Neural Network Gauge Field Transformation for 4D SU(3) gauge fields

Xiao-Yong Jin (Argonne National Laboratory)

August 3, 2023
Lattice 2023, Fermilab

EXASCALE
COMPUTING
PROJECT

Many thanks to Peter Boyle, Norman Christ, Sam Foreman, Taku Izubuchi, Luchang Jin, Chulwoo Jung, Nobuyuki Matsumoto, James Osborn, Akio Tomiya, and other ECP collaborators for insightful discussions and support.

Outline

- Change of variable, parametrized
- What to optimize
- Previous 2D U(1) results
- Preliminary 4D SU(3) results
- Outlook

Change of variables

- Use a continuously differentiable bijective map $\mathcal{F}^{-1}$ from target field U to the mapped field $V = \mathcal{F}^{-1}(U)$, same group manifold for us

$$\langle \mathcal{O} \rangle = \frac{1}{Z} \int \mathcal{D}U \mathcal{O}(U) e^{-S(U)} = \frac{1}{Z} \int \mathcal{D}V \mathcal{O}\big(\mathcal{F}(V)\big) e^{-S(\mathcal{F}(V)) + \ln|\mathcal{F}_*|} \text{ where } \mathcal{F}_* = \frac{\partial \mathcal{F}(V)}{\partial V}$$

- Sample V with HMC according to the new action: Field Transformation HMC (FTHMC)

$$S_{\mathrm{FT}}(V) = S\big(\mathcal{F}(V)\big) - \ln|\mathcal{F}_*(V)|$$

Luscher 2010
Luchang Jin 2021
Sam Foreman 2021

- Want the effective action to have lower potential barriers, or more uniform dynamics (smaller difference between slow and fast modes)

- The Jacobian determinant and its derivative must remain simple

Parametrized bijection map: Generalized Stout Smearing

- Gauge covariant, dynamics remain the same with local gauge transformations, $\Omega_x \in \mathrm{SU}(3)$

$$U_{x,\mu} \longrightarrow U'_{x,\mu} = \Omega_x^\dagger U_{x,\mu} \Omega_{x+\hat{\mu}}$$

- Lie group element, exponential map from the group algebra (differentials in tangent directions) **Nagai&Tomiya 2021**

$$\mathcal{F} : V_{x,\mu} \to U_{x,\mu} = e^{\Pi_{x,\mu}} V_{x,\mu} \text{ where } \Pi_{x,\mu} = \sum_l \epsilon_l \partial_{x,\mu} W_l(V)$$

- Generalize it with neural networks **X.Jin 2021**
  - Make the coefficients arbitrary functions of gauge invariant quantities

$$\epsilon_{x,\mu,l} = c \tan^{-1}\left[\mathcal{N}_l(X, Y, \ldots)\right]$$

    - $X, Y, \ldots$ a list of traced Wilson loops local to x, μ, and independent of $U_{x,\mu}$
    - $\mathcal{N}$ is an arbitrary function, parameterized by neural networks
    - $c \tan^{-1}[\,\cdot\,]$ ensures a positive definite Jacobian

What to optimize

- Minimize the difference in the force between the transformed action and the original action on a stronger training coupling (smaller β_T)

$$\Delta_{x,\mu} = \left(\sum_c \left(\partial_{x,\mu,c} S_\mathrm{FT}(V;\beta) - \partial_{x,\mu,c} S(V;\beta_\mathrm{T}) \right)^2 \right)^{1/2}$$

- Choices of loss functions
  - Sums of root mean powers used in 2D U(1)

