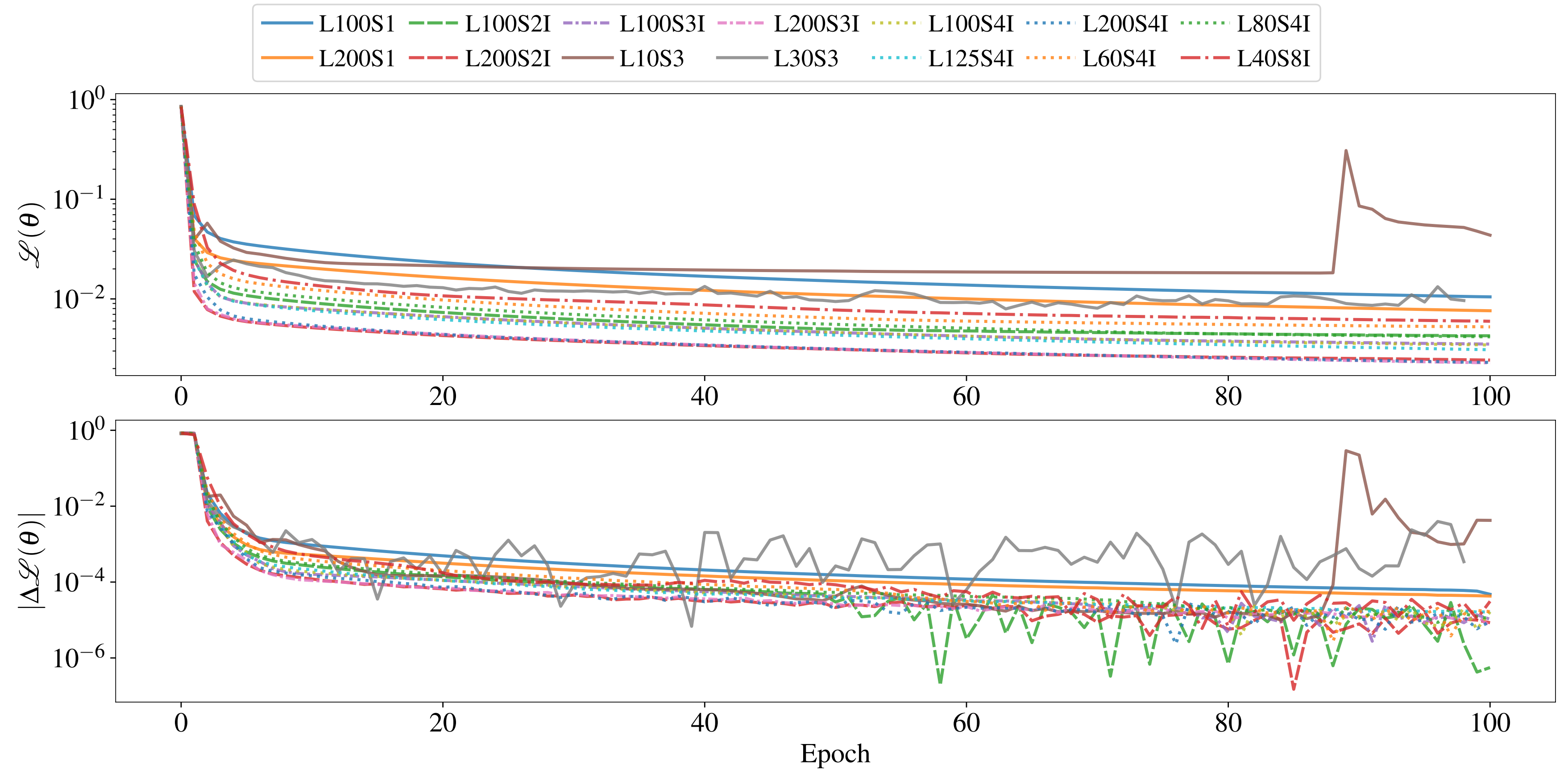
Preliminary results

Model scanning toward deep network

- Training histories
 - 100 epochs
 - 200 configurations
 - RC16x32 ensemble

❖ CNN models

- L[# layers]
- S[max. length]
- I: shape-I only



❖ Shifted objective function: $\mathcal{L}(\theta) = F_{\text{fixed}} - \mathcal{F}(\theta)$

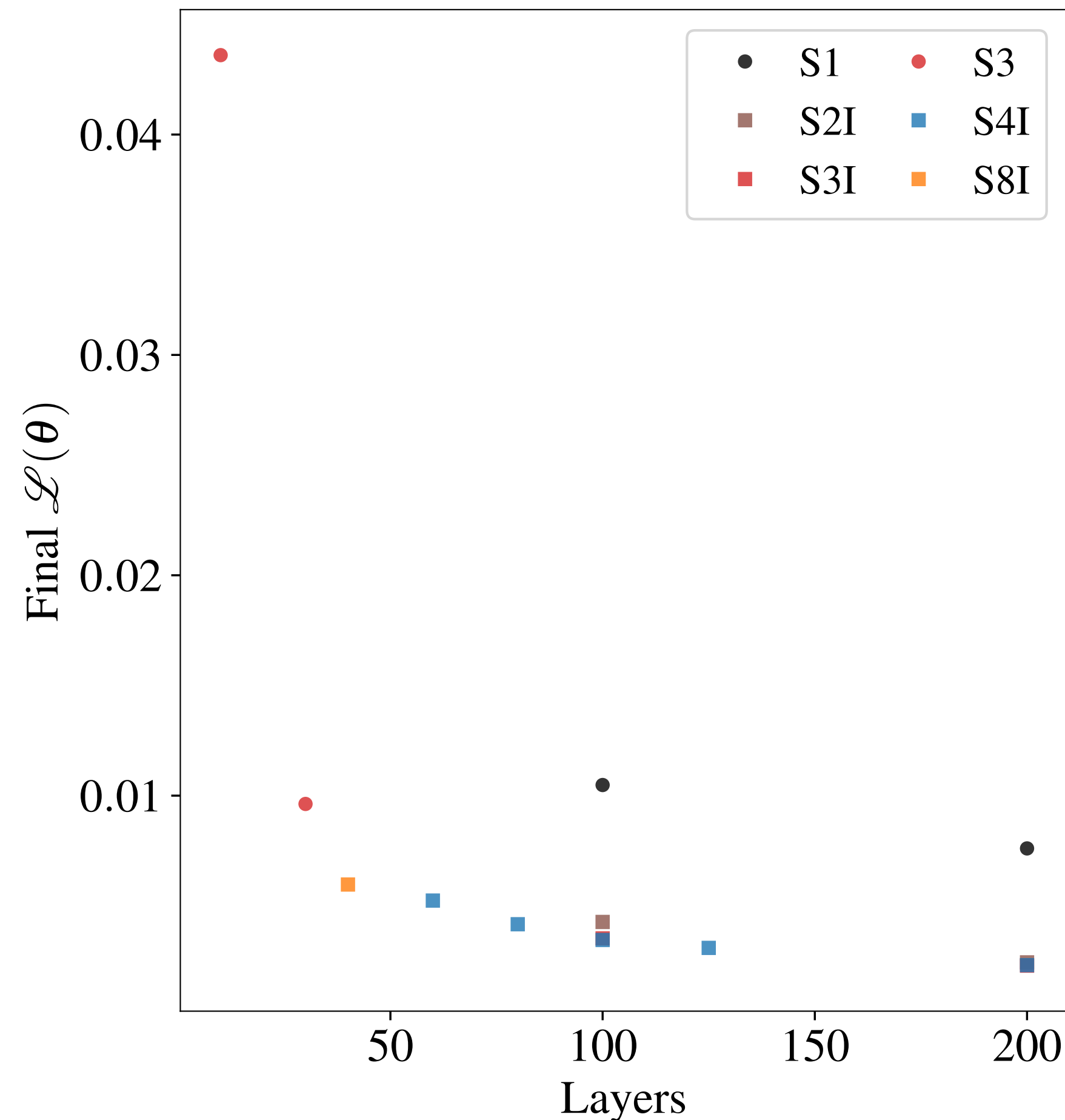
Preliminary results

Model scanning toward deep network

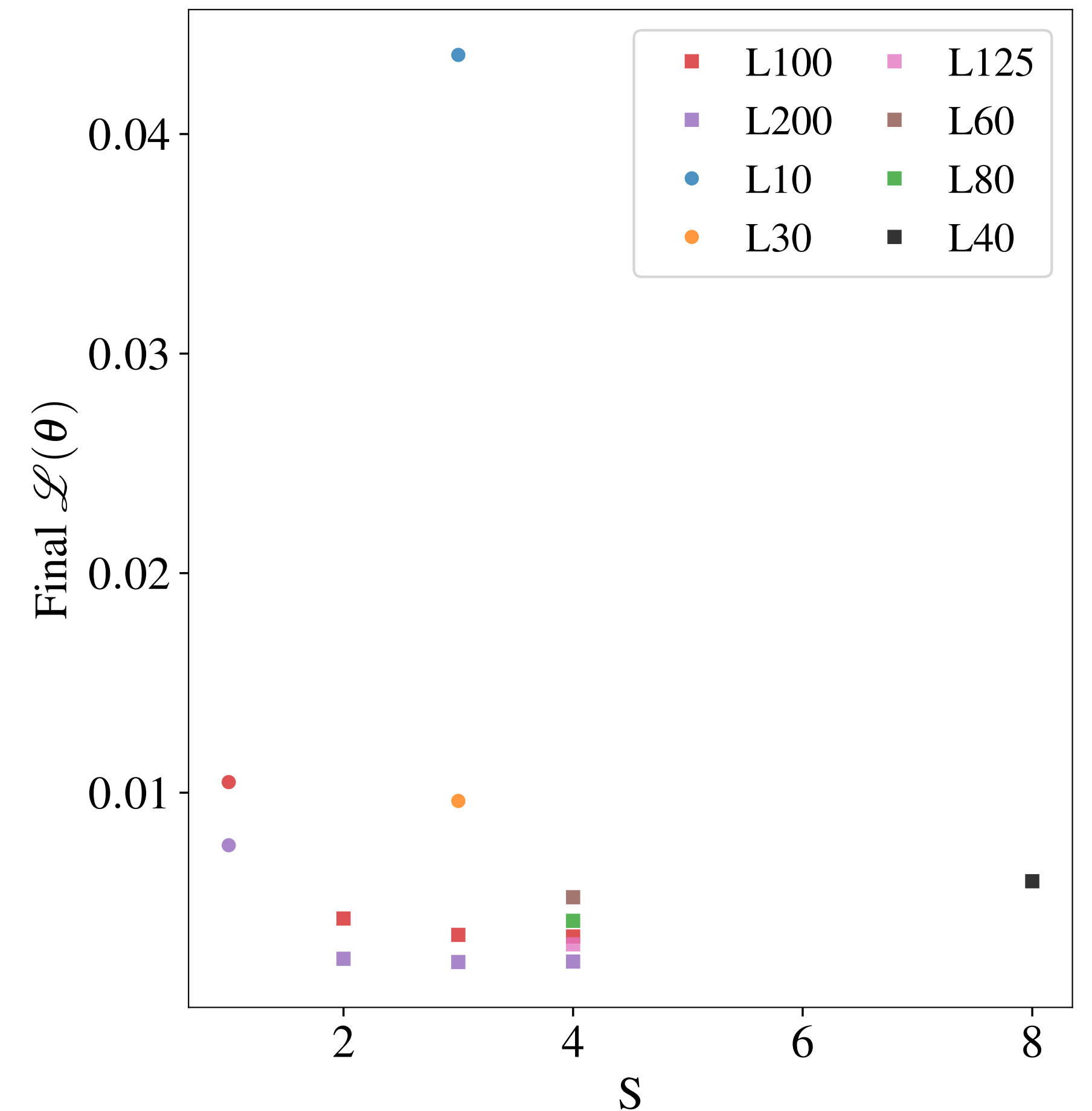
- Comparing the shifted training objective function

❖ CNN models

- L[# layers]
- S[max. length]
- I: shape-I only



❖ Shifted objective function: $\mathcal{L}(\theta) = F_{\text{fixed}} - \mathcal{F}(\theta)$



