PROCEEDINGS OF SCIENCE

CASK: A Gauge Covariant Transformer for Lattice Gauge Theory

Yuki Nagai,[a] **Hiroshi Ohno**[b] **and Akio Tomiya**[c,d,*]

[a] *Information Technology Center, University of Tokyo, Kashiwa, Chiba 277–0882, Japan*
[b] *Department of Advanced Materials Science, University of Tokyo, Kashiwa, Chiba 277-8561, Japan*
[c] *Center for Computational Sciences, University of Tsukuba, Tsukuba, Ibaraki 305-8577, Japan*
[d] *Department of Mathematics, Tokyo Woman's Christian University, Tokyo 167-8585, Japan*
[e] *RIKEN Center for Computational Science, Kobe 650-0047, Japan*
E-mail: nagai.yuki@mail.u-tokyo.ac.jp, hohno@ccs.tsukuba.ac.jp, akio@yukawa.kyoto-u.ac.jp

We propose a Transformer neural network architecture specifically designed for lattice QCD, focusing on preserving the fundamental symmetries required in lattice gauge theory. The proposed architecture is gauge covariant/equivariant, ensuring it respects gauge symmetry on the lattice, and is also equivariant under spacetime symmetries such as rotations and translations on the lattice. A key feature of our approach lies in the attention matrix, which forms the core of the Transformer architecture. To preserve symmetries, we define the attention matrix using a Frobenius inner product between link variables and extended staples. This construction ensures that the attention matrix remains invariant under gauge transformations, thereby making the entire Transformer architecture covariant. We evaluated the performance of the gauge covariant Transformer in the context of self-learning HMC. Numerical experiments show that the proposed architecture achieves higher performance compared to the gauge covariant neural networks, demonstrating its potential to improve lattice QCD calculations.

The 41st International Symposium on Lattice Field Theory (LATTICE2024)
28 July - 3 August 2024
Liverpool, UK

[*] Speaker

1. Introduction

Machine learning has become a powerful tool in high energy physics, where the computational cost of large-scale simulations is often prohibitively high. One promising approach involves the use of neural surrogate models, which serve as cheaper approximations to the exact theory and can significantly reduce computational cost [1]. In particular, gauge covariant/equivariant neural networks[1] are attracting considerable attention because they allow flexible and differentiable mappings between gauge fields, controlled by learnable parameters.

Machine learning (ML) techniques have recently made significant inroads into lattice QCD, offering novel strategies to tackle long-standing computational challenges. One notable direction is the development of flow-based sampling algorithms [3–9] and continuous flow approaches [10–12], which promise more efficient generation of gauge field configurations. ML has also begun to play a key role in non-equilibrium Monte Carlo [13–15]. Furthermore, perfect action techniques are being refined through advanced neural network methods [16], and preconditioning strategies employing neural networks have been shown to enhance numerical efficiency [17–19]. Taken together, these innovations highlight the rapidly evolving synergy between ML and lattice QCD, potentially broadening the scope of feasible calculations.

In order for a neural network architecture to be useful in lattice QCD, it must satisfy several important criteria without sacrificing either efficiency or physical rigor. It must be compatible with gauge symmetry, as gauge invariance is a cornerstone of QCD and its lattice formulation [20]. It must also be "fermion friendly" and accommodate modern lattice QCD simulations that incorporate dynamical fermions. Furthermore, the architecture must be fully differentiable to allow training via gradient-based methods, which have shown immense utility in machine learning [21–23].

Recent breakthroughs in deep learning suggest that Transformers, originally popularized in natural language processing [24] and known for their ability to capture non-local correlations, can be beneficial for problems in lattice QCD. Their core technology, the attention matrix, can handle long-range interactions [25], a feature particularly relevant in the presence of fermions. There is also growing interest in exploiting symmetries within the data, motivating research on how to scale equivariant architectures with Transformers [26].

Despite these promising directions, constructing a network that is both gauge covariant and capable of leveraging the flexibility of the Transformer paradigm remains a challenge. In this work, we address these issues by introducing a new neural network layer, *Covariant Attention with Stout Kernel* (CASK), which serves as a gauge covariant attention block. Our approach integrates the requirements of gauge covariance, differentiability, and fermion friendliness, while harnessing the ability of Transformer architectures to learn and exploit non-local correlations in lattice QCD.