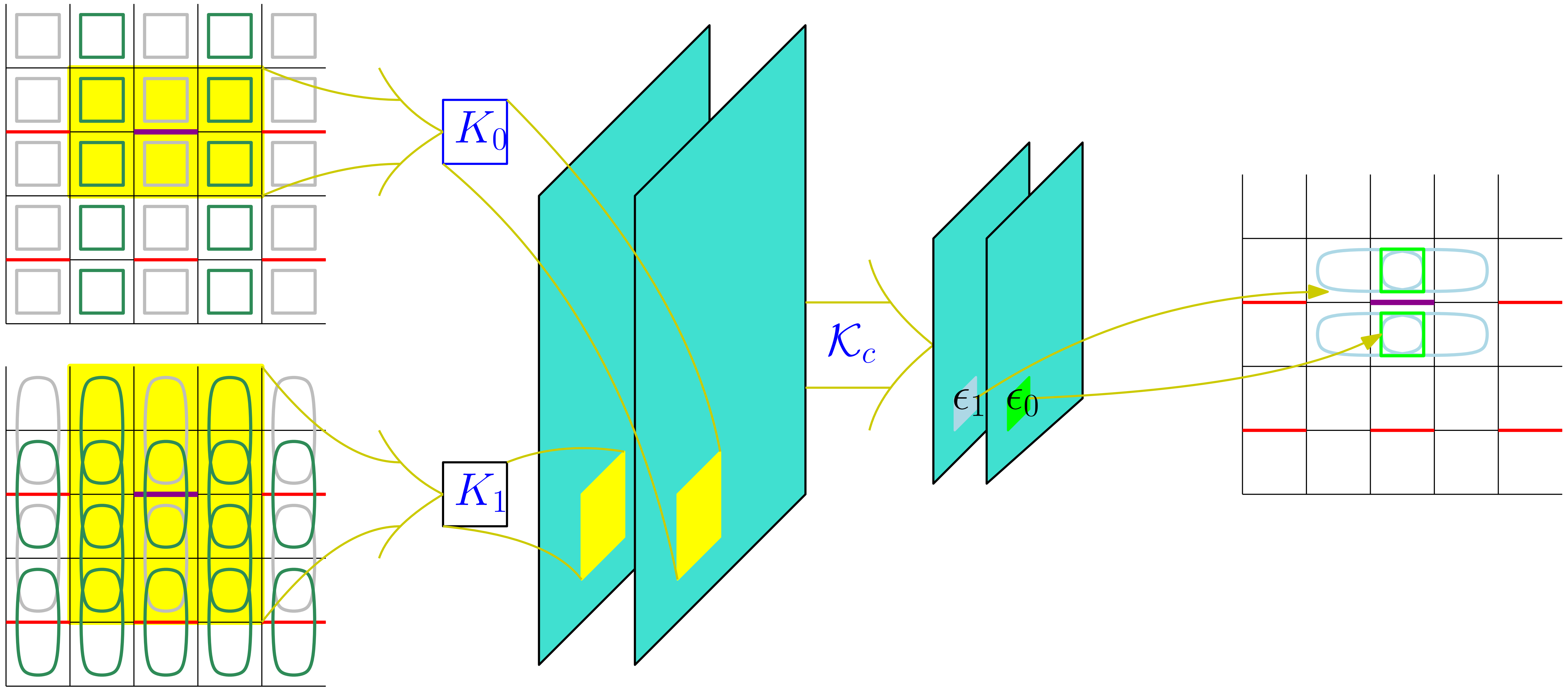
$$L_\mathrm{SRMP} = \sum_{p\in\{2,4,6,\infty\}} \left(\frac{C_p}{4\mathrm{Vol}} \sum_{x,\mu} \left(\Delta^2_{x,\mu} \right)^{p/2} \right)$$

  - Log mean exp norm used in 4D SU(3)