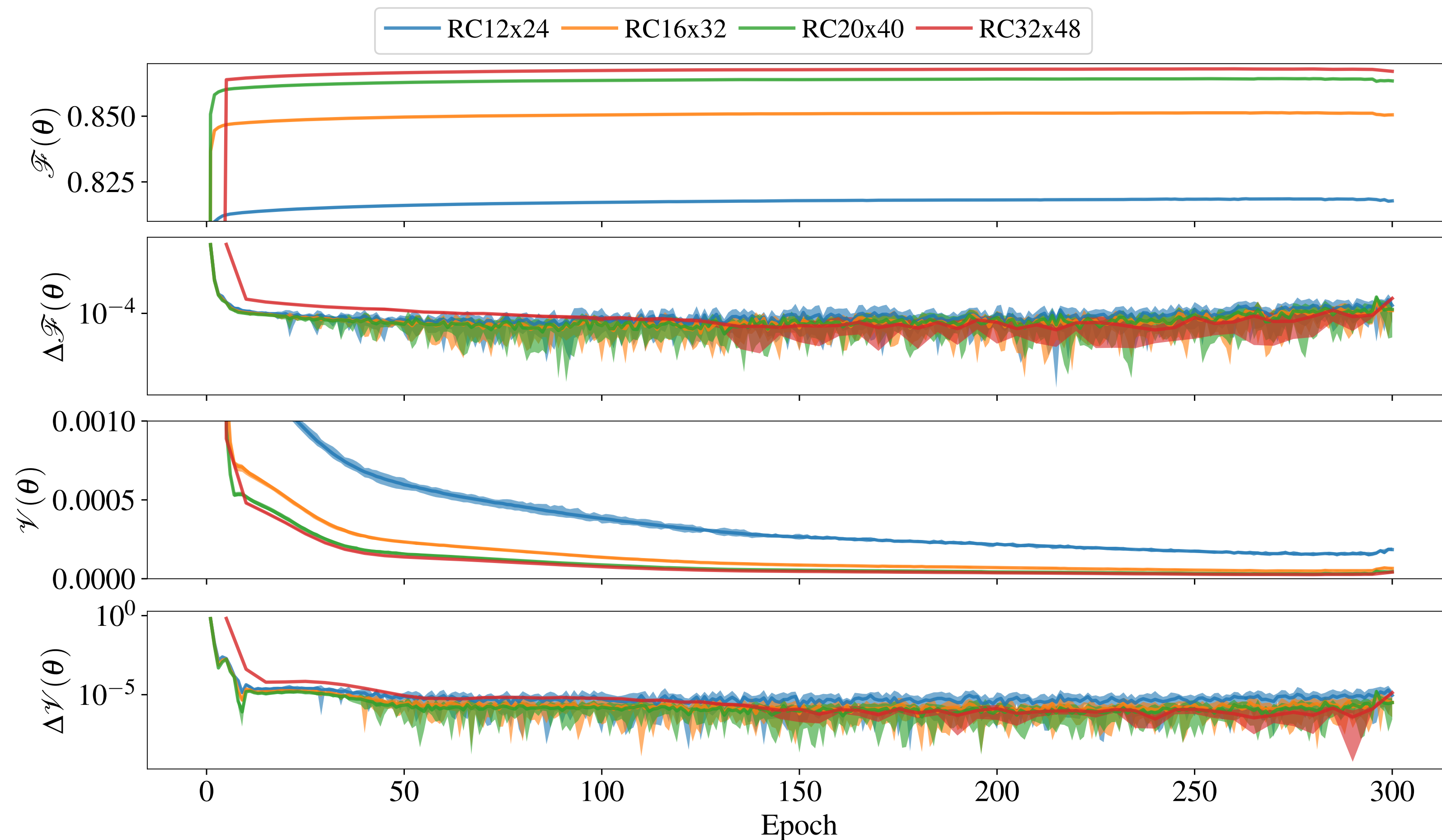
Preliminary results

Lattice size dependence

- Validating the training parameters with ensemble RC16x32 at various lattice sizes over the training history of the L200S3I model.

♣ objective function: $\mathcal{F}(\theta) = \frac{1}{d_{\text{fix}} N_c N_{\text{vol}}} \text{Re} \sum_n \sum_{\mu=1}^{d_{\text{fix}}} \text{Tr} U_{\mu}^{(L)}(n)$

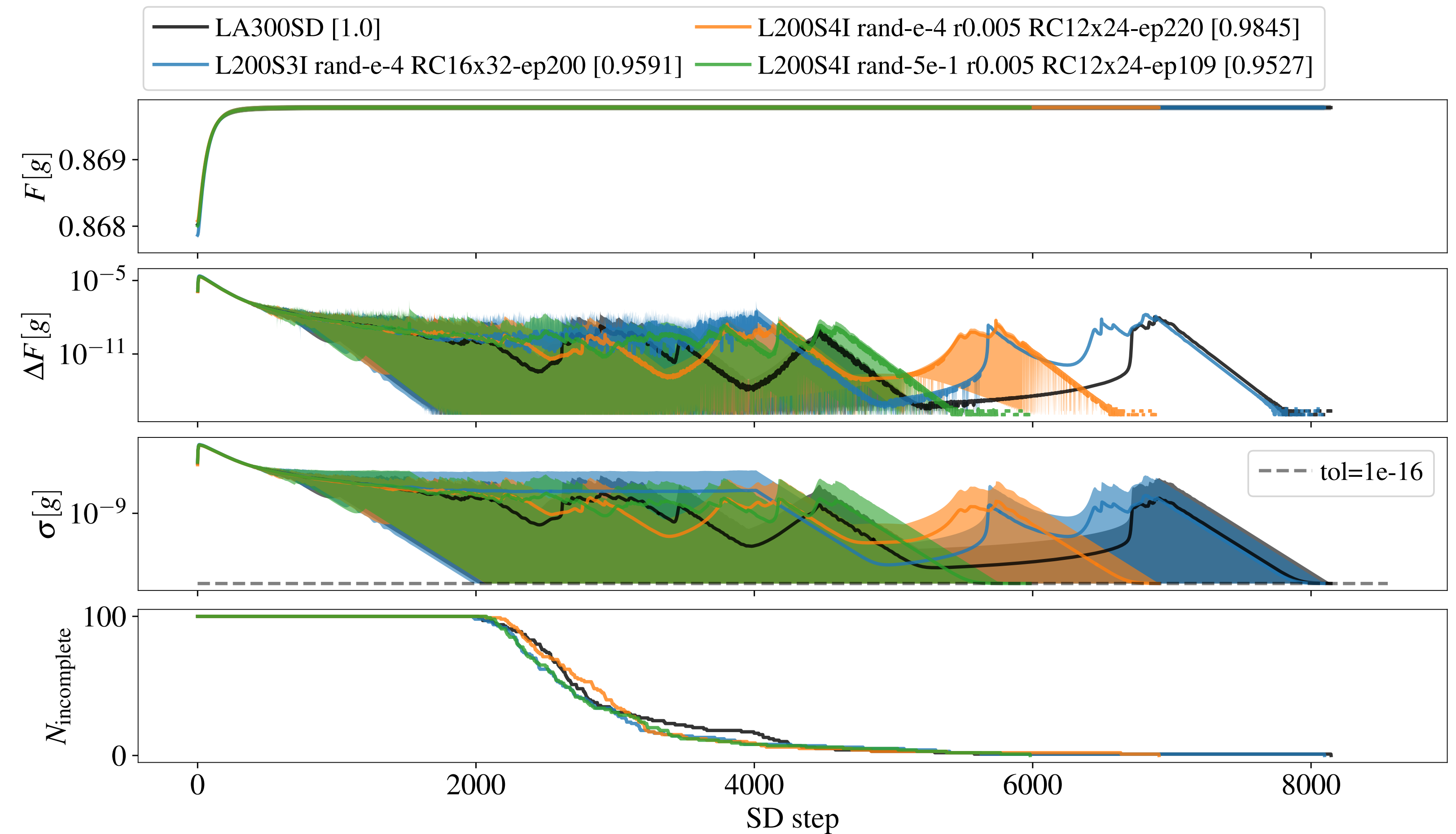
♣ Validation function: $\mathcal{V}(\theta) = \frac{1}{d_{\text{fix}} N_c N_{\text{vol}}} \sum_n \text{Tr} \Delta^{(L)}(n) \Delta^{(L),\dagger}(n)$



Preliminary results

Model performance of a deep network

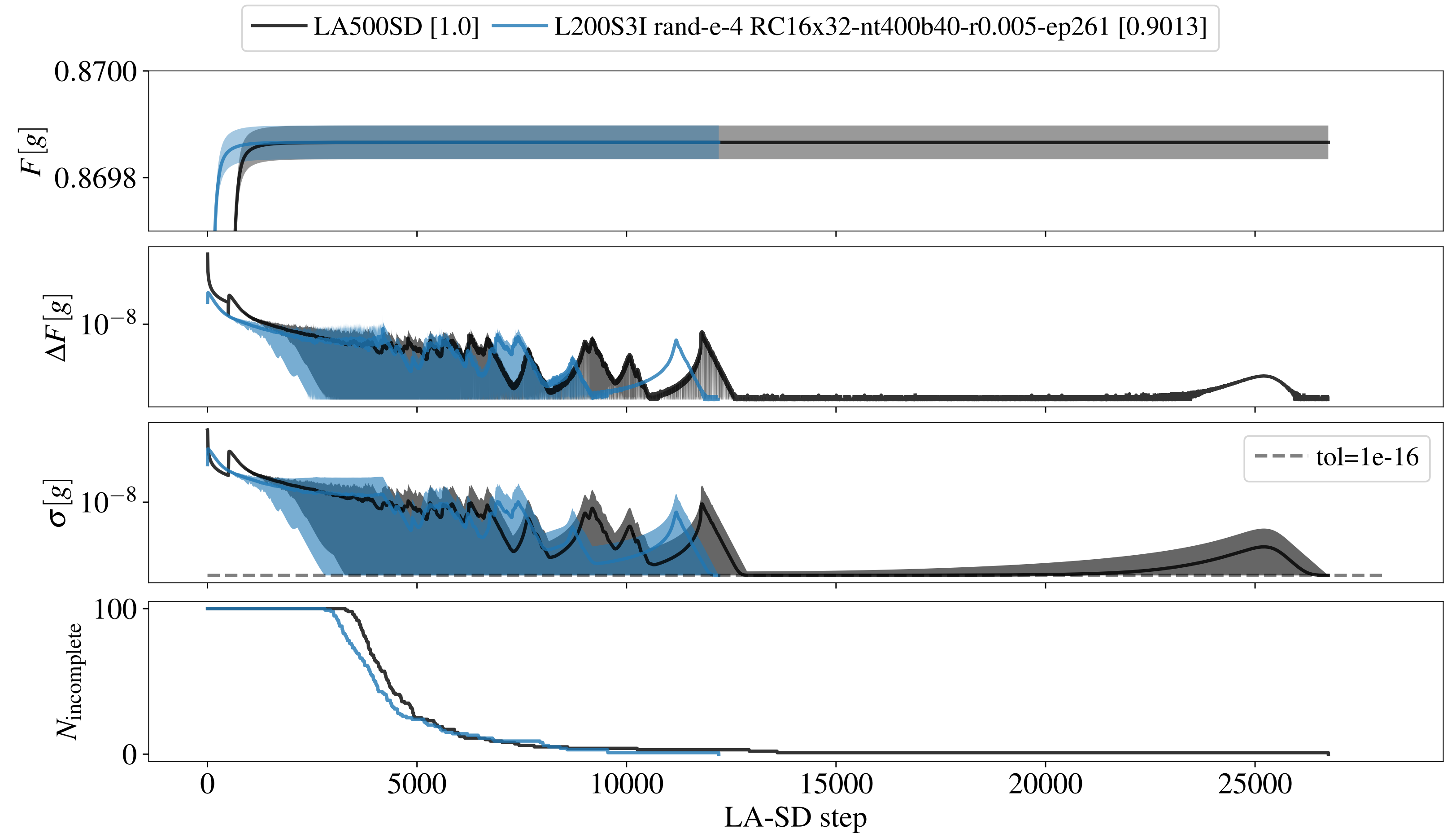
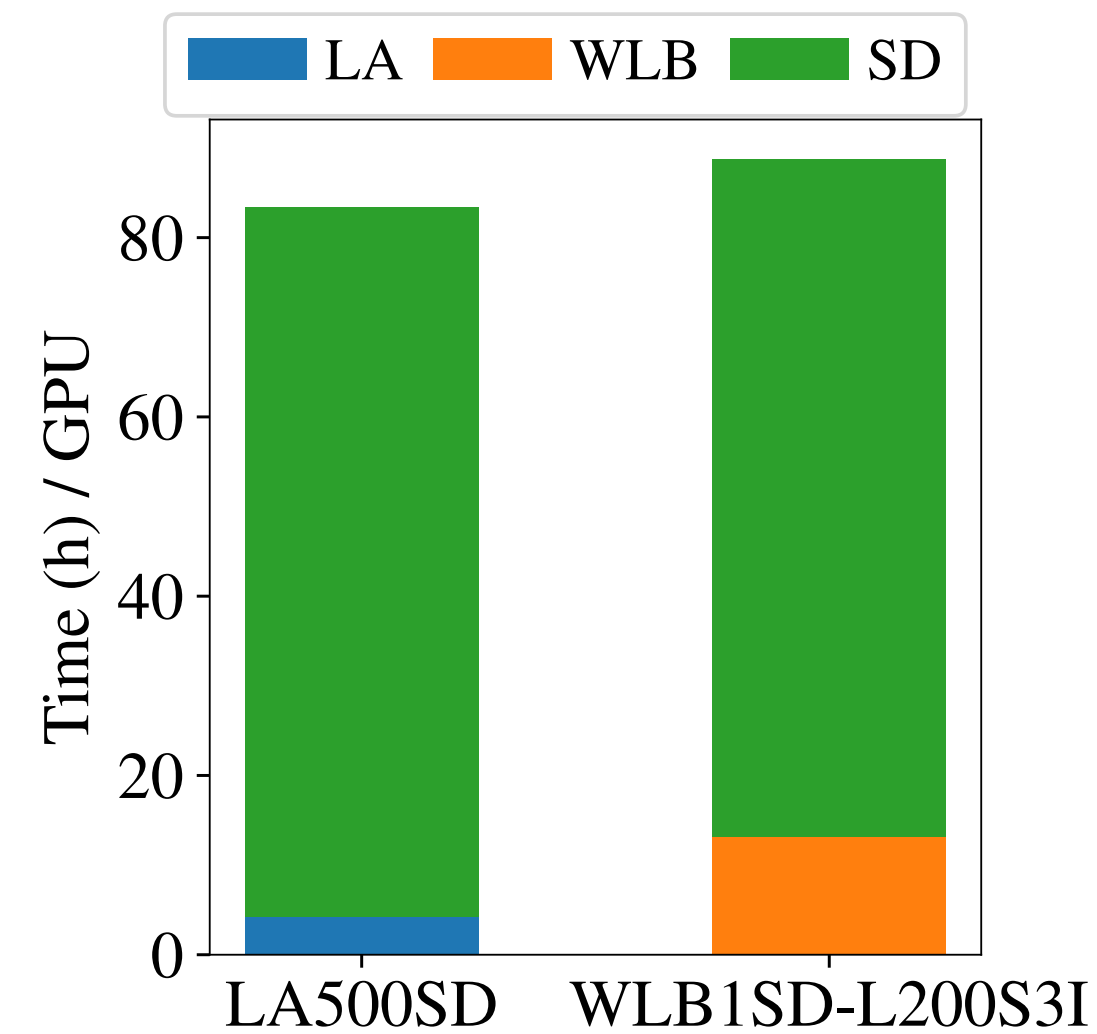
- Gauge fixing 64^4 lattice
 - CNN with 200 layers
- ➡ no LA iterations