2. Gauge covariant Transformer (CASK)

We first provide an overview of the gauge covariant neural network formalism, which serves as the foundation for constructing CASK.

[1] The conceptual foundations of these architectures draw on the distinction between equivariance and covariance [2].

2.1 Gauge covariant neural network

We first recall the gauge covariant neural network [27], which can be viewed as a trainable version of the stout smearing commonly used in lattice gauge theory. The gauge covariant neural network is also known as a residual flow [9]. In the following discussion, we focus on the simplest version of this gauge covariant neural network, namely the stout-type construction, which can be regarded as a convolutional layer acting on the links of the lattice gauge field.

To process gauge fields in a covariant manner, we consider an iterative evolution equation for the link variables $U_\mu^{(l)}(n) \in \mathrm{SU}(N_c)$ of the form

$$U_\mu^{(l+1)}(n) = g_{n,\mu}^{(l)} U_\mu^{(l)}(n), \tag{1}$$

where the gauge update

$$g_{n,\mu}^{(l)} = \exp\Big[\mathrm{i} \sum_f \rho^{(l,f)} Q_\mu^{(l,f)}(n)\Big] \in \mathrm{SU}(N_c) \tag{2}$$

encodes the action of the neural network at layer l. Here, l is the number of layers (or smearing levels), and f indicates the type of loops considered (for example, staples of various shapes). The real parameters $\rho^{(l,f)} \in \mathbb{R}$ are trainable weights.

Each $Q_\mu^{(l,f)}(n)$ is an element of the Lie algebra $\mathfrak{su}(N_c)$ constructed from a closed loop $\Omega_\mu(n)$ surrounding the link $U_\mu(n)$. Concretely,

$$Q_\mu(n) = \frac{\mathrm{i}}{2}\Big(\Omega_\mu^\dagger(n) - \Omega_\mu(n)\Big) - \frac{\mathrm{i}}{2\,N_c}\mathrm{Tr}\Big(\Omega_\mu^\dagger(n) - \Omega_\mu(n)\Big) \in \mathfrak{su}(N_c), \tag{3}$$

where $\Omega_\mu(n) \in \mathrm{SU}(N_c)$ is formed by the product of links associated with $U_\mu(n)$. In essence, this construction implements a local transformation that “smears” or modifies the original link $U_\mu^{(l)}(n)$ to produce $U_\mu^{(l+1)}(n)$, while ensuring that the operation remains gauge covariant. By design, this network preserves gauge symmetry and can be trained using a generalized backpropagation scheme adapted for matrix-valued variables.

Viewed this way, the stout-type gauge covariant neural network recasts a well-known smearing procedure as a trainable, parameterized transformation. This perspective not only bridges lattice smearing methods and modern deep learning, but also connects them in a unified framework.

2.2 Lesson from Transformer for spins

Before introducing gauge covariant Transformer, here we briefly review transformer for a classical O(3) spin model with quantum electrons in two dimensions [28, 29]. This is helpful to understand gauge covariant Transformer. Let $\vec{S}_n \in \mathbb{R}^3$ be a scalar field on lattice, which is a component of a classical spin. Here n indicates lattice site. Spin variables are normalized as $\sum_{\mu=1}^3 |\vec{S}_n|^2 = 1$ for all n. The Hamiltonian of the system is invariant under a spin rotation $\vec{S}_n \to R\vec{S}_n$ with $R \in \mathrm{O}(3)$. This transformation is independent of coordinate n (global symmetry).

As a first step of the procedure, we perform three different block spin transformations with different weights. The transformation is,

$$\vec{S}_n^{(\alpha)} = \sum_{k=0}^{6} \sum_{n' \in N_n^{(k)}} w_k^{(\alpha)} \vec{S}_{n'}, \tag{4}$$

for $\alpha = \mathrm{Q}, \mathrm{K}, \mathrm{V}$ and $w_k^{(\alpha)} \in \mathbb{R}$ is a weight. $N_n^{(k)}$ indicates a set of k-th neighbors for a lattice site n. We remark that this is covariant (equivariant) under global O(3) transformation for spin $S_{\mu,n}$. Next we construct an attention matrix. By using the standard inner product for two real vectors,

$$\tilde{M}_{n'n} = \sum_{\mu} \vec{S}_{n'}^{(\mathrm{K})} \cdot \vec{S}_n^{(\mathrm{Q})} = \sum_{\mu} \left(\vec{S}_{n'}^{(\mathrm{K})}\right)^{\top} \vec{S}_n^{(\mathrm{Q})}, \quad M_{n'n} = \mathrm{ReLu}(\tilde{M}_{n'n}) \tag{5}$$

