Combinational-convolution for flow-based sampling algorithm

Akio Tomiya
International Professional University of Technology in Osaka
Faculty of Technology and Science,
International Professional University of Technology,
3-3-1, Umeda, Kita-ku, Osaka, 530-0001, Osaka, Japan
akio@yukawa.kyoto-u.ac.jp

Satoshi Terasaki
AtelierArith
980-0004 Miyagi Japan
s.terasaki@atelier-arith.jp

Abstract

We propose a new class of efficient layer called *CombiConv* (Combinational-convolution) that improves the acceptance rate for the flow-based sampling algorithm for quantum field theory on the lattice. CombiConv is made from a d-dimensional convolution out of lower k-dimensional $\binom{d}{k}$ convolutions and combining their outputs, and CombiConv has fewer parameters than the standard convolutions. Furthermore we apply CombiConv to the flow-based sampling algorithm, Furthermore, we find that for every $d = 2, 3, 4$-dimensional scalar ϕ^4 theory CombiConv for $k = 1$ achieves a higher acceptance rate than others.

1 Introduction

Quantum Chromo-dynamics on the lattice (lattice QCD) is one of the most successful theories in high energy physics [1], which describes the subatomic world fundamentally. Lattice QCD is a quantum field theory defined on discretized spacetime, and the discretization makes degrees of freedom finite, which enables us to perform numerical integration of the quantum expectation values with Markov chain Monte-Carlo. However, the ultraviolet cutoff in lattice QCD, namely finite lattice spacing, brings unwanted artifacts into results, which have to be eliminated by extrapolation. Finer lattice spacing gives smaller lattice artifacts, but it causes long autocorrelation among the Monte Carlo samples, which is called critical slowing down [2].

To make Monte-Carlo sampling efficient, a new algorithm with a shorter autocorrelation time is demanded. Algorithms with machine learning for lattice field theory have been investigated [3, 4, 5, 6, 7, 8, 9] to solve the critical slowing down problem, and the flow-based sampling algorithm has the potential to resolve it [10, 11] , which has been extended to systems with gauge fields [11, 12] and fermions [13, 14, 15].

The flow-based sampling algorithm, an algorithm with a generative model, is an exact algorithm of Monte-Carlo configuration generation for quantum field theories associated with the Metropolis–Hastings test. As shown in [10], a larger acceptance rate gives a shorter autocorrelation time on the flow-based sampling algorithm; thus, a model with a large acceptance rate is demanded.

The flow-based sampling algorithm works in any number of physical dimension, but the acceptance rate would decrease in higher dimensions since the Metropolis-Hastings test is sensitive to the number

Machine Learning and the Physical Sciences workshop, NeurIPS 2022.

of active degrees of freedom. There are several proposals to improve the acceptance rate [16, 17], but here we propose a novel way to improve it based on the global symmetry of the target system.

In this paper, we study a model architecture of convolutional layers for the flow-based sampling algorithm. We propose a class of efficient layer called *CombiConv* (Combinational-convolution) and apply it to d-dimensional scalar field theory. We confirm that our proposal achieves a higher acceptance rate in $d = 2, 3, 4$ than others.

2 Preliminary

In this section, we explain our proposal and setup for our numerical experiments.

2.1 Combinational-convolution

Consider a standard d-dimensional convolution which is parameterized by d-dimensional convolution kernel of size $(f, \ldots, f) \in \mathbb{R}^d$ for some positive integer f. CombiConv factorizes the convolution into lower k-dimensional ones (fig. 1). For d-dimensional input data, there are $\binom{d}{k} = d!/(k!((d-k)!))$ patterns to select k-axes $(i_1, \ldots, i_k)$ from d-axes. For each pattern we create a d-dimensional convolution kernel $(f_1, \ldots, f_d)$ such that $f_i = f$ if $i \in \{i_1, \ldots, i_k\}$ otherwise $f_i = 1$ and convolve the input data using this convolution kernel. In practice it is realized by a k-dimensional standard convolution parameterized by k-dimensional convolution kernel of size $(f, \ldots, f) \in \mathbb{R}^k$. Then we calculate the mean of these $\binom{d}{k}$-output and pass it to an appropriate activation function. We call this architecture CombiConv-a or CombiConv(d, k) if we need explicitly specify d and k. The implementation details can be found in GomalizingFlow.jl [18]

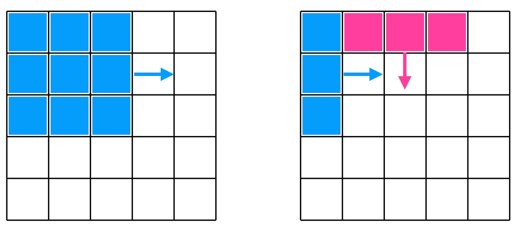

Figure 1: Schematic pictures of convolutional operations. (*left*) Conventional 3×3 convolutional operation on a two-dimensional image. (*right*) CombiConv operation with $d = 2$, $k = 1$ on a two-dimensional image. It consists of 3×1 convolution (blue) and 1×3 convolution (red) layers. Each convolution can be realized by one-dimensional convolution of kernel size 3.