Preliminary results

Model performance of a deep network

- Gauge fixing 48^4 lattice
 - CNN with 200 layers
- ➡ no LA iterations



Summary and Outlook

Machine learning approach with the CNN constructed by the Wilson-line bundle



The number of iterations is reduced



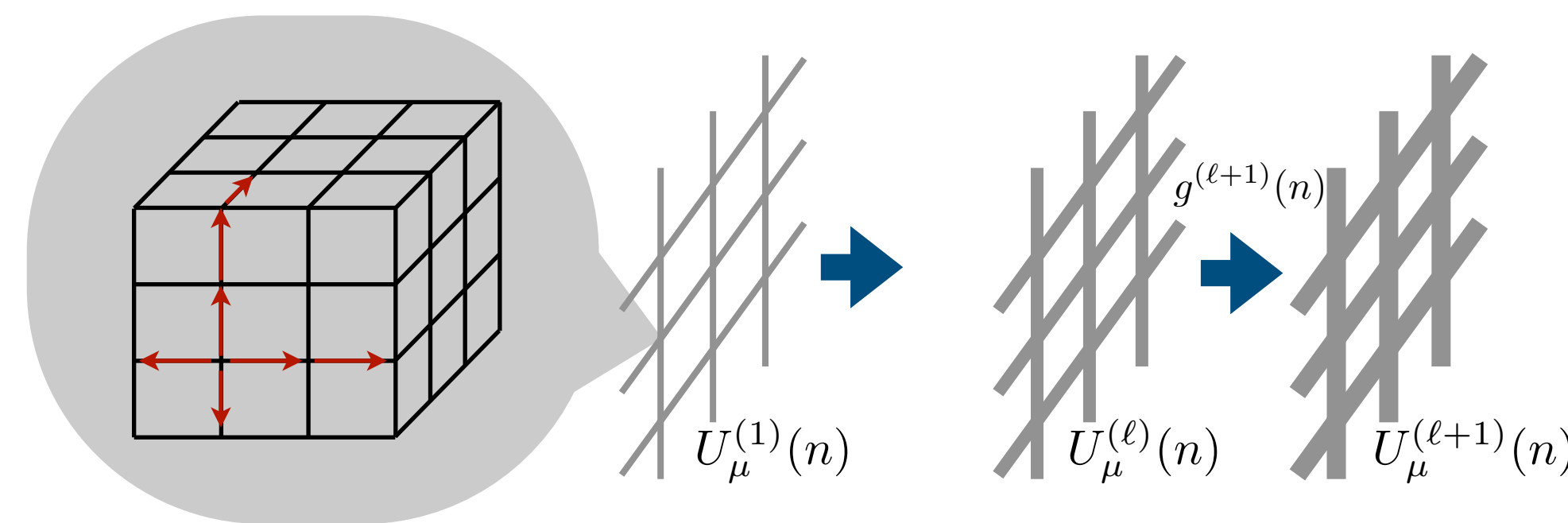
Volume transfer: training with small, applying to large

Apply to a larger lattice, 256^4 , as well as some ensemble that suffers from severe critical slowing down

Examination of the gauge-fixing quality by measuring the spectrum quantities, i.e., the pion mass

Hybrid method with advanced techniques?

Training for other gauges

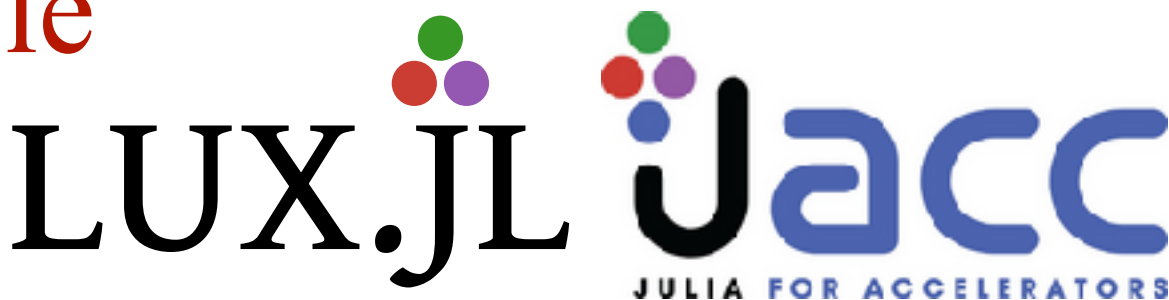


Software



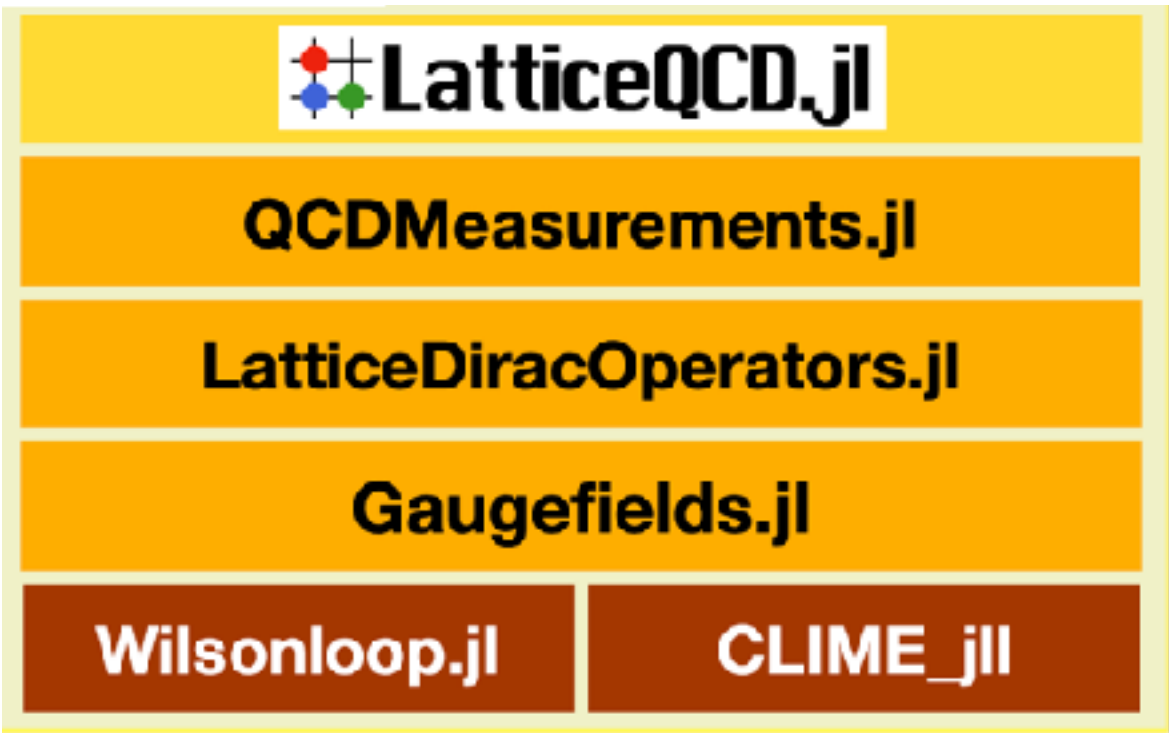
Fast as C, Easy as Python, ML-friendly, Portable

arXiv:1209.5145, <https://julialang.org/>

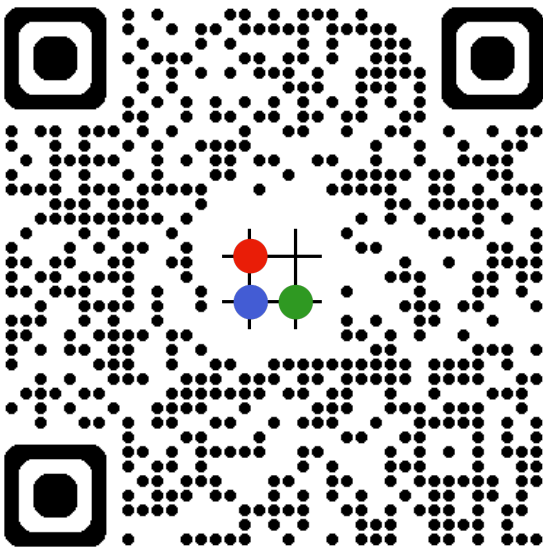
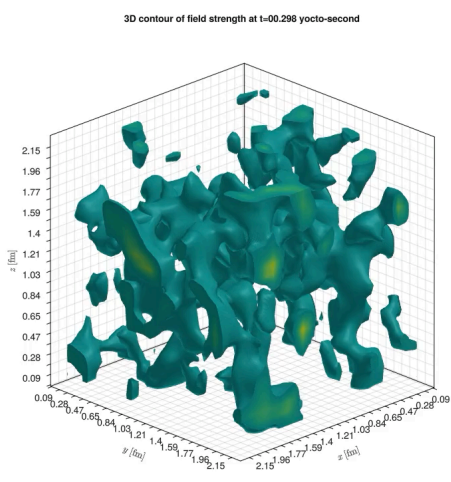
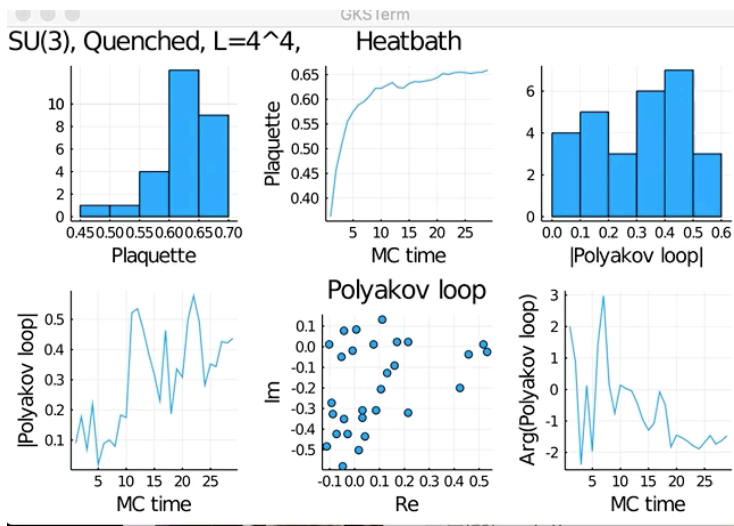