Here $M_{n'n} \in \mathbb{R}_+$ is called an attention matrix and ReLu is the rectified linear function[2]. $M_{nn'}$ is obviously related to correlation functions. This matrix connects correlations between all points to all points, so it is a dense matrix. We emphasize that this is *invariant* under global O(3) transformation for spin $\vec{S}_n$. Next we construct a spin operator with attention,

$$\vec{S}_n^{(\mathrm{A})} = \sum_{n'} M_{nn'} \vec{S}_{n'}^{(\mathrm{V})}, \tag{6}$$

and this is *covariant/equivariant* under global O(3) transformation. This contains correlations from all points from this system. Finally, we construct output of the self-attention block,

$$\vec{S}'_n = \mathcal{N}(\vec{S}_n + \vec{S}_n^{(\mathrm{A})}) \tag{7}$$

$\mathcal{N}(\cdot)$ is a normalization operation to keep length of output vector to be one point-wisely. In neural network language, this is a layer normalization. This is equivariant under global O(3) as well. Whole procedure can be nested, which makes the network deeper.

Lessons are as follows. The attention matrix is essentially a correlation function, which allows us to capture long-range correlations. The attention matrix should be invariant under the symmetry transformation. So the output of the attention operation is covariant. Output should be normalized, otherwise we cannot regard the output of a Transformer block as a spin configuration. The normalization operation helps the training because we initialize weights with nearly zero and the attention block behaves as an identity. This enables greedy layer-wise training [30].

2.3 CASK (Gauge covariant Transformer)

Here we introduce CASK, which is a new Transformer specifically designed to process gauge configurations in a manner analogous to the covariant attention for spin systems, but with additional structure so that CASK is covariant under local gauge transformations. CASK can be regarded as a synthesis of two core ideas, namely the gauge covariant neural network and the O(3) equivariant Transformer. The central challenge in constructing a gauge covariant Transformer is the design of an invariant attention matrix.

To achieve an invariant attention matrix, we employ the Frobenius inner product $\mathrm{tr}(A^\dagger B)$ for complex matrices A and B. This quantity is invariant under the transformation $A \to \omega_1 A \omega_2$ and $B \to \omega_1 B \omega_2$ for $\omega_1, \omega_2 \in \mathrm{SU}(N_c)$, because

$$\mathrm{tr}(A^\dagger B) \;\to\; \mathrm{tr}\big(\omega_2^\dagger A^\dagger \omega_1^\dagger \omega_1 B \omega_2\big) \;=\; \mathrm{tr}(A^\dagger B).$$

The attention matrix in a Transformer is a collection of pairwise correlations, we would like to construct the attention matrix from gauge symmetric two-point functions. A naive extension from

[2] In the original Transformer uses softmax function instead of ReLu. ReLu helps to keep symmetry and numerically cheaper than the softmax. The latter is philosophically similar to the flash attention.

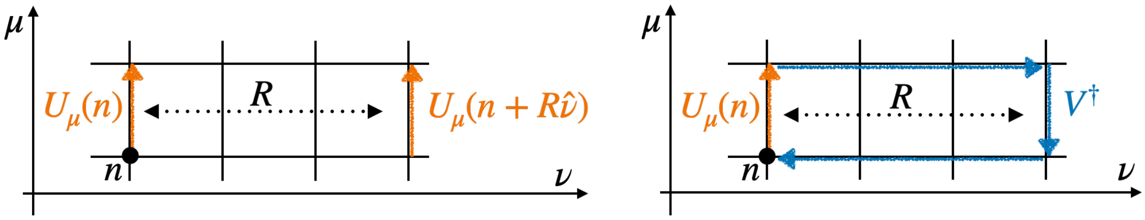

Figure 1: Construction of the attention matrix. *(Left):* Two links separated by R. *(Right):* Gauge covariant combination of two links separated by R.

the spin case might consider the two-point correlation of two links, as shown in the left panel of Fig. 1. However, that combination is not gauge symmetric. Instead, one must employ a Wilson loop, illustrated in the right panel of Fig. 1, which ensures gauge symmetry.