$$L_\mathrm{LMEN} = \ln \sum_{x,mu} \exp(\Delta_{x,\mu}) - \ln(4\mathrm{Vol})$$

Localized Coefficients, by Convolutional Neural Networks

2D lattice

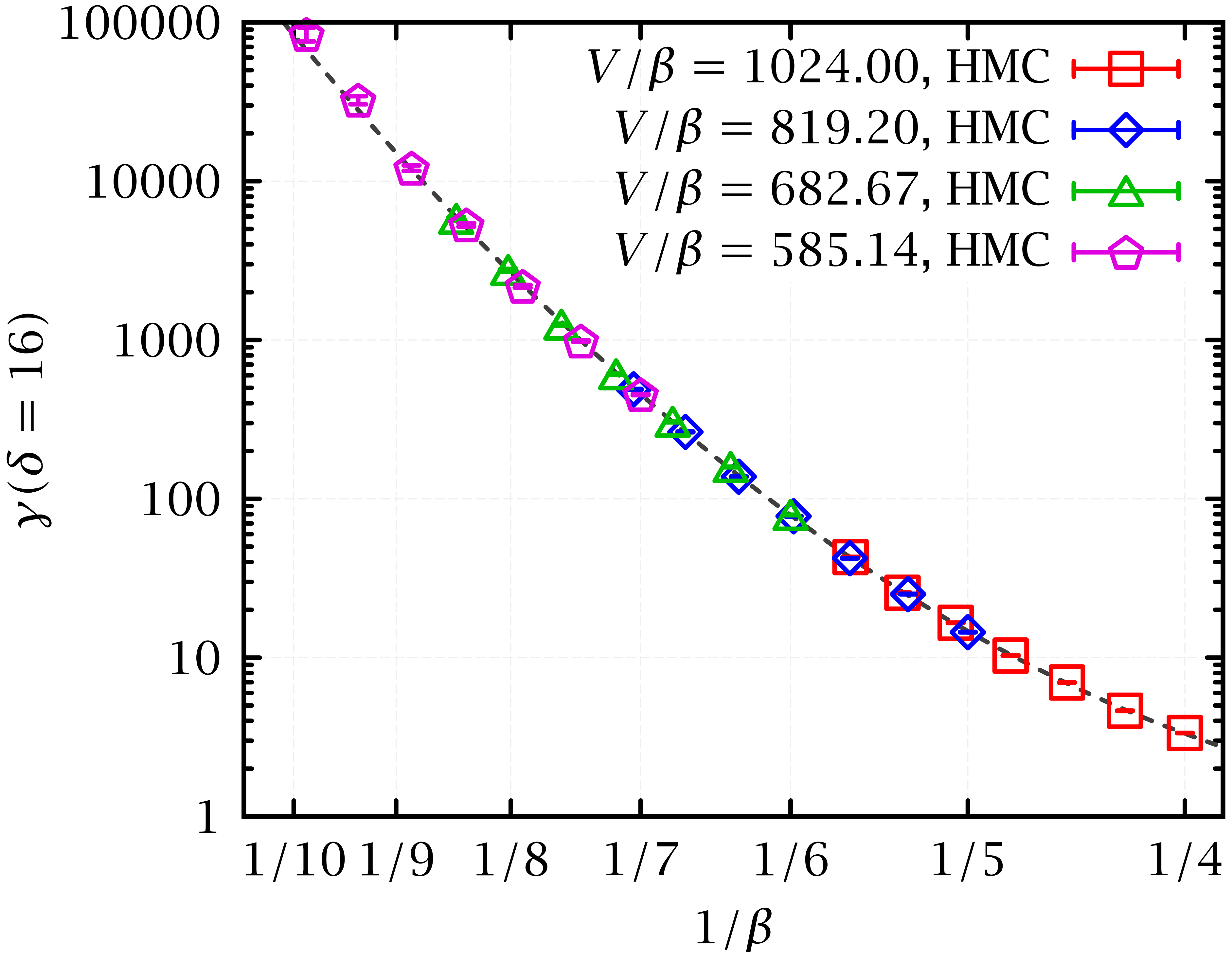


- Pick a subset of gauge links to update at a time (red links)
- Compute Wilson loops independent of the to-be-updated links (green loops)
- Pass through a series of convolutional neural networks and obtain coefficients