Run almost everywhere: Laptop/Colab/Jupyter/**Supercomputers**

Advantage: **Portability**, no-explicit compile*, fast, Quick trial-and-error feedback loop
Functionality: SU(Nc)-heatbath, (R)HMC, **Self-learning HMC**, **HMC with AD**,
Dynamical fermions, Measurements (chiral condensate, topological charge, etc)



Yuki Nagai, Akio Tomiya, arXiv:2409.03030, <https://github.com/JuliaQCD>



END

THANK YOU

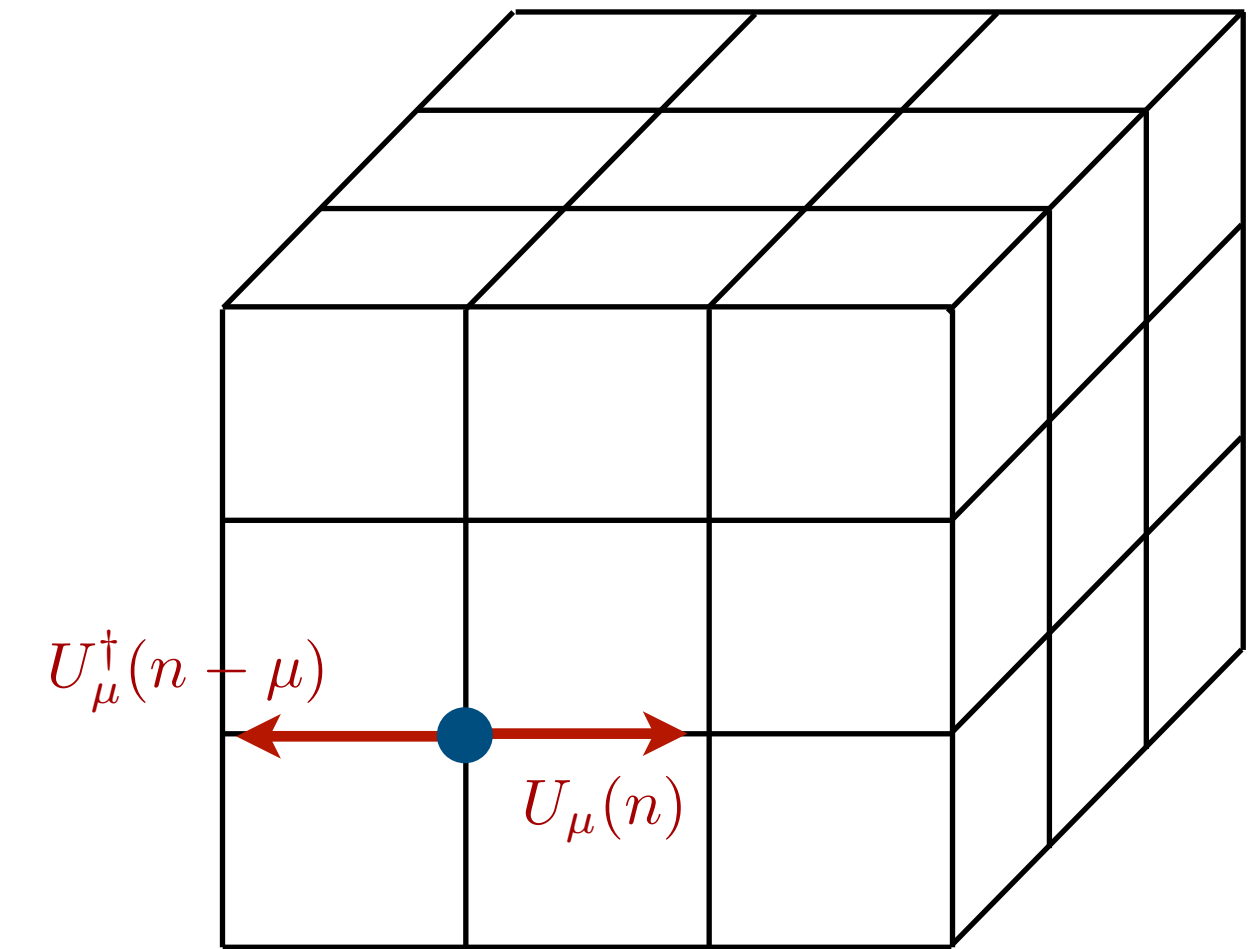
Lattice Gauge Fixing

Iterative approaches

- Los Alamos (LA) method: SU(2) subgroup update

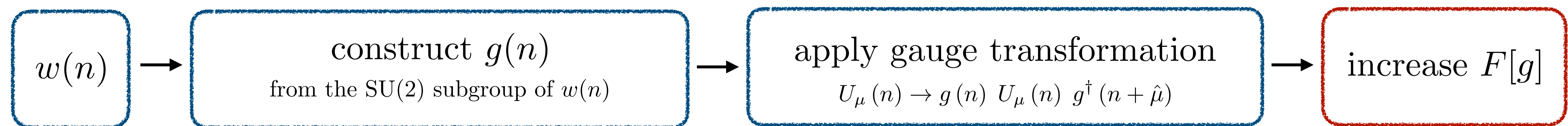
[R.Gupta, G.Guralnik, G.Kilcup, A.Patel, S. R. Sharpe, and T.Warnock, Phys.Rev. D36 (1987) 2813.]

$$w(n) = \sum_{\mu} [U_{\mu}(n) + U_{\mu}^{\dagger}(n - \mu)]$$



with the gauge transformation based on the Cabbibo-Marinari method (maximal trace)

$$w(n) \rightarrow g(n)w(n) \text{ such that } \mathbb{R}\text{Tr}g(n)w(n) \geq \mathbb{R}\text{Tr}w(n)$$



Lattice Gauge Fixing

Iterative approaches

- Cornell method: Steepest descent (SD)

[C.T.H.Davies, et al., Phys.Rev D37 (1988) 1581]

Differentiating $F[g]$
with respect to $w_j(n)$ where $g(n) = \exp(iw_a t^a)$

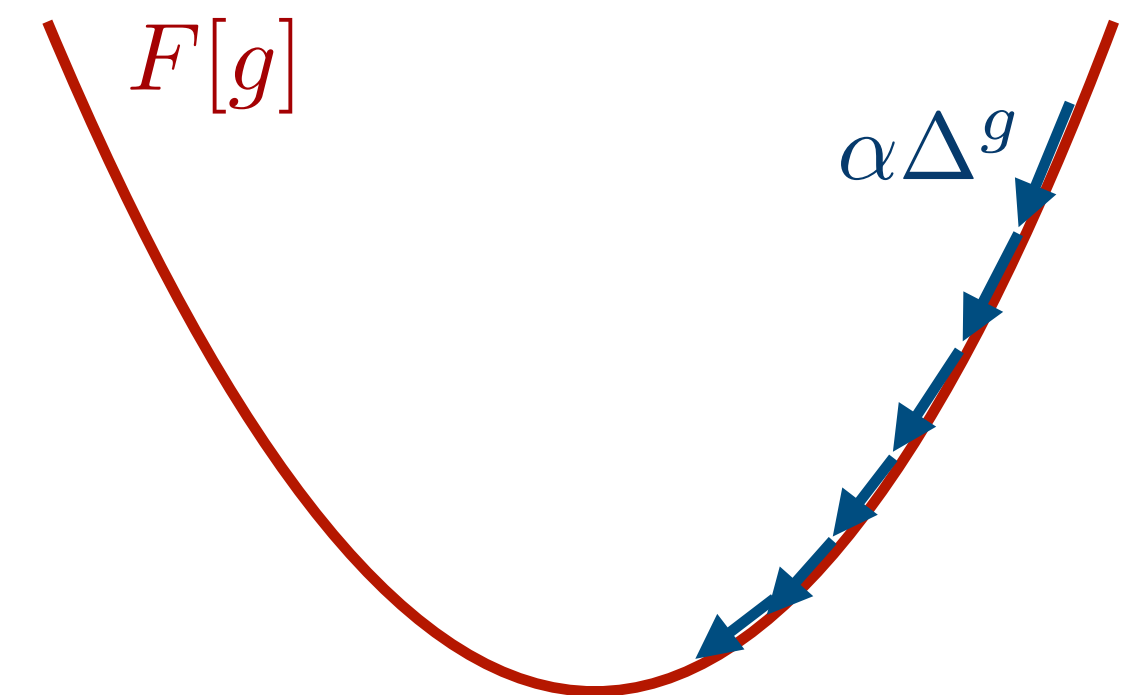
$$g(n) = \exp\left(\frac{1}{2}\alpha\Delta(n)\right), \text{ where } \Delta(n) = \sum_{\mu} \underbrace{[U_{\mu}^{\dagger}(n - \mu) - U_{\mu}(n)]_{\text{TA}}}_{\propto \partial_{\mu} A^{\mu}(n), \text{ where } U_{\mu}(n) = e^{iagA_{\mu}(n)}}$$

$\downarrow$
 tunable parameter

- Mino method: determining the optimal tuning parameters

$$\alpha(n) = \frac{\mathbb{R} \sum_{\mu} \text{Tr} [\Delta(n) (U_{\mu}(n - \hat{\mu}) - U_{\mu}(n))]}{\mathbb{R} \sum_{\mu} \text{Tr} [\Delta^2(n) (U_{\mu}(n - \hat{\mu}) + U_{\mu}(n))]}$$

by solving $\frac{\partial F[g]}{\partial \alpha(n)} = 0$



Our model: Machine learning approach

Network construction

- The gauge transform matrix is defined as

$$\begin{aligned}
 L_s \left(n; \left\{ U^{(\ell)} \right\} \right) &= \left[\prod_{b=1}^s \left\{ \sum_{\substack{\rho_b = -d_{\text{fix}} \\ \rho_b \neq -\rho_{b-1} \neq 0}}^{d_{\text{fix}}} U_{\rho_b}^{(\ell)} \left(n + \sum_{a=1}^{b-1} \hat{\rho}_a \right) \right\} \right]^{\dagger} \quad \text{where} \quad s \geq 1 \\
 &= \left[\sum_{\substack{\rho_1 = -d_{\text{fix}} \\ \rho_1 \neq 0}}^{d_{\text{fix}}} \sum_{\substack{\rho_2 = -d_{\text{fix}} \\ \rho_2 \neq -\rho_1 \neq 0}}^{d_{\text{fix}}} \cdots \sum_{\substack{\rho_s = -d_{\text{fix}} \\ \rho_s \neq -\rho_{s-1} \neq 0}}^{d_{\text{fix}}} U_{\rho_1}^{(\ell)}(n) U_{\rho_2}^{(\ell)}(n + \hat{\rho}_1) \cdots U_{\rho_s}^{(\ell)} \left(n + \sum_{a=1}^{s-1} \hat{\rho}_a \right) \right]^{\dagger} \\
 \xrightarrow{r=1} \quad L_1 \left(n; \left\{ U^{(\ell)} \right\} \right) &= \left[\sum_{\mu = -d_{\text{fix}}}^{d_{\text{fix}}} U_{\mu}^{(\ell)}(n) \right]^{\dagger}
 \end{aligned}$$