The definition of the CASK layer is as follows. To define the Transformer, we need to define three kinds of vectors. Smeared gauge links $U^{(Q)}$, $U^{(K)}$ and $U^{(V)}$, corresponding to the "query", "key" and "value" vectors, are defined as

$$U_\mu^{(Q)}(n) \equiv U_\mu^{(\alpha)}(n;\rho^{(\alpha)})\Big|_{\alpha\to Q}, \quad U_\mu^{(K)}(n) \equiv U_\mu^{(\alpha)}(n;\rho^{(\alpha)})\Big|_{\alpha\to K}, \quad U_\mu^{(V)}(n) \equiv U_\mu^{(\alpha)}(n;\rho^{(\alpha)})\Big|_{\alpha\to V}. \tag{8}$$

These are defined by gauge covariant layer using single plaquette with independent weights $\rho^{(\alpha)}$ in this work.

For general links field U, the extended staples $V_{\nu,n+\hat{\mu};s}(\{U\})$ as the functional of $\{U\}$ are defined as

$$V_{\nu,n+\hat{\mu};s}(\{U\}) \equiv \left(\prod_{t=0}^{s-1} U_\nu(n+\hat{\mu}+t\hat{\nu})\right) U_\mu^\dagger(n+\hat{\mu}+s\hat{\nu}) \left(\prod_{t=0}^{s-1} U_\nu^\dagger(n+(s-1-t)\hat{\nu})\right). \tag{9}$$

By using the staples, the attention matrix $a_{n,\mu,\nu,s}$ is defined as

$$a_{n,\mu,\nu,s} = \tan\left(\frac{4}{N_c}\tilde{a}_{n,\mu,\nu,s}\right), \tag{10}$$

$$\tilde{a}_{n,\mu,\nu,s} = \mathrm{Re\,Tr}\left[U_\mu^{(Q)}(n)V_{\nu,n+\hat{\mu};s}(\{U^{(K)}\})\right] - \mathrm{Re\,Tr}\left[U_\mu(n)V_{\nu,n+\hat{\mu};s}(\{U\})\right], \tag{11}$$

where the second term of (11) is the input of the l-th CASK layer. To amplify the signal, we introduce the tangent function. In principle, one could construct an all-to-all attention matrix, but this work uses a sparse attention matrix [25, 31] to reduce numerical cost. Specifically, links are connected within 1×1, 1×2 and 1×3 rectangular Wilson loops[3]. This attention matrix is gauge invariant.

A relation between $(l+1)$-th and l-th CASK layer is expressed as

$$U_\mu^{(l+1)}(n) \equiv e^{\mathrm{i}Q_\mu^A(n;\{\rho\delta_{\mu\nu}\})} U_\mu^{(l)}(n) \tag{12}$$

[3] This part is related to the definition of R in later sentence.

where

$$
\begin{aligned}
Q^A_\mu(n;\{a_{n,\mu,\nu,s}\}) = \frac{\mathrm{i}}{2}(\Omega^A_\mu(n;\{a_{n,\mu,\nu,s}\}) - \Omega^{A\dagger}_\mu(n;\{a_{n,\mu,\nu,s}\})) \\
- \frac{\mathrm{i}}{2N_c}\mathrm{Tr}\big(\Omega^{A\dagger}_\mu(n;\{a_{n,\mu,\nu,s}\}) - \Omega^A_\mu(n;\{a_{n,\mu,\nu,s}\})\big),
\end{aligned}
\tag{13}
$$

and

$$
\Omega^A_\mu(n;\{a_{n,\mu,\nu,s}\}) = C^A_\mu(n;\{a_{n,\mu,\nu,s}\})U^{(V)\dagger}_\mu(n), \tag{14}
$$

$$
\begin{aligned}
C^A_\mu(n;\{a_{n,\mu,\nu,s}\}) = \sum_{\nu\neq\mu}\sum_{s=1}^{R} a_{n,\mu,\nu,s}(U^{(V)}_\nu(n)U^{(V)}_\mu(n+\hat{\nu})U^{(V)\dagger}_\nu(n+\hat{\mu}) \\
+ U^{(V)\dagger}_\nu(n-\hat{\nu})U^{(V)}_\mu(n-\hat{\nu})U^{(V)}_\nu(x-\hat{\nu}+\hat{\mu}).
\end{aligned}
\tag{15}
$$

Here R is the same variable in Figure 1.

