Exploring gauge-fixing conditions with gradient-based optimization*

William Detmold[1] Gurtej Kanwar[2] Yin Lin[1] Phiala E. Shanahan[1] Michael L. Wagman[3]

[1] Massachusetts Institute of Technology, [2] University of Edinburgh, [3] Fermi National Accelerator Laboratory

1. Motivation

- **Lattice gauge fixing** is necessary to calculate gauge-variant quantities, such as those needed for renormaliazation, where one connects to perturbative schemes by minimizing an **gauge-fixing functional** (e.g. Landau gauge).
- **Maximal-tree gauges** are useful to deform the path integral contour for improving the signal-to-noise ratio of observables like Wilson loops [1].
- Systematic parameterization and **exploration of gauge-fixing schemes** provides a useful tool in these contexts.

2. Approach

- We define a **general parameterization of a gauge-fixing functional**. Minimizing over the gauge orbit fixes to a gauge defined by the parameters.
- The parameterization includes Landau gauge, Coulomb gauge, and maximal tree gauges, as well as more general gauges.
- We introduce a **gradient-based algorithm** that allows us to optimize parameters defining the gauge-fixing scheme to minimize an arbitrary loss function.

3. Algorithm

General gauge-fixing functional

$$E \propto -\sum_{x,\mu} \mathrm{Tr}\left(p_\mu(x) U^g_\mu(x)\right) \quad \begin{cases} p_i(x) = 1 \text{ (Coulomb gauge)} \\ p_\mu(x) = 1 \text{ (Landau gauge)} \\ p_\mu(x) = k_\mu(x) \in \{0,1\} \text{ (Max. trees)} \end{cases}$$

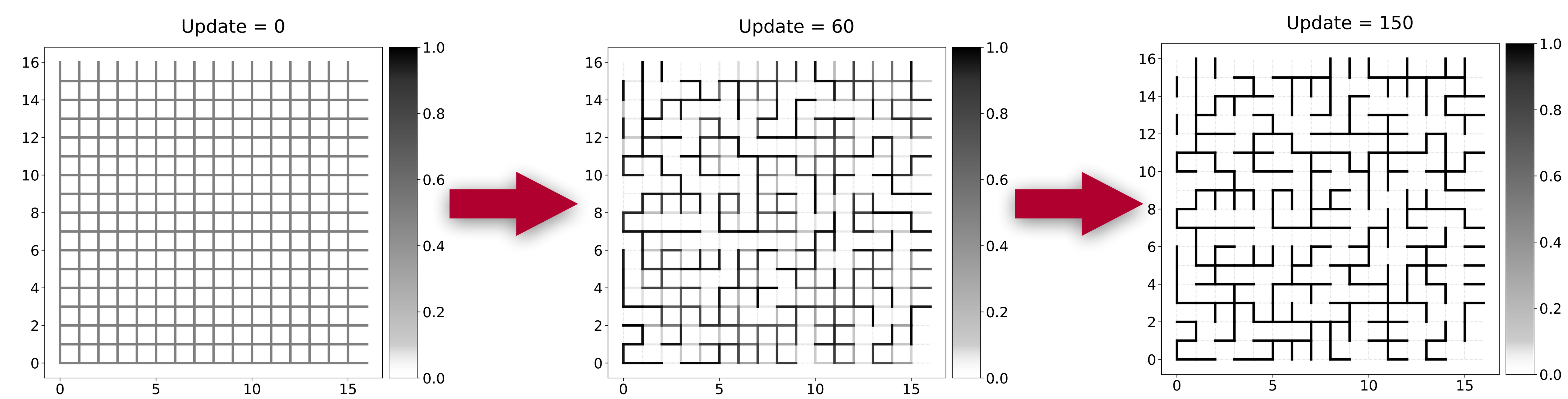
Gauge-fixed configurations, $U^g_\mu(x)$, are obtained by minimizing the functional. Given some objective function, $l[U^g_\mu(x)]$, **the adjoint-state method** [2] can be used to efficient compute the gradient $dl/dp_\mu(x)$ for optimizing the choice of $p_\mu(x)$.

