Neural Network Field Transformation and Its Application in HMC

Xiao-Yong Jin (Argonne National Laboratory)
LatticeQCD Exascale Computing Project

July 28, 2021
The 38th International Symposium on Lattice Field Theory, Zoom@MIT

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

Outline

- Field transformation, a.k.a. change of variables, a.k.a. contour deformation
- Construct gauge covariant field transformation with neural networks
- A test on 2D U(1) pure gauge:
  - Train the Field Transformation (FT) model at a coupling/volume
  - Run FTHMC using the model at other couplings/volumes
- Conclusion

Field Transformation and Trivializing Maps

- Change of variables: use a continuously differentiable bijective map $\mathscr{F}^{-1}$ from target field U to the mapped field $V = \mathscr{F}^{-1}(U)$, same group manifold for us

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

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

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

- Lüscher, 2010: construct $\mathscr{F}$ such that $S_{\mathrm{FT}}\big(V\big) = \text{const}$, a trivializing map

- More FTHMC tests:
  - $\mathscr{F}$ from stout smearing on 4D SU(3) pure gauge DBW2, Luchang Jin's Poster
  - $\mathscr{F}$ from MIT group's NVP flow (Sebastien Racaniere's Talk), Sam foreman's Poster

Gauge Covariant Link Update, Generalized for ML

- We know gauge covariant update, all the time in HMC and stout smearing, with a list of Wilson loops W_l

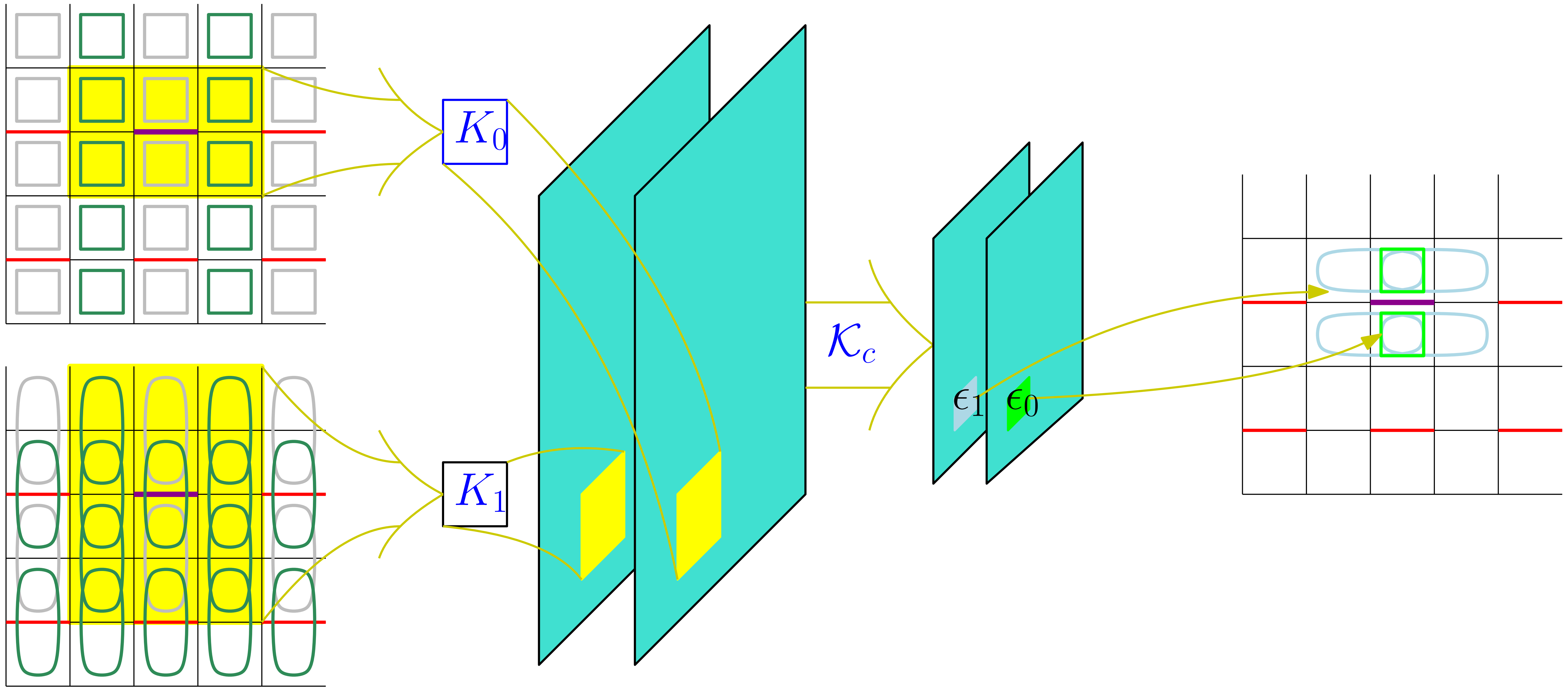
$$U_{x,\mu} \to U'_{x,\mu} = e^{\Pi_{x,\mu}} U_{x,\mu} \text{ where } \Pi_{x,\mu} = \sum_l \epsilon_l \partial_{x,\mu} W_l$$