Our model: Machine learning approach

backpropagation

- $$\Lambda_{\mu}^{H(\ell+1)}(m) = \left[\left\{ U_{\mu}^{(\ell)}(m) g^{(\ell+1)\dagger}(m + \hat{\mu}) \delta_{\mu}^{(\ell+1)}(m) \right\} \star \frac{\partial g^{(\ell+1)}(m)}{\partial Q^{(\ell+1)}(m)} \right]_{\text{TA}}$$

$$\Lambda_{\mu}^{S(\ell+1)}(m + \hat{\mu}) = \left[\left\{ U_{\mu}^{(\ell)\dagger}(m) g^{(\ell+1)\dagger}(m) \delta_{\mu}^{(\ell+1)\dagger}(m) \right\} \star \frac{\partial g^{(\ell+1)}(m + \hat{\mu})}{\partial Q^{(\ell+1)}(m + \hat{\mu})} \right]_{\text{TA}}$$

$$\delta_{\mu}^{(\ell)}(n) = g^{(\ell+1)\dagger}(m + \hat{\mu}) \delta_{\mu}^{(\ell+1)}(n) g^{(\ell+1)}(m)$$
- where N_c is the number of color, and N_{vol} is the number of the lattice sites

Training gradient $\Rightarrow$ update training parameters with an optimiser

$$\frac{dE(\theta)}{d\theta_r^{(\ell)}} = 2 \sum_{\mu=1}^{d_{\text{fix}}} \sum_n \text{Re Tr} \left[\Lambda_{\mu}^{H(\ell)}(n) L_r(n; \{U_{\mu}^{(\ell)}(n)\}) + \Lambda_{\mu}^{S(\ell)}(n + \hat{\mu}) L_r(n + \hat{\mu}; \{U_{\mu}^{(\ell)}(n)\}) \right]$$

$$+ \sum_m \sum_{\nu=1}^{d_{\text{fix}}} \left[U_{\nu}^{(\ell)}(m) g^{(\ell+1)\dagger}(m + \hat{\nu}) \delta_{\nu}^{(\ell+1)}(m) \right] \star \left[\frac{\partial g^{(\ell+1)}(m)}{\partial U_{\mu}^{(\ell)}(n)} \right]$$

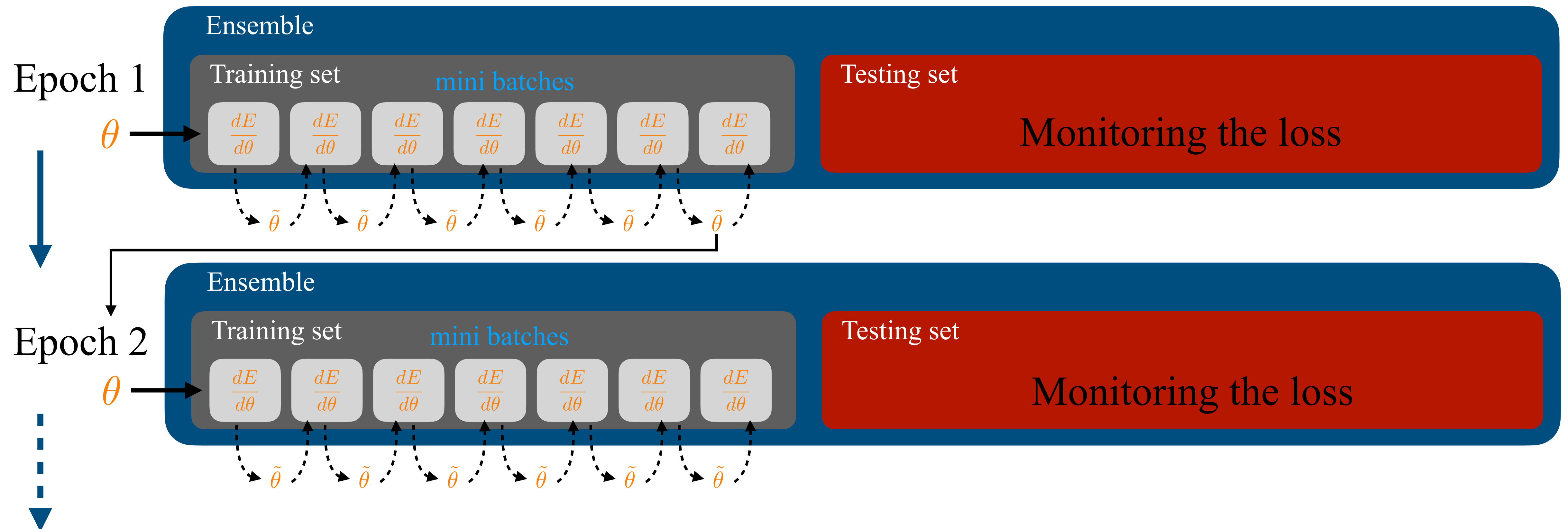
$$+ \sum_m \sum_{\nu=1}^{d_{\text{fix}}} \left[\delta_{\nu}^{(\ell+1)\dagger}(m) g^{(\ell+1)}(m + \hat{\nu}) U_{\nu}^{(\ell)\dagger}(m) \right] \star \left[\frac{\partial g^{(\ell+1)\dagger}(m)}{\partial U_{\mu}^{(\ell)}(n)} \right]$$

$$+ \sum_m \sum_{\nu=1}^{d_{\text{fix}}} \left[U_{\nu}^{(\ell)\dagger}(m) g^{(\ell+1)\dagger}(m) \delta_{\nu}^{(\ell+1)\dagger}(m) \right] \star \left[\frac{\partial g^{(\ell+1)}(m + \hat{\nu})}{\partial U_{\mu}^{(\ell)}(n)} \right]$$

$$+ \sum_m \sum_{\nu=1}^{d_{\text{fix}}} \left[\delta_{\nu}^{(\ell+1)}(m) g^{(\ell+1)}(m) U_{\nu}^{(\ell)}(m) \right] \star \left[\frac{\partial g^{(\ell+1)\dagger}(m + \hat{\nu})}{\partial U_{\mu}^{(\ell)}(n)} \right]$$

Gauge fixing procedure

Mini-batch training

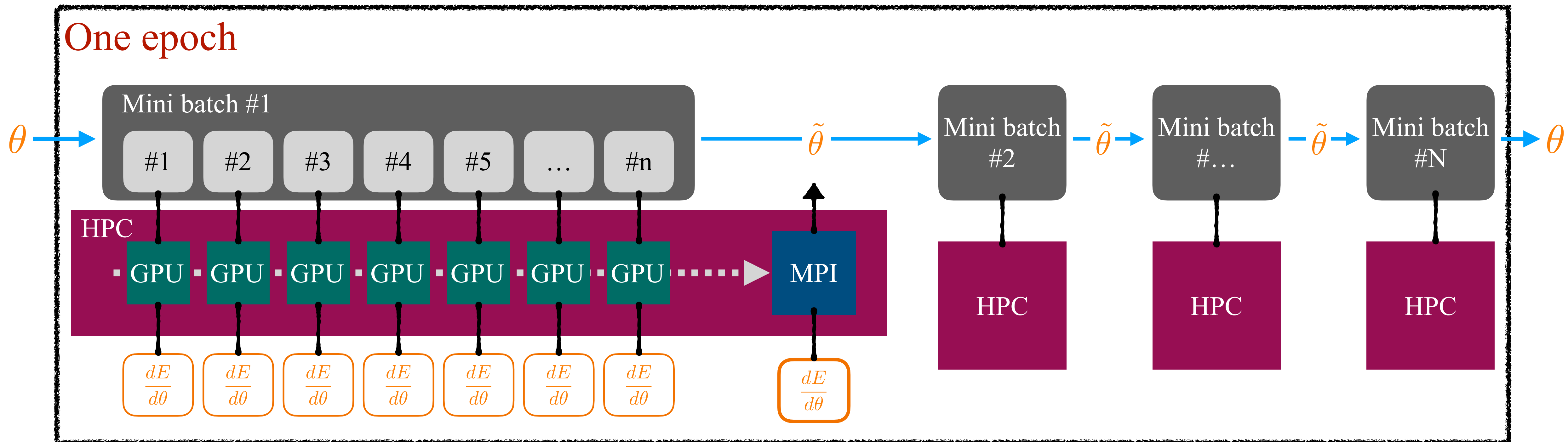


♣ Adam (Adaptive Moment Estimation) optimiser is applied to update the training parameters.

Gauge fixing procedure

Computation on HPC

- MPI for lattice (compute the gradient for a single configuration one by one)
- GPU for lattice + MPI for parallelising batch training



Preliminary results

RC32x48: training for Coulomb gauge

- Schemes:

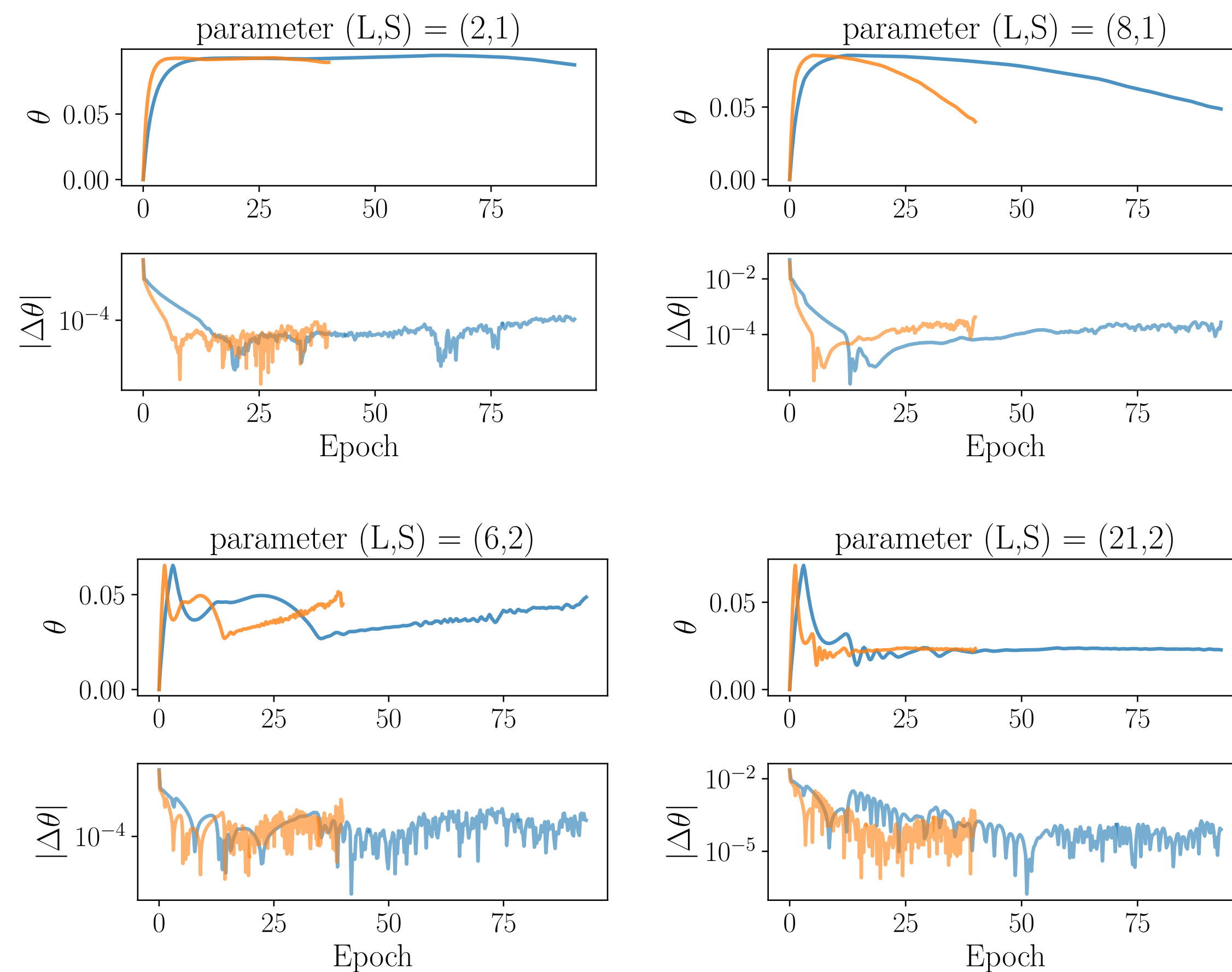
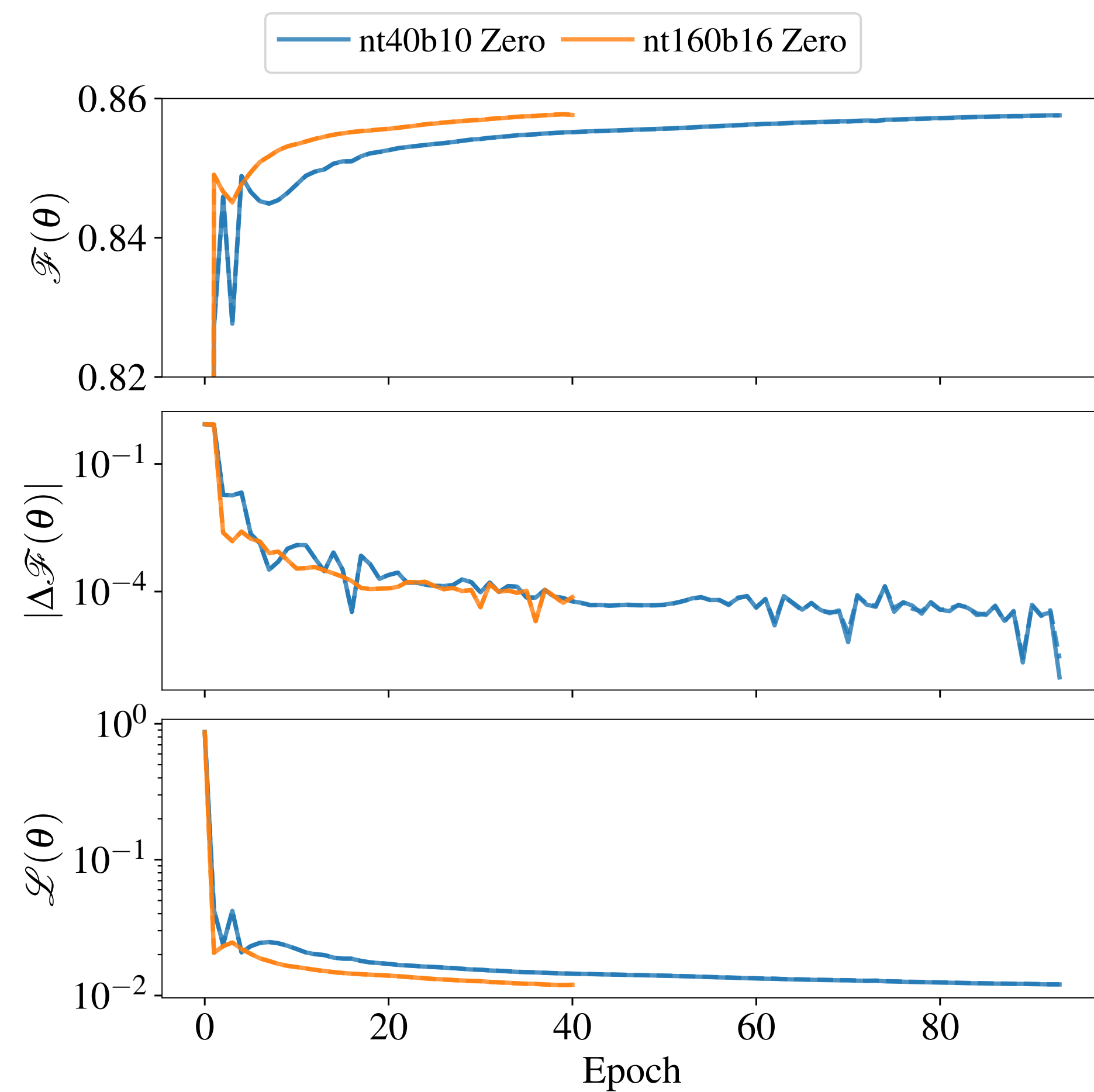
Training tag	configs	Batch size	Layers	S-length	Initial param.	Epoch	$\mathcal{F}$
L21S2 nt40b10 Zero	40	10	21	2	zeros	93	0.8576
L21S2 nt160b16 Zero	160	16	21	2	zeros	40	0.8576
L12S3 nt40b10 Zero	40	10	12	3	zeros	32	0.8498
L12S3 nt160b16 Param	160	16	12	3	[nt40b10 Zero]	14+34	0.8538
L12S3 nt160b16 Zero	160	16	12	3	zeros	39	0.8538

*From pure interative method: $F_{\text{fixed}} = 0.86965(8)$

- Investigation into the size of the training set.
- Investigation into the incremental training: starting with a smaller training set, then switching to a larger set.

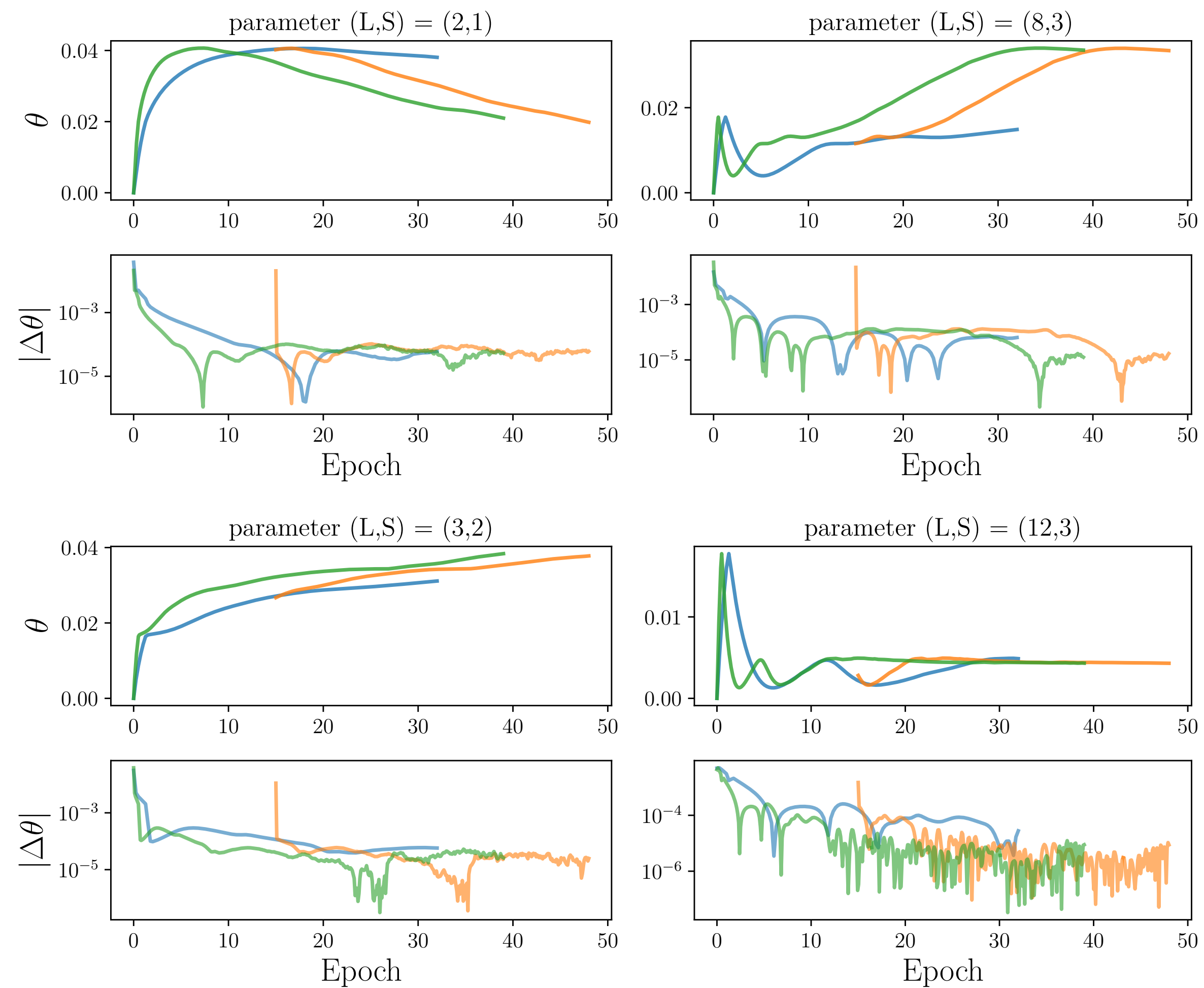
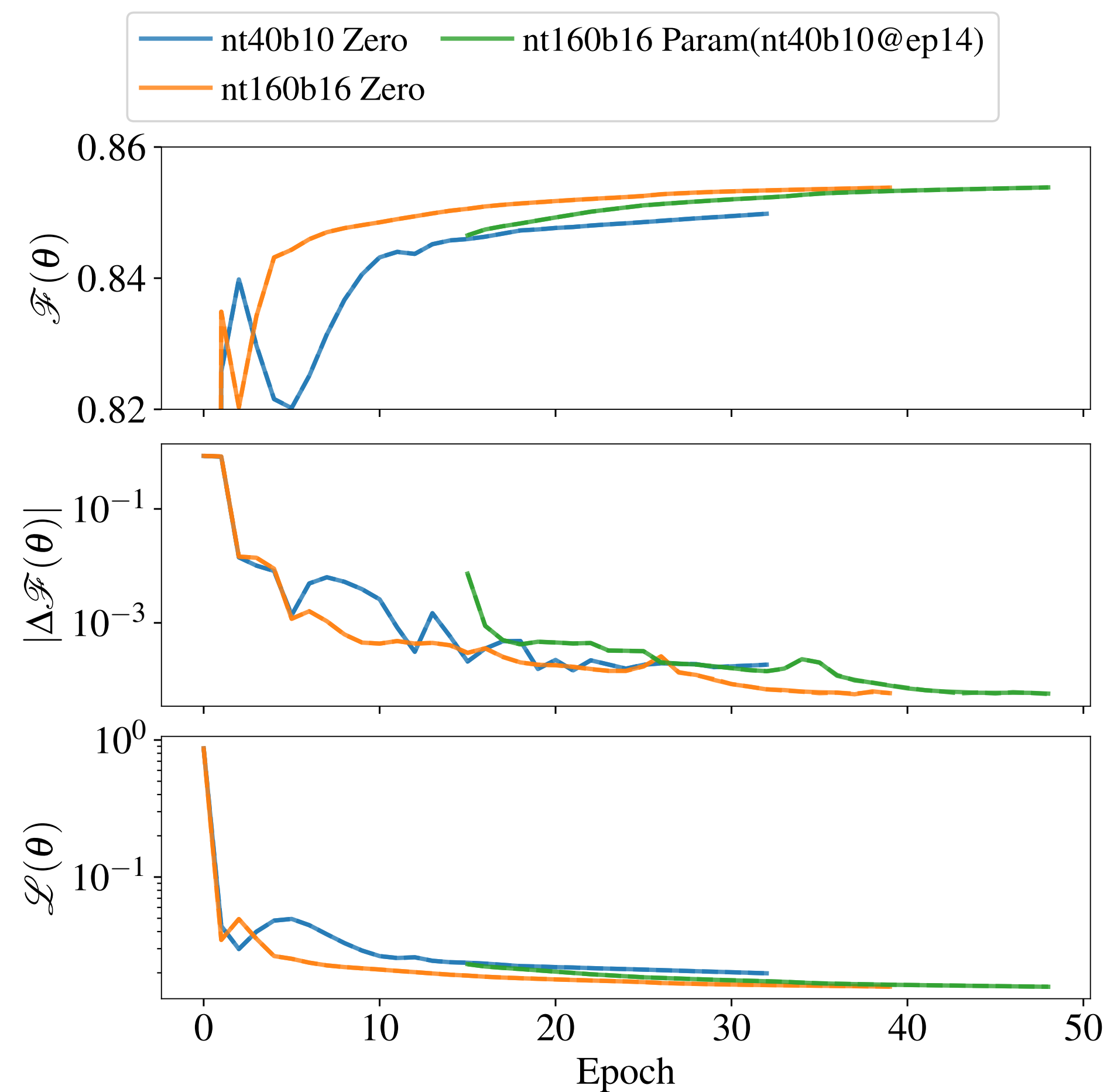
Preliminary results