By construction, the CombiConv layer have fewer parameters than that of the standard convolutional layer. The memory efficiency of parameters can be understood as follows. Let us consider d-dimensional configuration and let f be the kernel size for convolution. The number of parameters in the standard convolution is $W_{\text{std}} \propto f^d$ and our proposed CombiConv(d, k) is $W_{\text{CombiConv}} \propto \binom{d}{k} f^k$. For example, the efficiency is $W_{\text{std}}/W_{\text{CombiConv}} = 6.75$ for $d = 4$ and $k = 1$.

The CombiConv keeps translation, 90-degree planar rotations, and the reflection on the configuration while it suppresses unnecessary 45-degree planar rotations. In particular, for the case $k = 1$, the output of CombiConv removes the contributions of diagonal elements in each area that overlaps the kernel of the standard convolution.

Figure 2 shows the behavior of CombiConv in two dimension. (a) is an input example. (b) and (c) are output from the standard convolution with fixed weights (discretized Laplacian filter) and CombiConv with fixed weights (discretized Laplacian filter, $k = 1$), respectively. The standard one has extra points at 45-degree rotated points compared to CombiConv. To see the effect more clearly, we plot the difference between the output of a standard convolution and CombiConv for a two-dimensional feature map in (d). It shows CombiConv suppresses the contributions of diagonal elements of the center of the output feature compared to the standard convolution. Thanks to the architecture of CombiConv(d, k) we can implement convolutional layer using $k(< d)$-dimensional standard convolutions for an arbitrary d. In particular, for the case with more than three dimension, we can avoid using higher dimensional convolution which is not provided publicly by most of machine

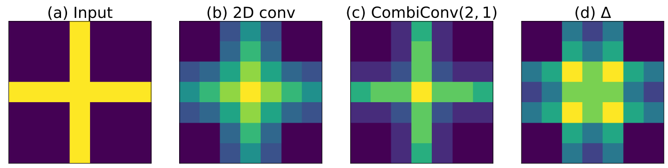

Figure 2: Comparison output of convolutional operations. *(a)* Input data with cross shape *(b)* Output of conventional convolutional for input data (a). *(c)* Output of CombiConv$(d = 2, k = 1)$ for input data (a). *(d)* Difference between output of (b) and (c)

learning frameworks and use optimized libraries for lower dimensional convolution. The code is implemented based on GomalizingFlow.jl [18], and calculations are performed on Google Cloud Platform with 4 vCPUs, 15 GB RAM and a single NVIDIA Tesla T4 (16GB) GPU.

2.2 Scalar field theory

Here we consider scalar field theory on a d-dimensional periodic square lattice of length L. Brief review of lattice QCD and scalar field theory is in Appendix. The partition function of the scalar field theory is,

$$Z = \int \left(\prod_{n=n_o}^{L^d} d\phi_n \right) \mathrm{e}^{-S^{(\mathrm{QFT})}[\phi]}, \quad S^{(\mathrm{QFT})}[\phi] = -\sum_{n=n_o}^{L^d} \phi(n) \partial^2 \phi(n) + \sum_{n=n_o}^{L^d} V[\phi](n), \tag{1}$$

where n_o is the origin, n is a point on the lattice. $\partial^2\phi(n) = \sum_{\mu=1}^{d} [\phi(n+\hat{\mu}) + \phi(n-\hat{\mu}) - 2\phi(n)]$ is a discretized Laplacian and, $\hat{\mu}$ is a unit vector for μ direction. $V[\phi](n) = m^2\phi^2(n) + \lambda\phi^4(n)$, is a potential term with mass $m^2 \in \mathbb{R}$ and coupling $\lambda \in \mathbb{R}_{\geq 0}$. Conventionally, we evaluate expectation values with eq. (1) with Hamiltonian/Hybrid Monte-Carlo [19], but here we utilize the flow-based sampling algorithm [10] and compare the acceptance rate.