- Generalize it for machine learning
  - Use stout smearing as neural networks, Akio Tomiya's talk
  - 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 a convolutional neural network, $\mathcal{N}_l$ is one of the output channels
    - $c \tan^{-1}[\,\cdot\,]$ ensures a positive definite Jacobian

Localized Coefficients, by Convolutional Neural Networks



- 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

Train Transformations, not Complete Trivializing Maps

- Do we need a complete trivializing map to improve HMC?

- Our test: training a transformation mapping the target field to a field distribution with the effective action similar to the gauge action at $\beta_{\text{map}} = 2.5$

  - On 2D U(1) pure gauge with the Wilson plaquette action

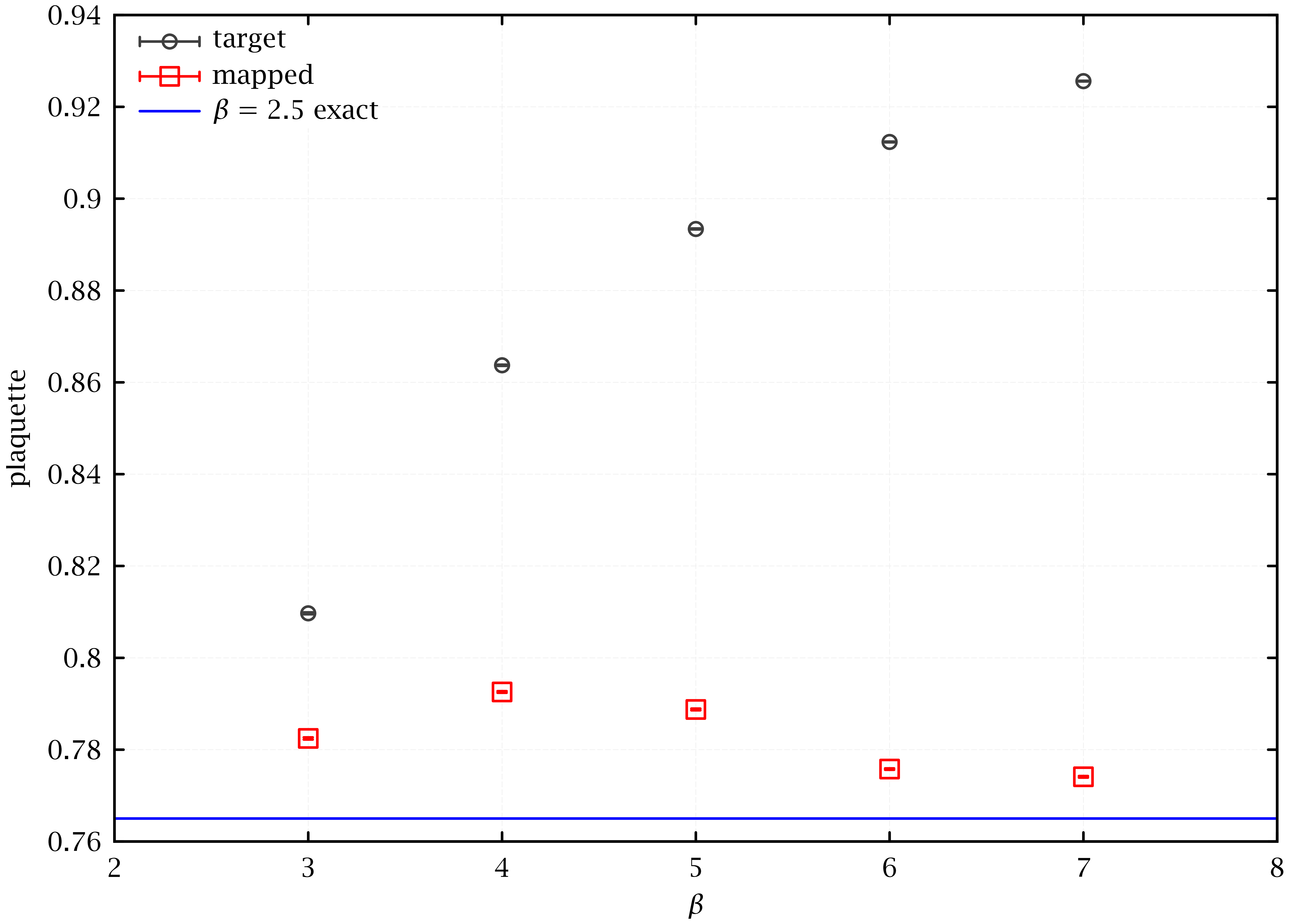
  - Use HMC to generate configurations at target $\beta = 3, 4, 5, \ldots$