RC32x48: Different size of training set - L21S2



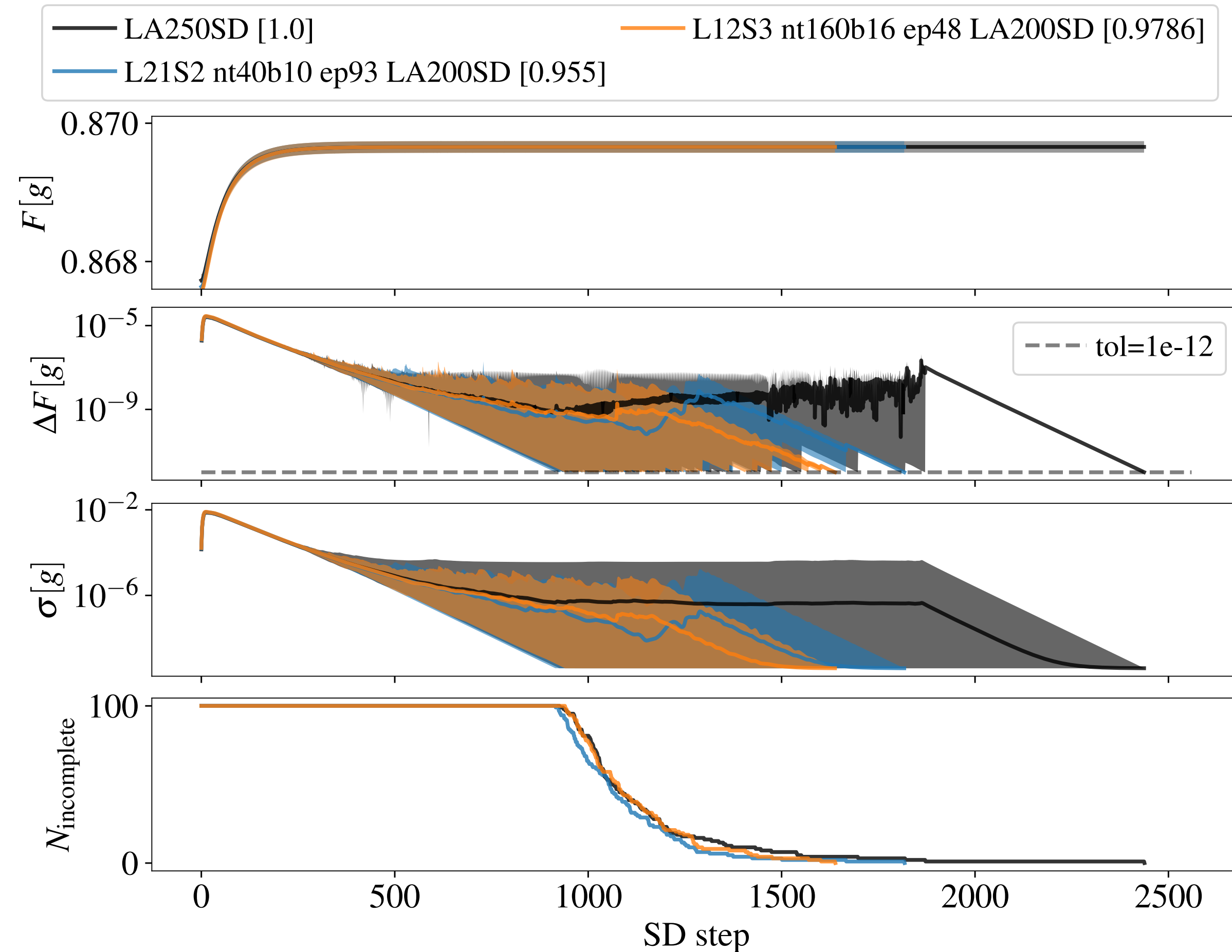
Preliminary results

RC32x48: Increamental training - L12S3



Preliminary results

RC32x48: gauge fixing comparison

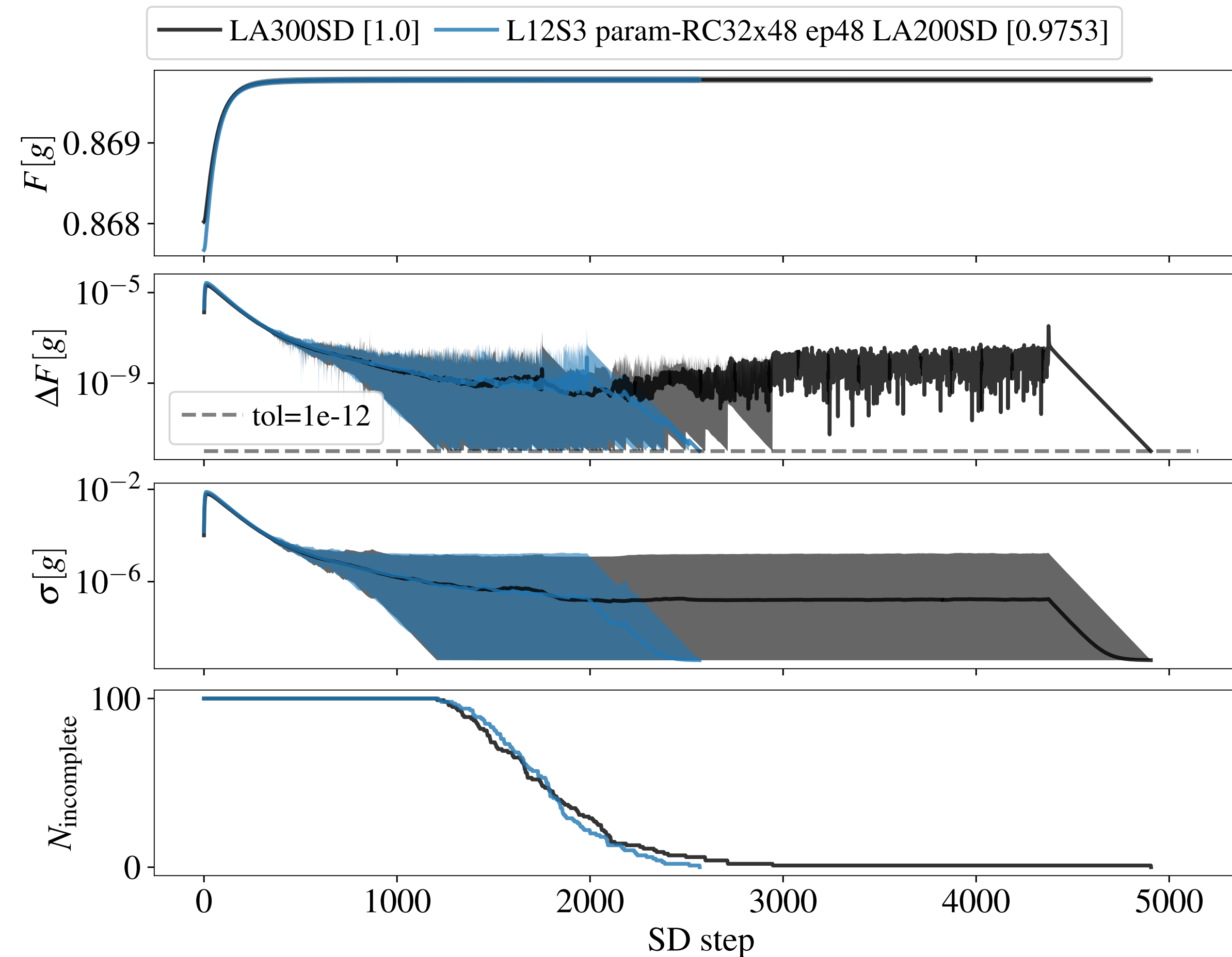


Comparison of the SD section in the gauge fixing procedure with different training schemes, using

- Pure iterative method (250 steps of LA then SD) as a baseline.
- Apply the WLB gauge transformation with parameters trained with 21 layers and up to length-2 Wilson lines at epoch 93 using 40 configurations.
- Apply the WLB gauge transformation with parameters trained with 12 layers and up to length-3 Wilson lines at epoch 48 using an incremental training scheme.

Preliminary results

RC48x48: using parameters from RC32x48

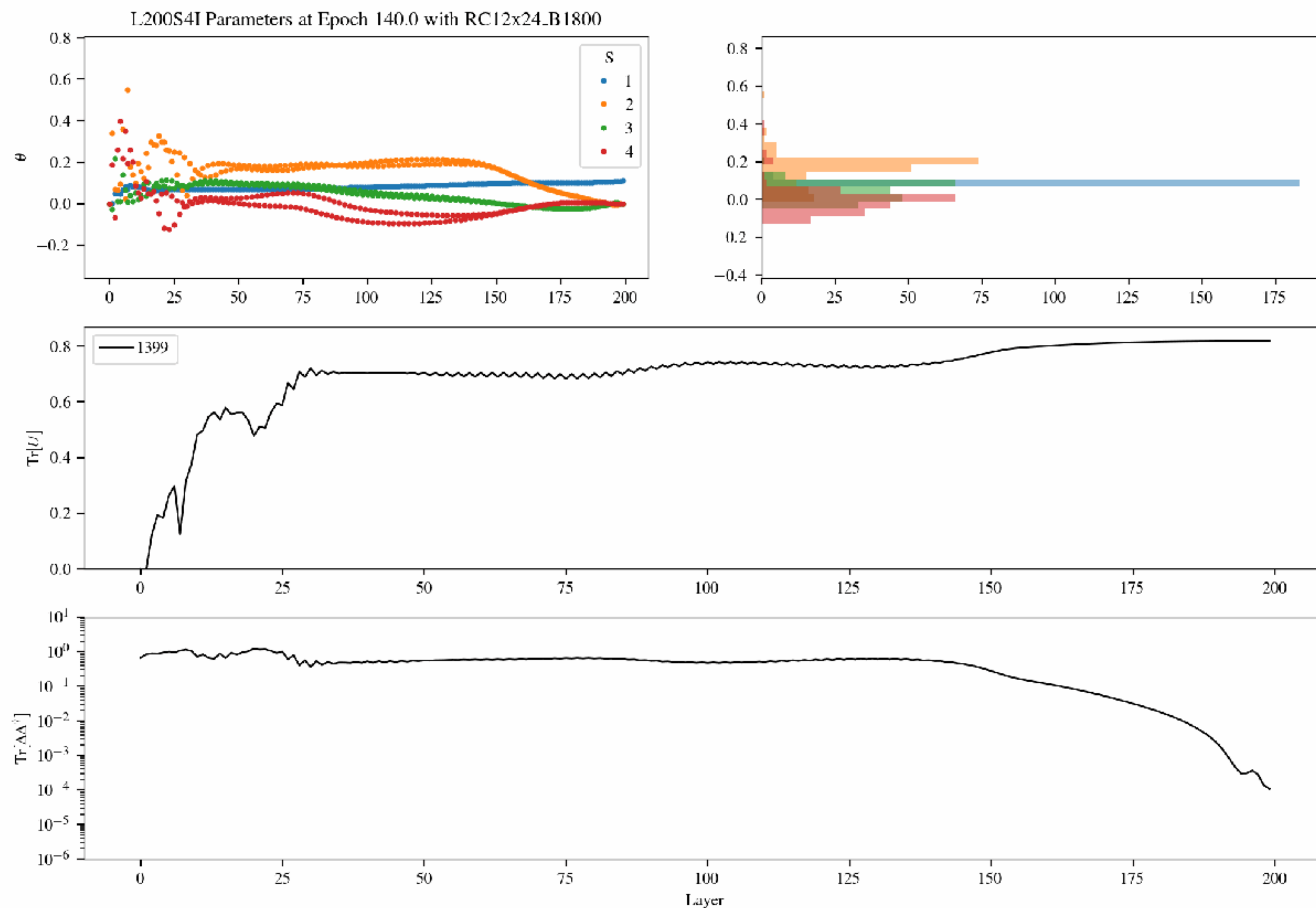


An exemplified application of using the training parameters from the ensemble RC32x48, which shares the same lattice parameters as those of RC48x48, except for the spatial extent.