3 Results

To examine the validity of CombiConv, we show how our architecture increase the acceptance rate. We construct the affine coupling layer as in [20] by replacing standard convolution with CombiConv. For our experiments, we use $m^2 = -4.0$ and $\lambda = 5.113$ (in the vicinity of critical regime) for each lattice size $L = 4, 8$ and the dimension of theory $d = 2, 3, 4$. We show results from $L = 8$ in this article.

The left panel of fig. 3 shows the history of acceptance rate as a function of training steps for two dimensions. Light blue, orange, and green lines are results for the standard convolution, CombiConv-b, and CombiConv-a, respectively. Improvement by CombiConv is relatively small in 2 dimensions, but CombiConv-a achieves the highest acceptance rate after 500 training steps.

The right panel of fig. 3 shows the history of acceptance rate in training steps for three dimensions. Light blue, orange, green, violet, and ocher line are results for the standard three-dimensional convolution, CombiConv-b ($d = 3, k = 1$), CombiConv-a ($d = 3, k = 1$), CombiConv-b ($d = 3, k = 2$), CombiConv-a ($d = 3, k = 2$), respectively. In three dimensions, CombiConv improves the acceptance rate significantly. Moreover, results with $k = 1$ show the highest acceptance rate.

We also perform calculations in four dimensions. We do not perform a simulation with the standard four-dimensional convolution since the library does not have it. In addition, due to the lack of numerical resource, we do not perform calculations with CombiConv-a. Figure 4 shows the history of acceptance rate in training steps for four dimensions. Light blue, orange, and green lines are results for CombiConv-b ($d = 4, k = 1$), CombiConv-b ($d = 4, k = 2$), CombiConv-b ($d = 4, k = 3$), respectively. As in the three-dimensional case, $k = 1$ shows the best performance.

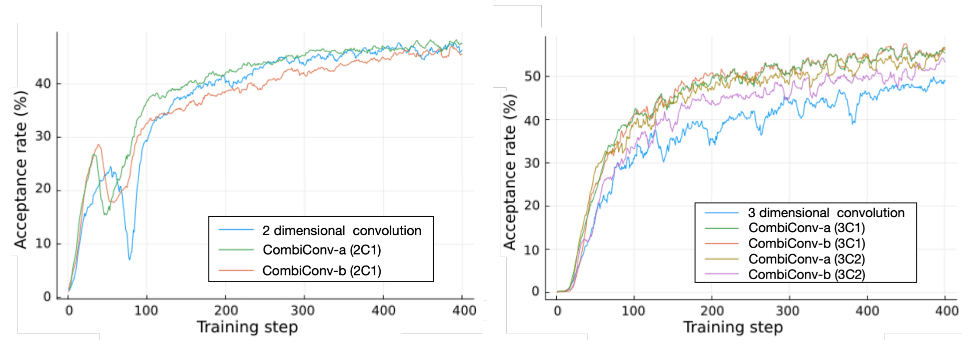

Figure 3: Acceptance rate for $L = 8$ in $d = 2, 3$. Label without $2C1$ is the standard 2-dimensional convolution and label without *activation_last* is CombiConv-b and the other is CombiConv-a. (*left*) 2-dimensional field theory. (*right*) 3-dimensional field theory. Label without $3Ck$ $(k = 1, 2)$ is the standard 3-dimensional convolution.

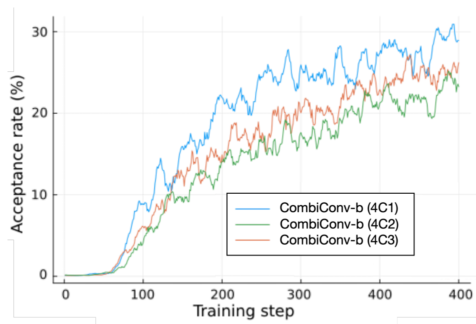

Figure 4: Acceptance rate for $L = 8$ in $d = 4$. No plot for the 4-dimensional conventional convolution and only alternative proposals (CombiConv-b) are shown (see main text).

4 Summary and Conclusion

In this work, we propose a novel way to improve the acceptance rate of the flow-based sampling algorithm with a decomposed convolutional layer. We construct d-dimensional convolution from $k(< d)$-dimensional convolution (CombiConv), which has fewer parameters in each layer than the standard one. CombiConv enables us to perform the convolutional operation using convolutional layers in the lower dimension, which is optimally implemented in many libraries. We perform numerical experiments for the scalar field theory with the flow-based sampling algorithm with CombiConv in $d = 2, 3, 4$. We find that the algorithm with CombiConv shows a higher acceptance rate for $d = 2, 3$ than the standard one, and CombiConv with $k = 1$ shows the best performance for $d = 2, 3, 4$.