Training is done with backprop as the gauge covariant neural network, which is an extension of [27, 32].

2.4 Self-learning Hybrid Monte Carlo

To examine the expressibility of the Transformer, we perform self-learning Hybrid Monte Carlo (SLHMC) [27, 33], which incorporates an approximated model. In SLHMC, two different actions are involved: one is the exact action governing the target system, and the other is an approximate action used to guide the evolution. The acceptance rate in SLHMC is given by $\min(1, \exp[-(H' - H)])$ where the primed quantities indicate the updated configuration. SLHMC employs the approximate action in the molecular dynamics evolution, and, because the molecular dynamics trajectory is invertible, one can still perform a conventional accept-reject step based on the actual HMC Hamiltonian difference. As a result, we can get exact expectation values.

3. Lattice setup

In order to test the expressivity of CASK, we carry out simulations in SU(2) lattice gauge theory with dynamical fermions with SLHMC. As a proof-of-principle study, we use a 4^4 lattice at gauge coupling $\beta = 2.7$ and employ naive staggered fermions with mass $m = 0.3$. CASK is utilized to represent the effective action in SLHMC. Both the exact gauge action and the fermion action match those of the target system. However, during the molecular dynamics evolution, we replace the fermion mass $m = 0.3$ by a different mass $m^{\mathrm{eff}} = 0.4$ in the effective action and use CASK links in place of the thin links. CASK here relies solely on the plaquette kernel for smearing, but introduces three neighboring rectangular Wilson loops in the attention block to capture extended correlations. The intention is that CASK links absorb the difference arising from the modified Dirac operator.

All simulations are implemented using `Gaugefields.jl` and `LatticeDiracOperators.jl` in JuliaQCD [34], which are written in the Julia language [35]. The network parameters in CASK are trained via the Adam optimizer [36].

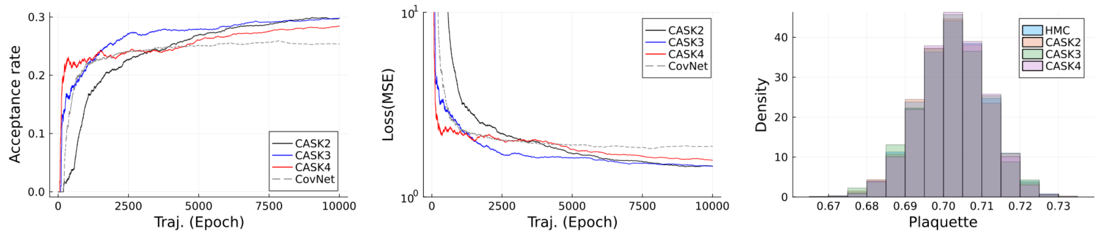

Figure 2: Comparison of algorithms. (*Left*) History of acceptance rate. (*Middle*) History of estimated loss function. (*Right*) Histogram of plaquette.

4. Results

We present our numerical findings in Fig. 2. In our setup, the SLHMC algorithm employs a Metropolis Hamiltonian $S = S_g + S_f[U, m]$ for the accept-reject step, while the molecular dynamics evolution uses an effective action $S = S_g + S_f[U^{\text{eff}}, m^{\text{eff}}]$. Here, U^{eff} is generated by a gauge covariant Transformer, and m^{eff} differs from the target fermion mass. Each Monte Carlo step corresponds to one epoch of training, during which the network parameters in the gauge covariant Transformer are updated. Without any training, the acceptance rate in this self-learning scheme is nearly zero, so any nonzero acceptance demonstrates that the network has sufficient expressive power to approximate the difference between $S_f[U, m]$ and $S_f[U^{\text{eff}}, m^{\text{eff}}]$.

The left panel of Fig. 2 shows the acceptance rate and the middle panel is the estimated loss function as a function of the epoch by a formula in [28]. Different colors represent distinct setups. CovNet indicates a result of the gauge covariant network and CASKn is CASK with n attention blocks. Over successive epochs, all networks learn to decrease the loss function. The gauge covariant neural network eventually saturates and ceases to improve, whereas the gauge covariant Transformer continues to learn at later epochs, achieving lower loss. This behavior highlights the Transformer's enhanced capability to capture non-local correlations and model the effective action more flexibly.