  - Compute the force of the effective action on the generated configurations

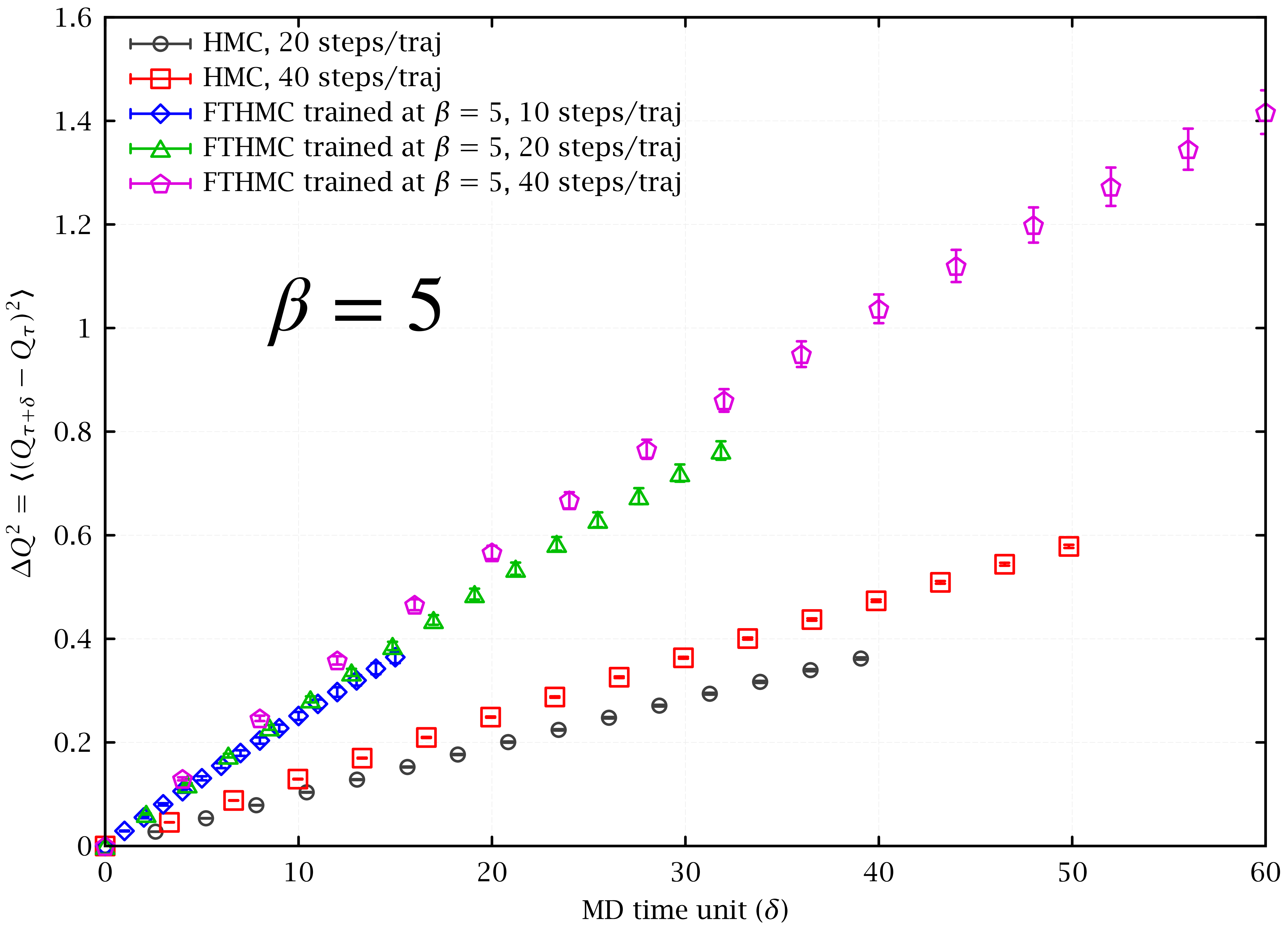
  - Minimize L2-norm and L∞-norm of the difference in the effective force and the force with $\beta_{\text{map}} = 2.5$