The impact statement

This work provides a novel way to improve the acceptance rate (approximately efficiency) of the flow-based sampling algorithm for quantum field theory on the lattice in four dimensions. Metropolis-type algorithms, including the flow-based sampling algorithm, could decrease in higher dimensions because the acceptance rate depends on active degrees of freedom in the system. In addition, most libraries for machine learning do not provide a convolutional layer for more than three dimensions publicly. Our framework, associated with the global symmetry of the system, enables to use of an optimized library for lower dimensional convolution. Moreover, the implementation convenience and our framework show a higher acceptance rate in the Monte-Carlo calculations with the flow-based sampling algorithm.

Acknowledgments and Disclosure of Funding

This work of AT was supported by JSPS KAKENHI Grant Number JP20K14479, JP22H05112, JP22H05111, and JP22K03539.

References

[1] P. A. Zyla et al. Review of Particle Physics. *PTEP*, 2020(8):083C01, 2020.

[2] Stefan Schaefer, Rainer Sommer, and Francesco Virotta. Critical slowing down and error analysis in lattice QCD simulations. *Nucl. Phys. B*, 845:93–119, 2011.

[3] Akinori Tanaka and Akio Tomiya. Towards reduction of autocorrelation in HMC by machine learning. 12 2017.

[4] Kai Zhou, Gergely Endrodi, Long-Gang Pang, and Horst Stöcker. Generative Model Study for 1+1d-Complex Scalar Field Theory. *PoS*, AISIS2019:007, 2020.

[5] Jan M. Pawlowski and Julian M. Urban. Reducing Autocorrelation Times in Lattice Simulations with Generative Adversarial Networks. *Mach. Learn. Sci. Tech.*, 1:045011, 2020.

[6] Yuki Nagai, Akinori Tanaka, and Akio Tomiya. Self-learning Monte-Carlo for non-abelian gauge theory with dynamical fermions. 10 2020.

[7] Akio Tomiya and Yuki Nagai. Gauge covariant neural network for 4 dimensional non-abelian gauge theory. 3 2021.

[8] Sam Foreman, Taku Izubuchi, Luchang Jin, Xiao-Yong Jin, James C. Osborn, and Akio Tomiya. HMC with Normalizing Flows. *PoS*, LATTICE2021:073, 2022.

[9] Sam Foreman, Xiao-Yong Jin, and James C. Osborn. LeapfrogLayers: A Trainable Framework for Effective Topological Sampling. *PoS*, LATTICE2021:508, 2022.

[10] 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(3):034515, 2019.

[11] Gurtej Kanwar, Michael S. Albergo, Denis Boyda, Kyle Cranmer, Daniel C. Hackett, Sébastien Racanière, Danilo Jimenez Rezende, and Phiala E. Shanahan. Equivariant flow-based sampling for lattice gauge theory. *Phys. Rev. Lett.*, 125(12):121601, 2020.

[12] Denis Boyda, Gurtej Kanwar, Sébastien Racanière, Danilo Jimenez Rezende, Michael S. Albergo, Kyle Cranmer, Daniel C. Hackett, and Phiala E. Shanahan. Sampling using $SU(N)$ gauge equivariant flows. *Phys. Rev. D*, 103(7):074504, 2021.

[13] Michael S. Albergo, Gurtej Kanwar, Sébastien Racanière, Danilo J. Rezende, Julian M. Urban, Denis Boyda, Kyle Cranmer, Daniel C. Hackett, and Phiala E. Shanahan. Flow-based sampling for fermionic lattice field theories. *Phys. Rev. D*, 104(11):114507, 2021.

[14] Ryan Abbott et al. Gauge-equivariant flow models for sampling in lattice field theories with pseudofermions. 7 2022.

[15] Ryan Abbott et al. Sampling QCD field configurations with gauge-equivariant flow models. In *39th International Symposium on Lattice Field Theory*, 8 2022.

[16] Pim de Haan, Corrado Rainone, Miranda C. N. Cheng, and Roberto Bondesan. Scaling Up Machine Learning For Quantum Field Theory with Equivariant Continuous Flows. 10 2021.