To restrict exploration to only **maximal tree gauges**, one must restrict $p_\mu(x)$. Gradient-based optimization requires a smooth parameterization, thus define $p_\mu(x)$ to be **a "soft" tree to smoothly interpolate between maximal trees**:

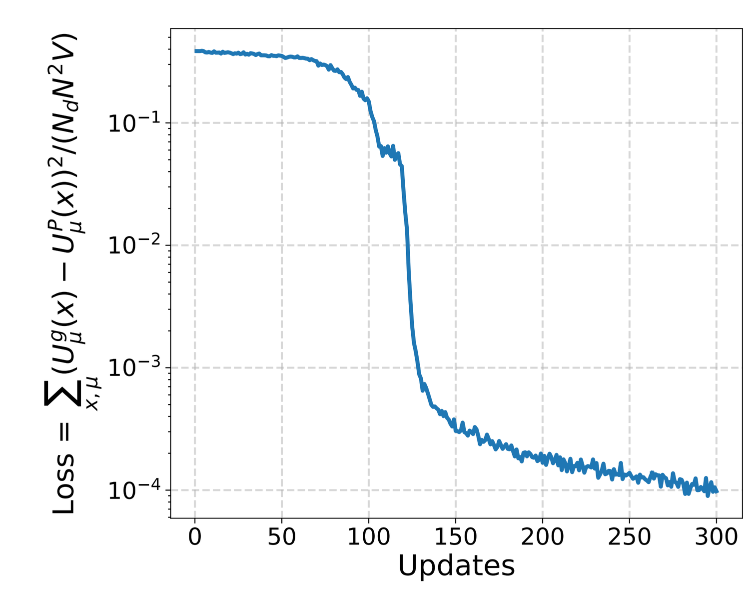
$$\boxed{p_\mu(x) = \text{soft tree} = \text{weighted sum of all maximal trees}}$$

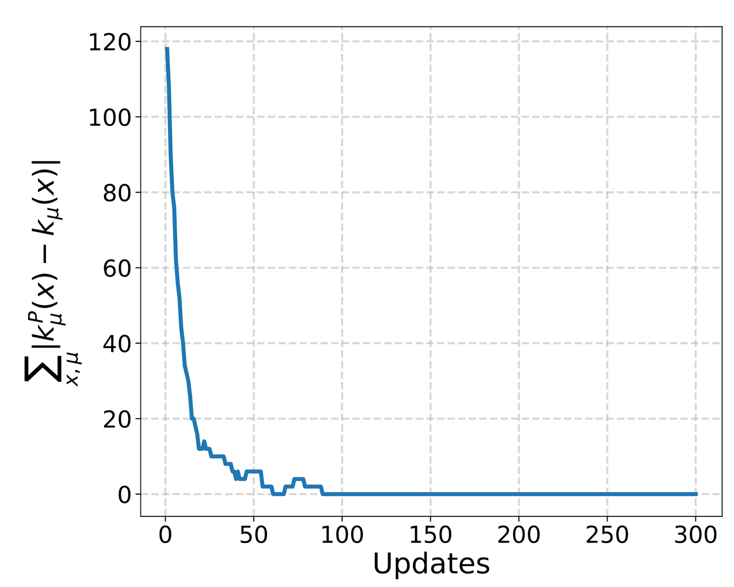
This can be done by using **Kirchhoff's matrix-tree theorem** [3]. The final maximal tree can be obtained by using **Kruskal's algorithm**.

4. Results



- Generate a random maximal tree specified by $k^P_\mu(x) \in \{0,1\}$ and obtain the corresponding $U^P_\mu(x)$
- Create a soft tree $p_\mu(x)$ using the matrix-tree theorem
- Gauge fix with $p_\mu(x)$ and minimize to obtain $U^g_\mu(x)$
- Minimize the loss function $$l \propto \sum_{x,\mu} \left(U^P_\mu(x) - U^g_\mu(x)\right)^2$$





* W. Detmold, G. Kanwar, Y.Lin, P.E Shanahan, M.L. Wagman, PoS LATTICE2024 (2025), arXiv:2410.03602 [hep-lat]

[1]: Y. Lin, W. Detmold, G. Kanwar, P. E. Shanahan, and M. L. Wagman PoS LATTICE2023 (2024) 043, arXiv:2309.00600 [hep-lat]

[2]: https://math.mit.edu/~stevenj/18.336/adjoint.pdf

[3]: Paulus, D. Choi, D. Tarlow, A. Krause, and C. J. Maddison Advances in Neural Information Processing Systems 33 (2020) 5691–5704, arXiv:2006.08063

The NSF Institute for Artificial Intelligence and Fundamental Interactions

SciDAC Scientific Discovery through Advanced Computing