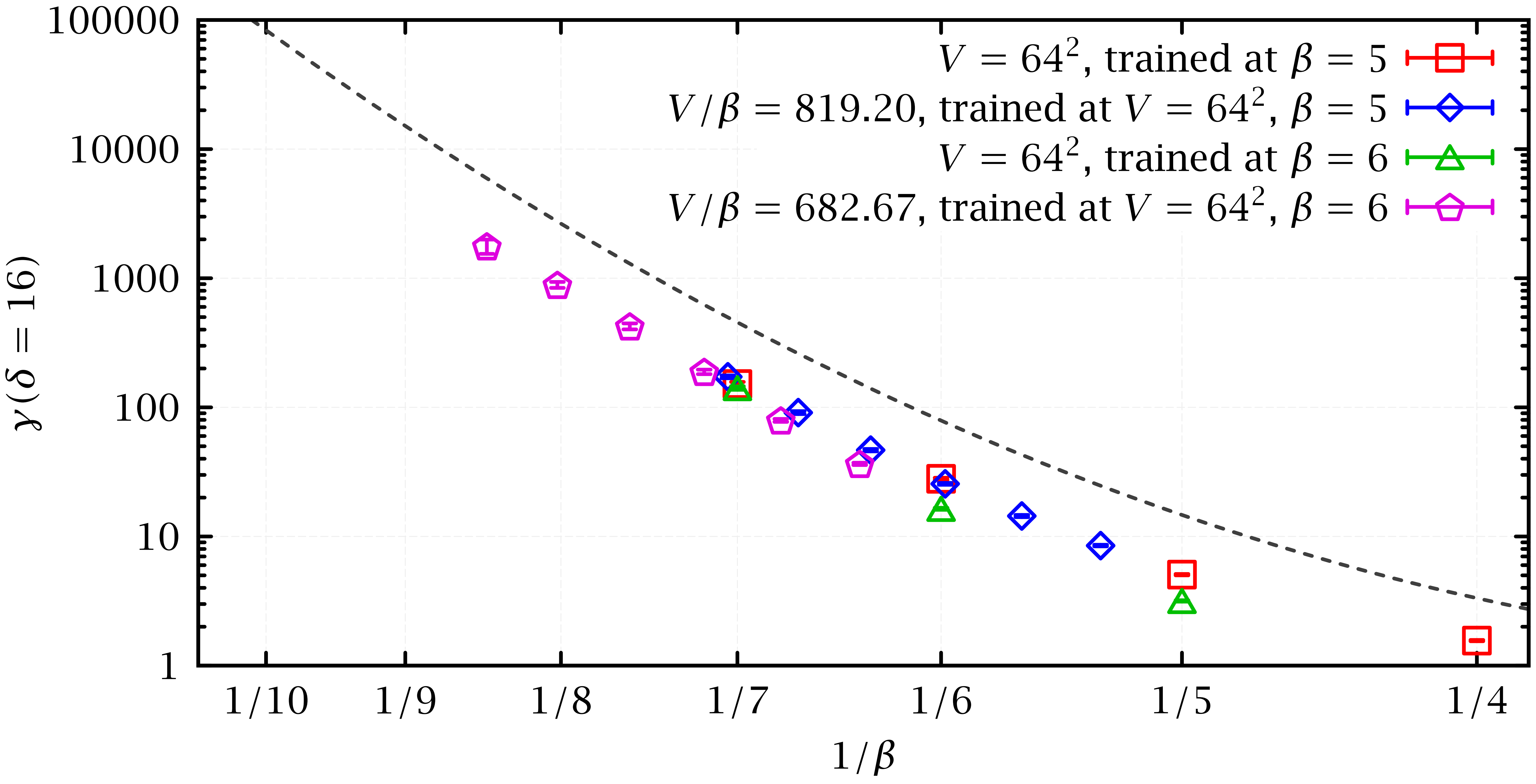
X.Jin 2021, arXiv:2201.01862

Results from 2D U(1) lattice fields, correlation of topological charge

Scaling of the integrated autocorrelation length



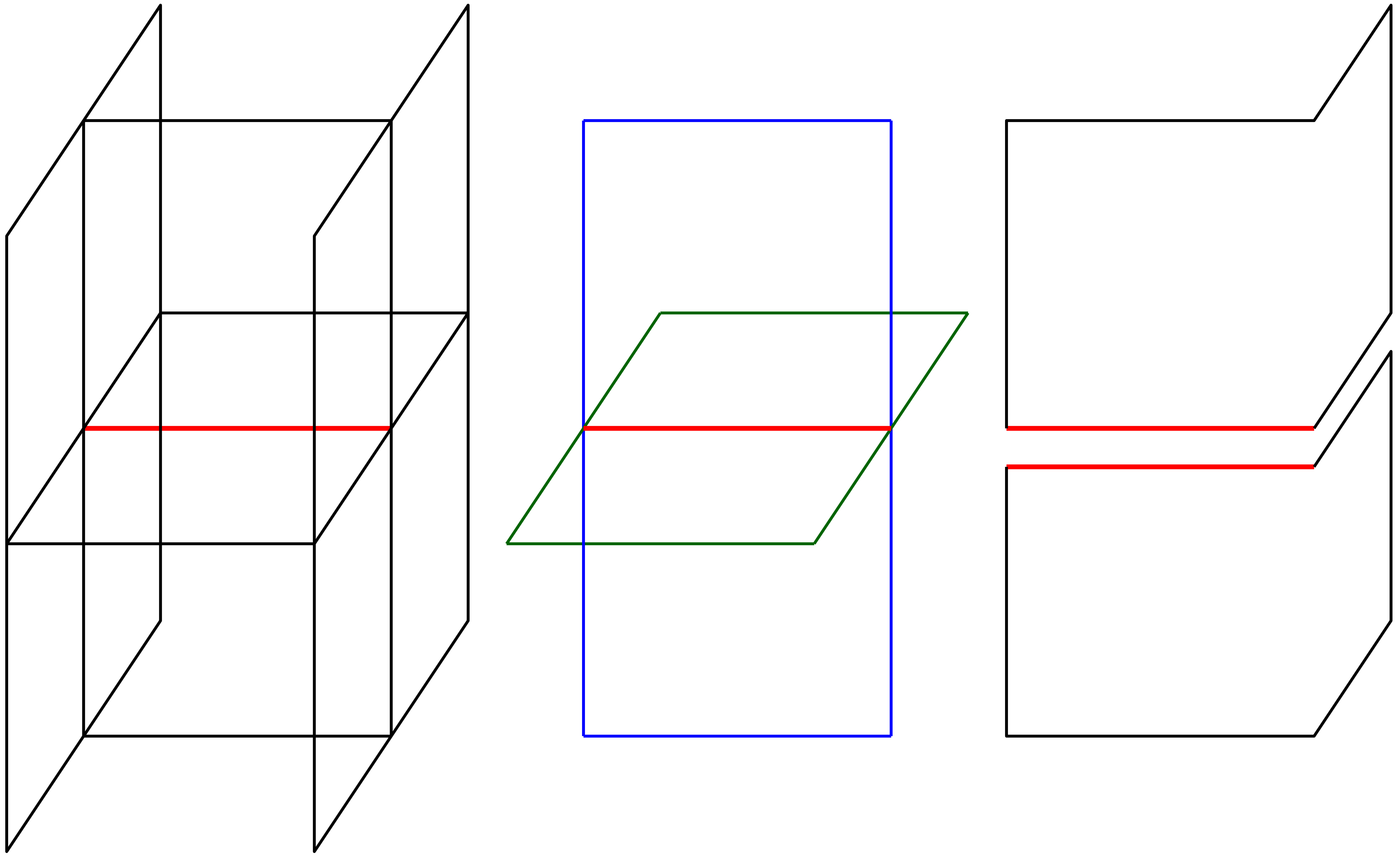
HMC



Fixed neural network architecture
Trained weights at four different conditions