[17] Michele Caselle, Elia Cellini, Alessandro Nada, and Marco Panero. Stochastic normalizing flows as non-equilibrium transformations. *JHEP*, 07:015, 2022.

[18] Akio Tomiya and Satoshi Terasaki. GomalizingFlow.jl: A Julia package for Flow-based sampling algorithm for lattice field theory. 8 2022.

[19] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid Monte Carlo. *Phys. Lett. B*, 195:216–222, 1987.

[20] Michael S. Albergo, Denis Boyda, Daniel C. Hackett, Gurtej Kanwar, Kyle Cranmer, Sébastien Racanière, Danilo Jimenez Rezende, and Phiala E. Shanahan. Introduction to Normalizing Flows for Lattice Field Theory. 1 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [N/A]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [N/A]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [No]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Appendix: Lattice field theory and QCD

Here let us briefly review lattice field theory and lattice QCD. In addition, we note the necessity of an efficient sampling method for it.

Quantum field theory is a fundamental language to describe quantum and relativistic elementary particles. In the path integral formalism in the euclidean spacetime, expectation values can be evaluated by the following integral formula,

$$\langle O \rangle = \frac{1}{Z} \int \mathcal{D}\phi \; \mathrm{e}^{-S[\phi]} \tag{2}$$

where $\phi(x) \in \mathbb{R}$ and $x \in \mathbb{R}^4$ is a real classical field (roughly speaking, it is a function on $\mathcal{R}^4$). $S[\phi] = \int d^4x(-\phi\partial^2\phi/2 + V(\phi)$. The integral measure $\mathcal{D}\phi$ is

$$\mathcal{D}\phi = \prod_{x\in\mathbb{R}^4} d\phi(x). \tag{3}$$

This integral measure is not well defined. Here we descretize the system as,

$$\partial^2\phi(n) = \frac{1}{a^2}\sum_{\mu=1}^{d}[\phi(an + a\hat{\mu}) + \phi(an - a\hat{\mu}) - 2\phi(an)] \tag{4}$$

We get (1). We remark that we set $a = 1$ in the main text for notational simplicity. In this case, $a \to 0$ limit is rephrased as a limit of other parameters in the system. To this end, we get field theory on the lattice. If we define the system in the finite box with a discretized one, the system has a finite number of degrees of freedom, and we can calculate the integral with computers. This finite discretization size is called lattice spacing, regarded as a cutoff for a high energy regime in the momentum space. We remark that, we can reduce the descretization error by taking small a in (4).

If we consider a function for $U_\mu(n) \in \mathrm{SU}(3)$ where $\mu = 1, 2, 3, 4$ and n is a coordinate on the lattice, instead of $\phi(n)$, (1) is expectation values for lattice QCD. $U_\mu(n)$ is defined on links (bonds) in a four-dimensional hypercube, and $U_\mu(n)$ is understood as gluons in elementary particles. Throughout of this paper, we focus on scalar field theory on the lattice.

In practice, we evaluate the expectation value using a numerical integration scheme. However, the trapezoidal integration or its improved one does not work because the dimensionarity of integration is too high (typically it is like $L^d = 10,000$ in (1) for $d = 4$ in practical cases and the error is roughly proportional to $O(n^{-1/L^d})$ for n pieces of trapezoid). Markov-Chain Monte Carlo (MCMC) does work for any number of integral variable, even more than $10,000$. MCMC generates samples from the probability distribution, which has high probability. For example, if we integrate with $\exp[-(x-\mu)^2]$ with respect to x, samples of MCMC are mostly from a regime around $x \approx \mu$. The integration error from MCMC is typically $O(1/\sqrt{N_{\mathrm{smp}}})$ where N_{smp} is the number of independent samples.

For lattice field theory, we use HMC (Hamiltonian/Hybrid Monte Carlo), a sophisticated MCMC for continuous variables. HMC is succeeded property of MCMC, and the integration error from MCMC is typically $O(1/\sqrt{N_{\mathrm{smp}}})$ where N_{smp} is the number of independent samples.

HMC applying the critical regime of the system has a problem called critical slowing down. Samples are generated with the equation of motion for an augmented physical system, but the equation shares the criticality of the original system. Thus, generated samples are correlated, and this makes the efficiency of MCMC down. This means that, if we fix the numerical integration time, the statistical error from MCMC increases. The continuum limit, namely $a \to 0$ limit, is a typical example of the critical regime. So we have a dilemma. To reduce the discretization error, we want to shrink a, but it increases the statistical error from MCMC and vice versa. To solve this problem, we need a new algorithm with milder critical slowing.