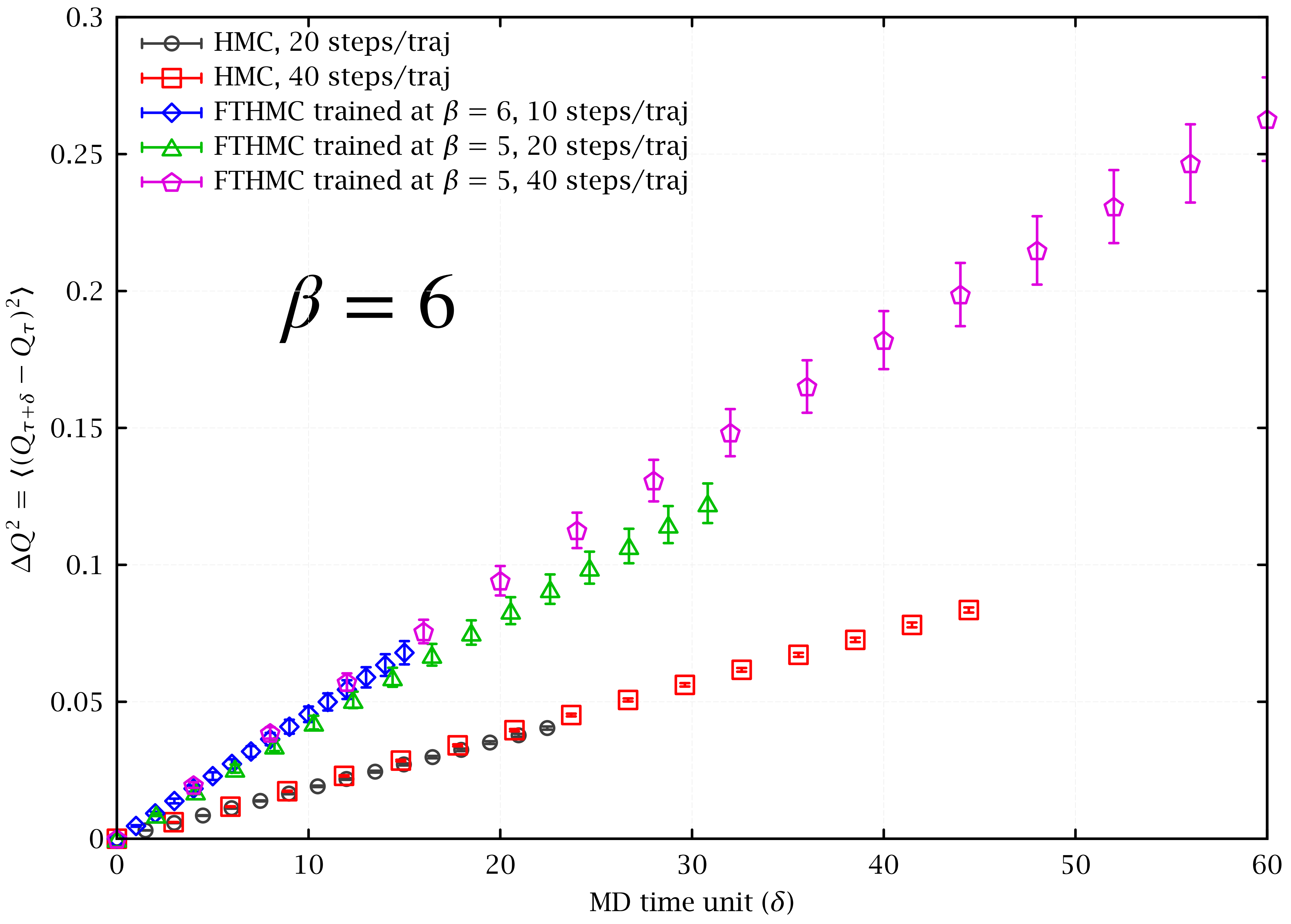
  - Transfer training the trained model at $\beta = 3$ to $\beta = 4$, and so on with increasing β