The right panel of Fig. 2 illustrates that, even while the Transformer's acceptance rate continues to grow, key observables remain consistent with expected physical behavior. This consistency confirms that the learned surrogate action does not distort essential physics, underscoring the practicality and reliability of gauge covariant Transformers in SLHMC.

5. Summary

In this work, we introduced the gauge covariant Transformer architecture CASK and demonstrated its utility in SLHMC simulations. By combining the essential features of gauge covariant neural networks with Transformer-based attention, CASK effectively incorporates both gauge symmetry and non-local correlations. In our numerical experiments, the surrogate links generated by CASK successfully absorbed the differences arising from the modified massive Dirac operator, resulting in an improved acceptance rate. The method consistently outperformed the gauge covariant neural networks ("adaptive stout") developed in our previous study, illustrating how the attention-based design can enhance expressivity. These findings suggest that the gauge covariant Transformer approach is a promising route toward more efficient and flexible simulations in lattice QCD. Future work will explore larger lattice volumes, extended loop structures in the attention matrix, and further optimization of the training process to fully leverage the potential of CASK.

Acknowledgments

The work of authors was partially supported by JSPS KAKENHI Grant Numbers 20K14479, 22K03539, 22H05112, and 22H05111, and MEXT as “Program for Promoting Researches on the Supercomputer Fugaku” (Simulation for basic science: approaching the new quantum era; Grant Number JPMXP1020230411, and Search for physics beyond the standard model using large-scale lattice QCD simulation and development of AI technology toward next-generation lattice QCD; Grant Number JPMXP1020230409). Numerical computation in this work was partially carried out at the Yukawa Institute Computer Facility.

References

[1] A. Adelmann *et al.*, *New directions for surrogate models and differentiable programming for High Energy Physics detector simulation*, 2203.08806.

[2] D. Marcos, M. Volpi, N. Komodakis, and D. Tuia, *Rotation equivariant vector field networks*, in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5058–5067.

[3] M. S. Albergo, G. Kanwar, and P. E. Shanahan, *Flow-based generative models for Markov chain Monte Carlo in lattice field theory*, Phys. Rev. D **100** (2019) 034515, [1904.12072].

[4] G. Kanwar, M. S. Albergo, D. Boyda, K. Cranmer, D. C. Hackett, S. Racanière, D. J. Rezende, and P. E. Shanahan, *Equivariant flow-based sampling for lattice gauge theory*, Phys. Rev. Lett. **125** (2020) 121601, [2003.06413].

[5] D. Boyda, G. Kanwar, S. Racanière, D. J. Rezende, M. S. Albergo, K. Cranmer, D. C. Hackett, and P. E. Shanahan, *Sampling using $SU(N)$ gauge equivariant flows*, Phys. Rev. D **103** (2021) 074504, [2008.05456].

[6] M. S. Albergo, G. Kanwar, S. Racanière, D. J. Rezende, J. M. Urban, D. Boyda, K. Cranmer, D. C. Hackett, and P. E. Shanahan, *Flow-based sampling for fermionic lattice field theories*, Phys. Rev. D **104** (2021) 114507, [2106.05934].

[7] M. S. Albergo, D. Boyda, K. Cranmer, D. C. Hackett, G. Kanwar, S. Racanière, D. J. Rezende, F. Romero-López, P. E. Shanahan, and J. M. Urban, *Flow-based sampling in the lattice Schwinger model at criticality*, Phys. Rev. D **106** (2022) 014514, [2202.11712].

[8] R. Abbott *et al.*, *Gauge-equivariant flow models for sampling in lattice field theories with pseudofermions*, Phys. Rev. D **106** (2022) 074506, [2207.08945].

[9] R. Abbott, A. Botev, D. Boyda, D. C. Hackett, G. Kanwar, S. Racanière, D. J. Rezende, F. Romero-López, P. E. Shanahan, and J. M. Urban, *Applications of flow models to the generation of correlated lattice QCD ensembles*, Phys. Rev. D **109** (2024) 094514, [2401.10874].

[10] P. de Haan, C. Rainone, M. C. N. Cheng, and R. Bondesan, *Scaling Up Machine Learning For Quantum Field Theory with Equivariant Continuous Flows*, 2110.02673.