- Training with a smaller lattice for a larger one might be possible.

Preliminary results

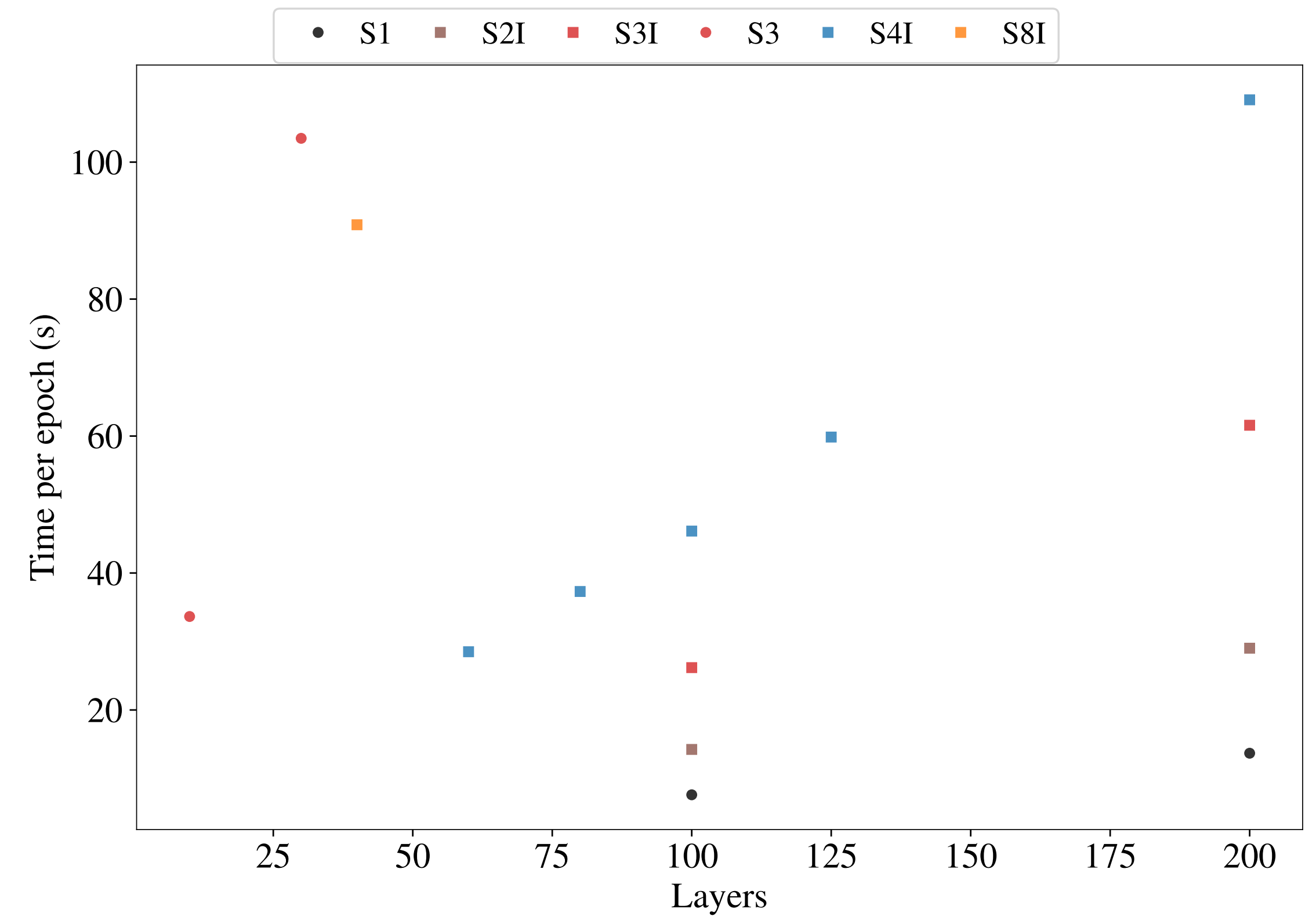
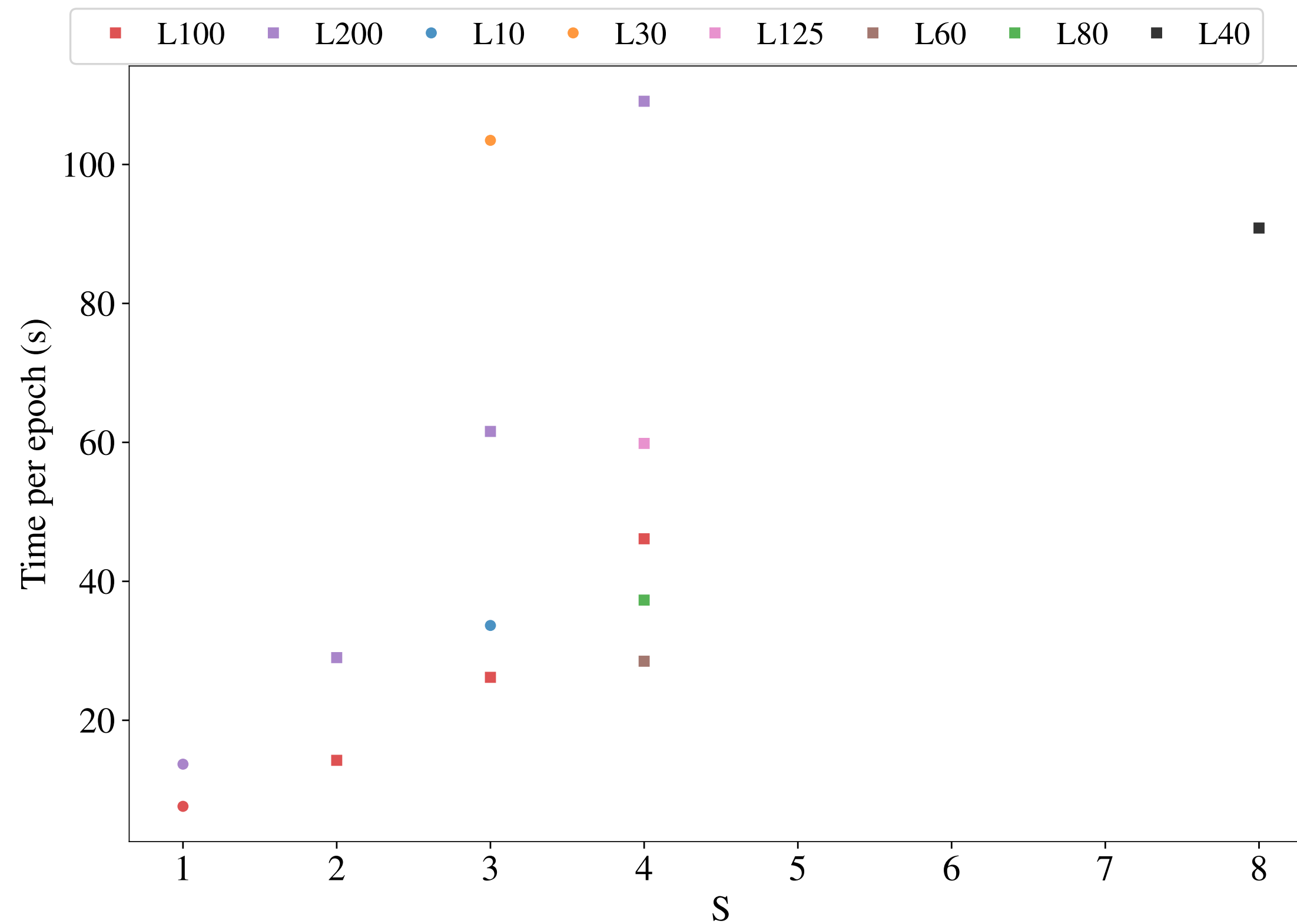
Details of CNN



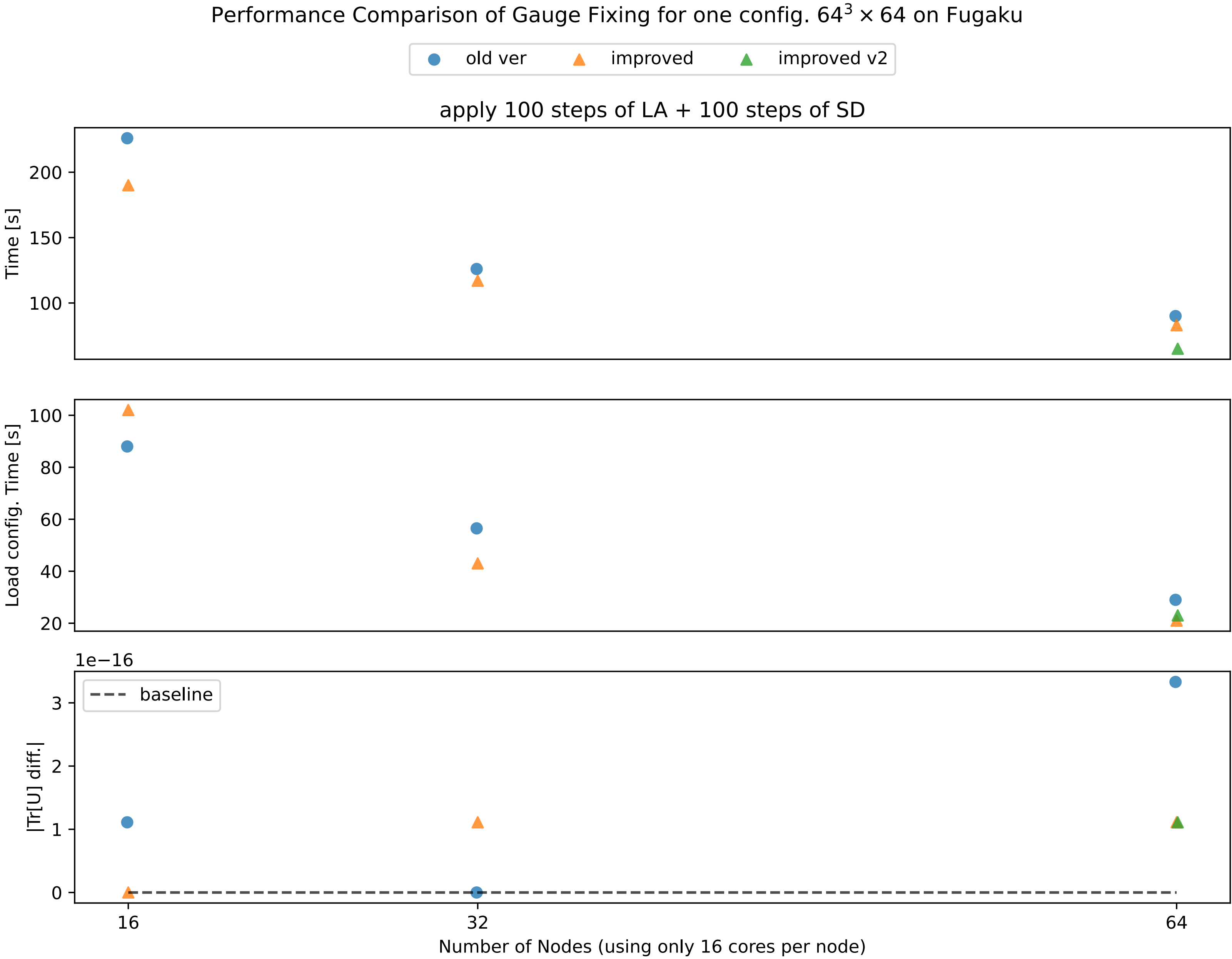
Preliminary results

Model scanning toward deep network

- Comparing training cost



Performance Improvement On Fugaku



Performance Improvement On Miyabi

Performance Comparison of Gauge Fixing for one config. $48^3 \times 48$ on Miyabi