Current work, onward to 4D SU(3) gauge fields

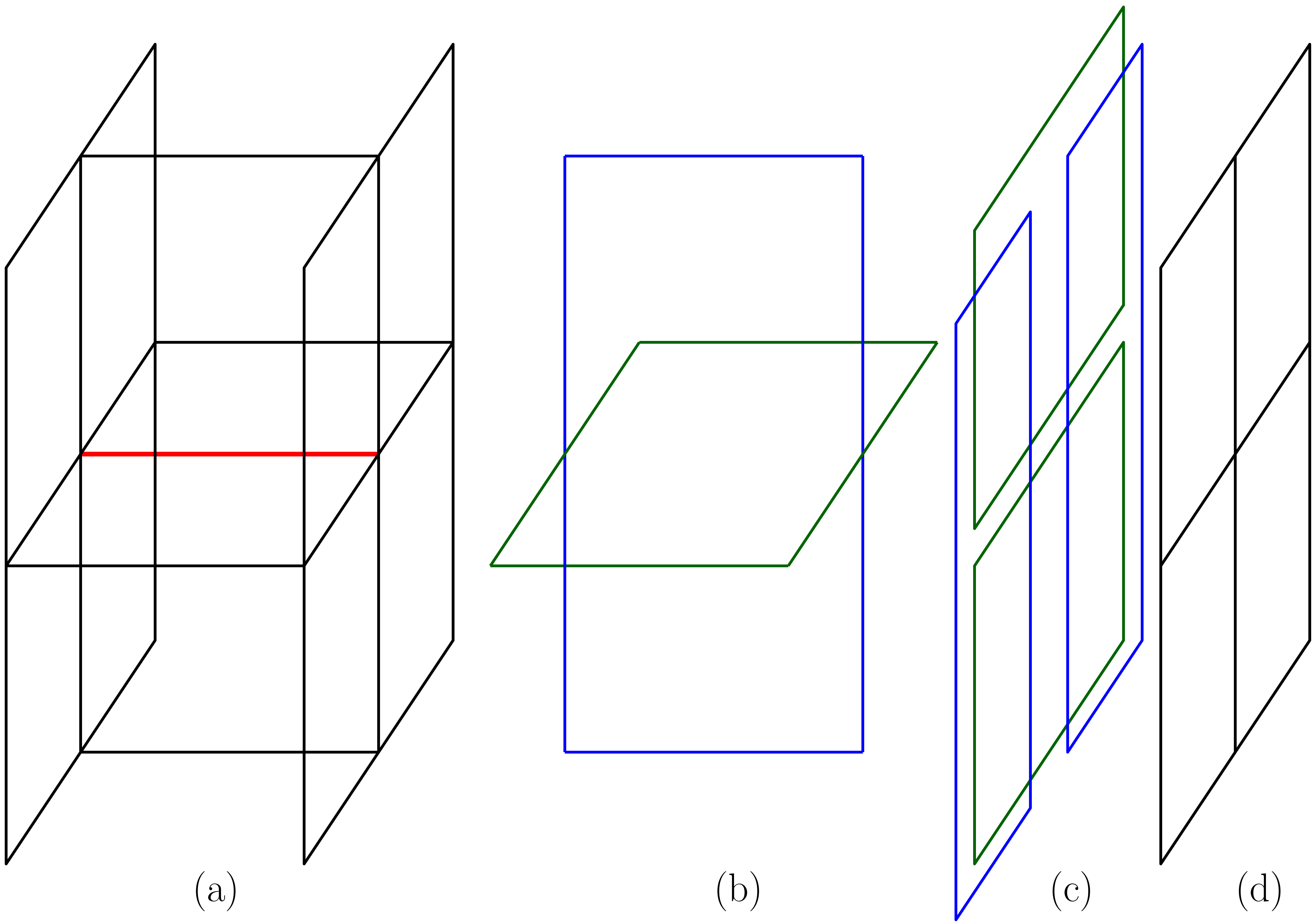
- The number of terms grows, and cost grows combinatorially
- Tractable Jacobian: updating 4 dir and even-odd separately
- Smearing with only links from other directions and parity
- For one link
  - 6 plaquette
  - 48 chair