[11] M. Gerdes, P. de Haan, C. Rainone, R. Bondesan, and M. C. N. Cheng, *Learning lattice quantum field theories with equivariant continuous flows*, SciPost Phys. **15** (2023) 238, [2207.00283].

[12] M. Gerdes, P. de Haan, R. Bondesan, and M. C. N. Cheng, *Continuous normalizing flows for lattice gauge theories*, 2410.13161.

[13] A. Bulgarelli, E. Cellini, K. Jansen, S. Kühn, A. Nada, S. Nakajima, K. A. Nicoli, and M. Panero, *Flow-based Sampling for Entanglement Entropy and the Machine Learning of Defects*, 2410.14466.

[14] A. Bulgarelli, E. Cellini, and A. Nada, *Sampling SU(3) pure gauge theory with Stochastic Normalizing Flows*, PoS **LATTICE2024** (2025) 040, [2409.18861].

[15] M. Caselle, E. Cellini, and A. Nada, *Numerical determination of the width and shape of the effective string using Stochastic Normalizing Flows*, 2409.15937.

[16] K. Holland, A. Ipp, D. I. Müller, and U. Wenger, *Machine learning a fixed point action for SU(3) gauge theory with a gauge equivariant convolutional neural network*, Phys. Rev. D **110** (2024) 074502, [2401.06481].

[17] C. Lehner and T. Wettig, *Gauge-equivariant neural networks as preconditioners in lattice QCD*, Phys. Rev. D **108** (2023) 034503, [2302.05419].

[18] C. Lehner and T. Wettig, *Gauge-equivariant pooling layers for preconditioners in lattice QCD*, Phys. Rev. D **110** (2024) 034517, [2304.10438].

[19] S. Calì, D. C. Hackett, Y. Lin, P. E. Shanahan, and B. Xiao, *Neural-network preconditioners for solving the Dirac equation in lattice gauge theory*, Phys. Rev. D **107** (2023) 034508, [2208.02728].

[20] S. Cuomo, V. S. di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, *Scientific machine learning through physics-informed neural networks: Where we are and what's next*, 2201.05624.

[21] S. Ruder, *An overview of gradient descent optimization algorithms*, 1609.04747.

[22] M. Blondel and V. Roulet, *The elements of differentiable programming*, 2403.14606.

[23] W. Moses and V. Churavy, *Instead of rewriting foreign code for machine learning, automatically synthesize fast gradients*, *Advances in neural information processing systems* **33** (2020) 12472.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, *Attention is all you need*, 1706.03762.

[25] T. Lin, Y. Wang, X. Liu, and X. Qiu, *A survey of transformers*, 2106.04554.

[26] J. Brehmer, S. Behrends, P. de Haan, and T. Cohen, *Does equivariance matter at scale?*, 2410.23179.

[27] Y. Nagai and A. Tomiya, *Gauge covariant neural network for quarks and gluons*, 2103.11965.

[28] Y. Nagai and A. Tomiya, *Self-learning Monte Carlo with equivariant Transformer*, J. Phys. Soc. Jap. **93** (2024) 114007, [2306.11527].

[29] A. Tomiya and Y. Nagai, *Equivariant transformer is all you need*, PoS (LATTICE2023) (2024) 001, [2310.13222].

[30] E. Belilovsky, M. Eickenberg, and E. Oyallon, *Greedy layerwise learning can scale to imagenet*, 1812.11446.

[31] R. Child, S. Gray, A. Radford, and I. Sutskever, *Generating long sequences with sparse transformers*, 1904.10509.

[32] C. Morningstar and M. J. Peardon, *Analytic smearing of SU(3) link variables in lattice QCD*, Phys. Rev. D **69** (2004) 054501, [hep-lat/0311018].

[33] Y. Nagai, M. Okumura, K. Kobayashi, and M. Shiga, *Self-learning hybrid monte carlo: A first-principles approach*, Phys. Rev. B **102** (2020) 041124.

[34] Y. Nagai and A. Tomiya, *JuliaQCD: Portable lattice QCD package in Julia language*, 2409.03030.

[35] J. Bezanson, S. Karpinski, V. B. Shah, and A. Edelman, *Julia: A fast dynamic language for technical computing*, 1209.5145.

[36] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 1412.6980.