Plaquette Values with trained models, lattice size 16×16



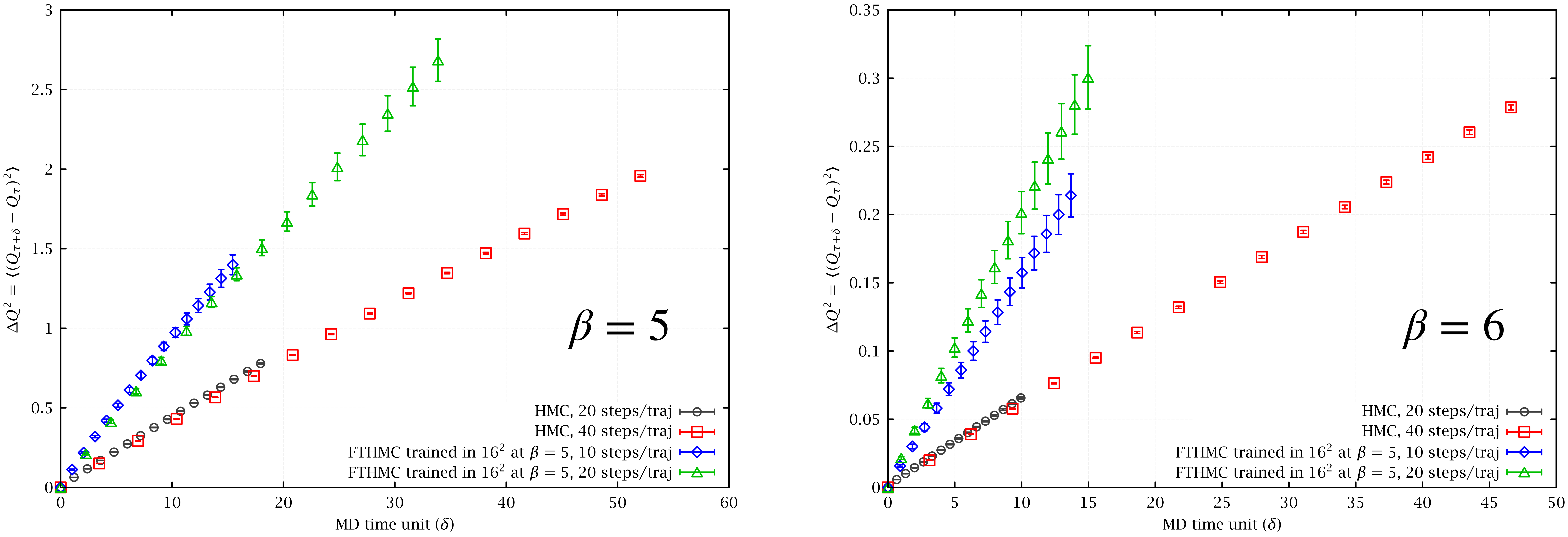
ΔQ^2 HMC with Neural Network Field Transformation, 16×16





Performs well with model trained at $\beta = 5$

ΔQ^2 FTHMC, lattice size 32×32, using model trained in 16×16



Poorly tuned acceptance rates that vary between 0.7 to 0.95

Numbers for Geeks

- Everything ran on 2012 Ivy Bridge i7-3770 at 3.40 GHz, 4 cores, 2 threads/core
- The test model uses 8 link update layers to update the whole lattice
- Each link update layer contains (each Conv2D layer has 2 channels),
  - 1 Conv2D, 3x2 kernel size, K_0, for traced plaquette loops
  - 1 Conv2D, 3x3 kernel size, K_1, for traced rectangle (2x1) loops
  - 2 Conv2D, 3x3 kernel size, K_c, for the combined filter to get coefficients
- For each β, the training uses batch size 64, 1024 training steps, each step takes 5 seconds (HMC included), uses about 8 GB; an inference (FTHMC) trajectory of 10 leapfrog steps takes less than 3 seconds
- The same code with identity transformation (HMC), 1 trajectory of 20 steps takes 0.6 sec for a batch of 2048
- For this code and this model, FTHMC is $\sim$300x slower than HMC

TensorFlow, the Good, the Bad, and the Ugly

- Good
  - Fast. Trace compiling of TensorFlow graph optimizes code and remove python overhead. A few factors faster than other ML frameworks.
- Bad
  - Limited. No periodic padding. No Conv4D. Derivatives w.r.t. input do not seem to use optimized code path and are memory intensive.
- Ugly
  - Non-Pythonic. Different semantics and cryptic error messages inside `tf.function`.

Conclusion and Outlook

- We propose a general construction of gauge covariant neural networks that is group-agnostic.
- We train a simple model to map 2D U(1) gauge configurations to a stronger coupling, and use the trained model in HMC with different values of coupling and lattice sizes and see improvement in tunneling of topological sectors.
- Code and extra goodies: `https://github.com/nftqcd/nthmc`
- Future
  - Careful study of scaling behavior to determine cost-effectiveness.
  - Explore other uses of tunable field transformations.
  - Software: optimize our code; may need to either restrict our field models to existing APIs, or invest in creating an optimized ML framework for lattice fields.