(a) (b) (c)

Input to the coefficient neural network

- All the dependent links (a)
- Traced Wilson loops as input
  - Parallel rectangles (b)
  - Perpendicular rectangles (c)
  - Perpendicular plaquette (d)



(a) (b) (c) (d)

Training models and evaluations

scales from McGlynn&Mawhinney 2014

- Trained at DBW2 $\beta = 0.7796$, $8^3 \times 16$, $a \simeq 0.2$ fm
  - Tuned to optimize force matching $\beta_\text{T} = 0.7099$, $a \simeq 0.3$ fm
  - Adam optimizer, LR warm up and decay
  - Single A100 40GB on Polaris
- Evaluated at
  - DBW2 $\beta = 0.7796$, $8^3 \times 16$, $a \simeq 0.2$ fm
  - DBW2 $\beta = 0.8895$, $12^3 \times 24$, $a \simeq 0.13$ fm

Selected models, preliminary

- (GC) Global coefficients only

```
[StoutSmearSlice(coeff=CoefficientVariable(p0, chair=c0, rng=rng), dir=dir, is_odd=eo)
    for _ in range(3) for dir in range(4) for _ in range(3) for eo in {False,True}]
```

  - Trained with 64 configs, 8 epochs, 3 hours, 30 GB GPU Mem
  - Single force evaluation: 1.6 sec, 10 GB GPU Mem

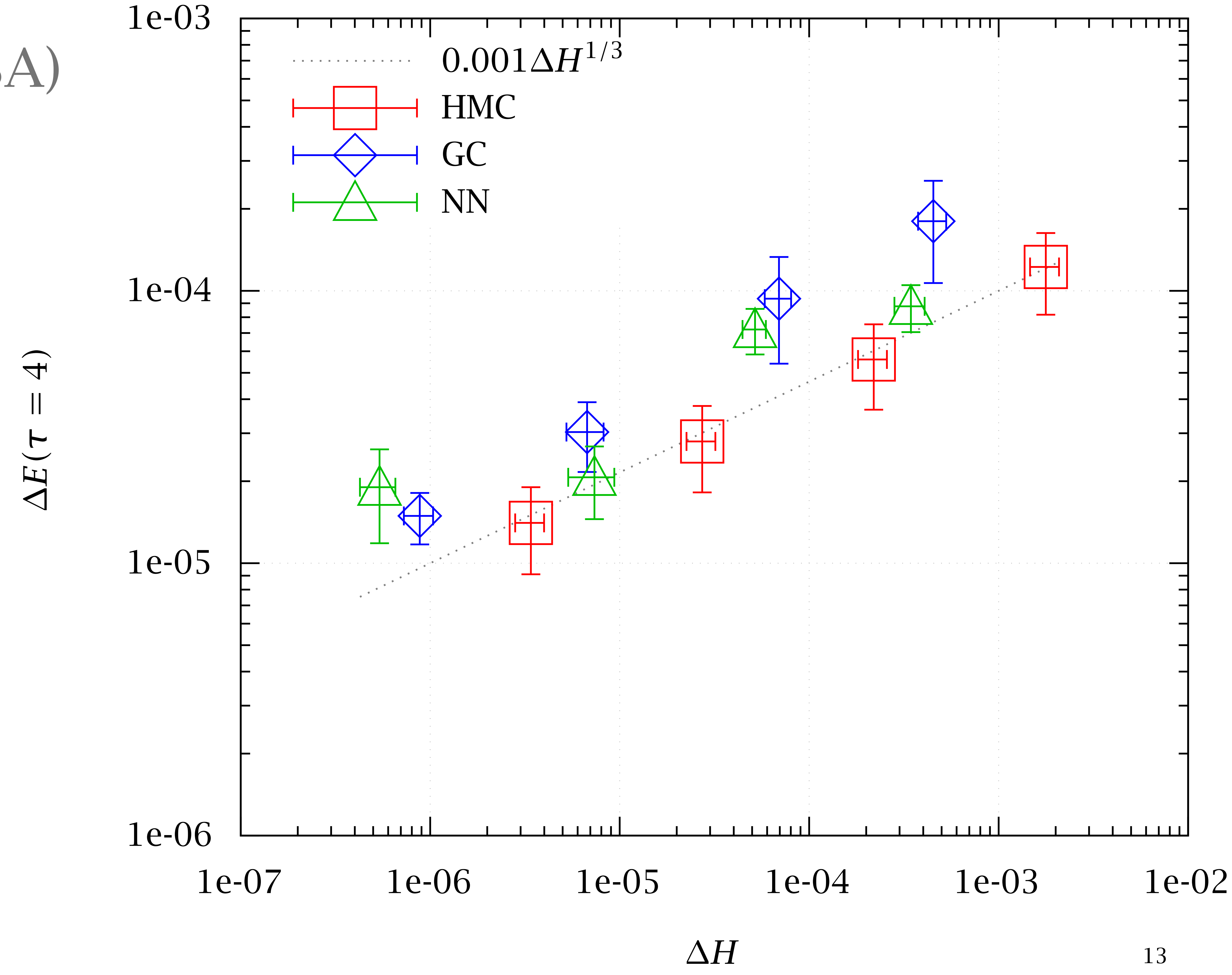
- (NN) Neural network local coefficients

```
[StoutSmearSlice(coeff=CoefficientNets([Dense(units=8, activation='swish'),
                                         Normalization(),
                                         Dense(units=54, activation=None)]), dir=dir, is_odd=eo)
    for _ in range(4) for dir in range(4) for _ in range(1) for eo in {False,True}]
```

  - Trained with 256 configs, 16 epochs, 11 hours, 20 GB GPU Mem
  - Single force evaluation: 1.1 sec, 8 GB GPU Mem

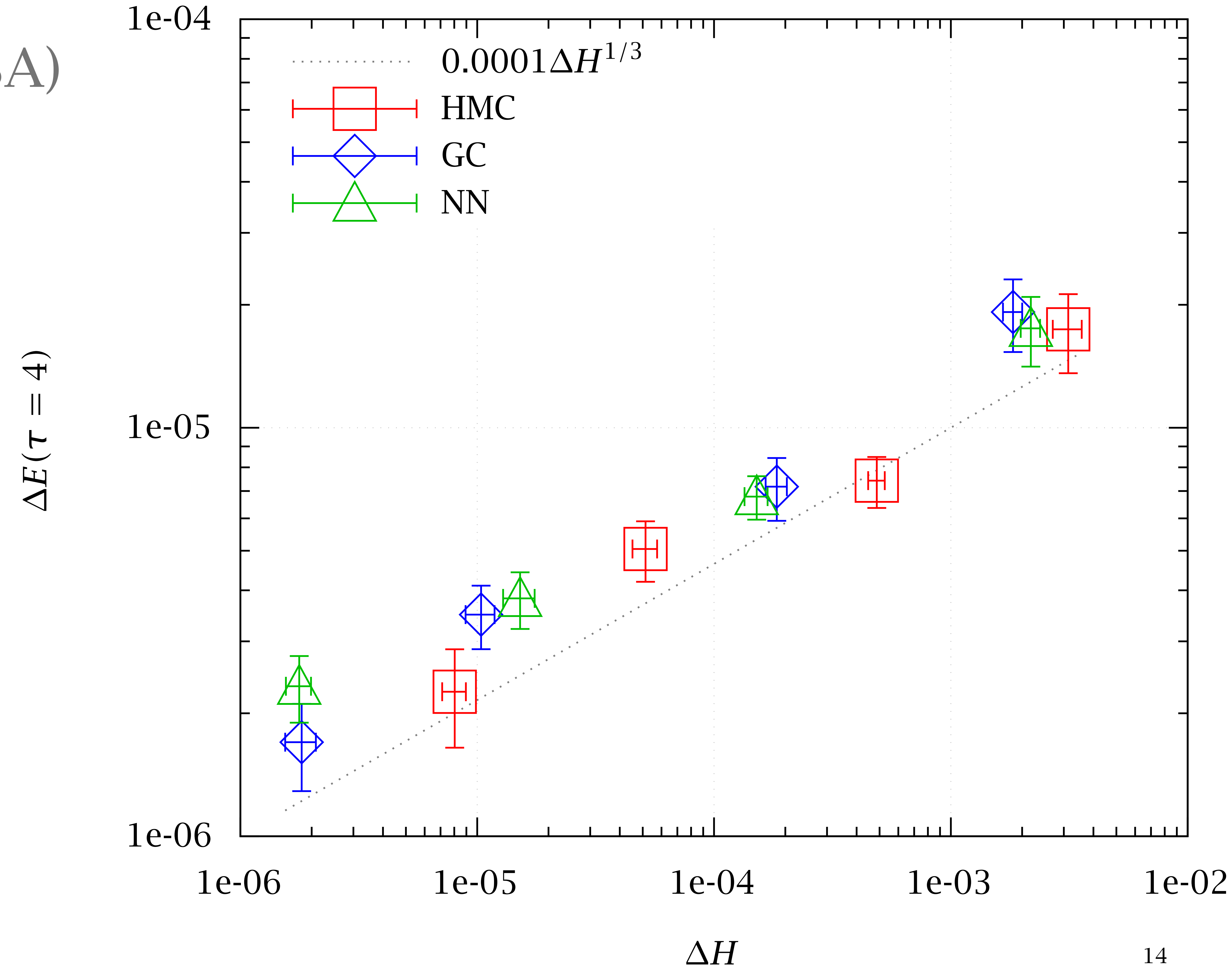
ΔH versus $\Delta E(\tau = 4)$, $8^3 \times 16$

- Single step Omelyan 2MN (ABABA)
- $\delta t = 0.0025, 0.005, 0.01, 0.02$
- 1 loop Clover Energy
- Wilson flow time, $\tau = 4$



ΔH versus $\Delta E(\tau = 4)$, $12^3 \times 24$

- Single step Omelyan 2MN (ABABA)
- $\delta t = 0.0025, 0.005, 0.01, 0.02$
- 1 loop Clover Energy
- Wilson flow time, $\tau = 4$



Code

- https://github.com/nftqcd/nthmc using TensorFlow
- Supports
  - Generalized stout smearing
  - HMC
  - Staggered D, CG solver
  - R/W SciDAC lime, R/O ILDG lime
- TensorFlow and Performance
  - Hard to optimize custom ops in Python
  - Autograd wastes memory
  - XLA helps (~100X) on Nvidia
  - Some takes >1hour JIT compiling

Outlook

- Trained field transformation to match the force with those from a stronger coupling
- When used as a change of variable the model provides reduction in ΔH by a factor of 4 and mild increase in $\Delta E(\tau = 4)$, in MD with a small step
- Directly applying $8^3 \times 16$ trained model to $12^3 \times 24$ and weaker coupling retains some benefit
- We need
  - Better evaluations for use in HMC
  - Frameworks that allows us better controls of performance, memory usage
  - Look for better transformation models