Control variates with neural networks

Hyunwoo Oh[*]

*Department of Physics, University of Maryland,
College Park, MD 20742, USA*

E-mail: hyunwooh@umd.edu

The precision of lattice QCD calculations is often hindered by the stochastic noise inherent in these methods. The control variates method can provide an effective noise reduction but are typically constructed using heuristic approaches, which may be inadequate for complex theories. In this work, we introduce a neural network-based framework for parametrizing control variates, eliminating the reliance on manual guesswork. Using $1 + 1$ dimensional scalar field theory as a test case, we demonstrate significant variance reduction, particularly in the strong coupling regime. Furthermore, we extend this approach to gauge theories, showcasing its potential to tackle signal-to-noise problems in diverse lattice QCD applications.

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

---

[*]Speaker

1. Introduction

Lattice Quantum Chromodynamics (QCD) using Monte Carlo methods has achieved remarkable success across a wide range of problems. However, these methods face significant limitations, particularly when dealing with large baryon chemical potentials or real-time dynamics, due to the well-known sign problem. Even in the absence of a baryon chemical potential and under thermodynamic equilibrium, certain observables are plagued by the signal-to-noise problem, which can render Monte Carlo calculations exponentially challenging.

A novel approach to reducing variance is by the construction of a control variate [1]. The idea of control variates is simple. If an observable O is given, an unbiased estimator $O - f$ can be constructed, provided that f satisfies $\langle f \rangle = 0$. However, their variances can be different:

$$\langle (O - f)^2 \rangle = \langle O^2 \rangle + \langle f^2 \rangle - 2 \langle Of \rangle. \tag{1}$$

which implies $\text{Var}(O - f) = \text{Var}(O) - 2 \langle Of \rangle + \langle f^2 \rangle$. Therefore, finding f, referred to as a control variate, that is highly correlated with O and has a small expectation value of its square, can potentially reduce the variance of $O - f$.

In general, identifying an operator with a zero expectation value, ensuring that the new observable $\tilde{O} \equiv O - f$ remains unbiased, is a challenging task. One way to find such operators are through the Stein's method [2] (or so-called the lattice version of the Schwinger-Dyson equations [1, 3]):

$$\int D\phi \frac{\delta}{\delta\phi}(g \mathrm{e}^{-S}) = 0 \tag{2}$$

for a well-defined g with a proper boundary condition. Therefore, if a control variate f is defined as $f \equiv \partial g - g \partial S$, it directly follows that $\langle f \rangle = 0$. The objective then becomes finding a function g which minimizes Eq. (1). In this work, we study the control variates method with neural network parametrizations, by summarizing and extending the work in [4].

2. Neural network

While there are attempts to find an optimal g from perturbative ways [1, 3], one can directly parametrize g with a neural network. Neural networks provide a framework for parametrizing functions through a sequence of affine transformations followed by activation functions. Since neural networks are known to be universal approximators [5, 6], they offer a promising approach for identifying more effective choices for g.

The general structure of a neural network is as follows:

$$g(\phi) = W_n \sigma_n(\cdots \sigma_1(W_0 \phi + b_0)) + b_n \tag{3}$$

where W_i and b_i represent affine transformations, and σ_i are activation functions which is non-linear in general. In our case, the input ϕ corresponds to a single configuration on the lattice. When there is no hidden layer, Eq. (3) corresponds to an affine transformation.

2.1 Symmetry

In principle, neural networks can learn the symmetries of the control variates, which are essential for achieving optimal results. However, because the model is trained only a finite number of times using a limited set of training samples, it is advisable to impose constraints on the neural network or on the construction of the control variates. This ensures that the symmetry is built into the framework, reducing the burden on the neural network to learn it during training.

Among many symmetries, the first example is translational symmetry. If both the action and the observable respect the translational symmetry, the control variate f also must preserve it:

$$f(T_y[\phi]) = f(\phi), \tag{4}$$

where T_y represents a translation with an amount y: $T_y[\phi_x] = \phi_{x+y}$.

In this section, we will use the Schwinger-Dyson control variates from a vector-valued function g with the following form:

$$f(\phi) = \sum_i \left(\frac{\partial g_i}{\partial \phi_i} - g_i \frac{\partial S}{\partial \phi_i} \right) \text{ where } g : \mathbb{R}^V \to \mathbb{R}^V, \tag{5}$$

where V denotes the degrees of freedom. Other constructions of the Schwinger-Dyson control variates and their universality will be discussed in Section 5.

In order for Eq. (5) to respect the translational symmetry, g should be translationally covariant (or equivariant):

$$g(T_y[\phi])_x = g(\phi)_{x+y}. \tag{6}$$

Note that $g(\phi)_x$ denotes the xth component of the vector-valued function g at ϕ.

The translational covariance of g can be enforced by first defining a real-valued function $g_0 : \mathbb{R}^n \to \mathbb{R}$, which can be parametrized b a neural network, and then constructing a vector-valued function g by

$$g(\phi)_x = g_0(T_x[\phi]). \tag{7}$$

With this construction, one can easily check that the control variate f is translationally invariant:

$$
\begin{aligned}
f(T_y[\phi]) &= \sum_x \left(\frac{\partial g(T_y[\phi])_x}{\partial \phi_{x+y}} - g(T_y[\phi])_x \frac{\partial S(T_y[\phi])}{\partial \phi_{x+y}} \right) \\
&= \sum_x \left(\frac{\partial g_0(T_{x+y}[\phi])}{\partial \phi_{x+y}} - g_0(T_{x+y}[\phi]) \frac{\partial S(T_{x+y}[\phi])}{\partial \phi_{x+y}} \right) = f(\phi).
\end{aligned}
\tag{8}
$$

Moreover, this construction does not depend on the structure of neural networks.

Another example is the Z_2 symmetry. It is satisfied if g is an odd function: $g(-\phi) = -g(\phi)$. To ensure that the neural network in Eq. (3) is odd, the affine layers $\phi \to W_i \phi + b_i$ must have zero biases, i.e., $b_i = 0$. Additionally, the activation functions must be odd: $\sigma_i(-x) = -\sigma_i(x)$ for each i. Suitable activation functions include arcsinh or tanh. In this work, all activation functions are chosen to be arcsinh, as its range is $(-\infty, \infty)$.

2.2 Training and overfitting

The ideal loss function, L, for optimizing the performance of control variates is the variance of $\tilde{O} = O - f$. Since $\langle O - f \rangle = \langle O \rangle$, which is independent of the neural network parameters, this constant term can be omitted. Consequently, the loss function simplifies to:

$$L(w) = \frac{1}{N} \sum_{i=1}^{N} (O(\phi_i) - f(\phi_i))^2 , \tag{9}$$

where w represents all the network parameters, such as W_i and b_i, and N denotes the number of training configurations.

Since the number of training configurations is finite (typically on the order of thousands in lattice calculations) and neural networks usually have a large number of parameters, overfitting is a common issue. In our case, overfitting imposes $O(\phi_i) = f(\phi_i)$ for every training configuration, which leads the sample average to be $\bar{f} \approx \bar{O}$, which is generally not zero. This issue was recognized in [7], where the authors suggested introducing an auxiliary parameter μ:

$$L(w, \mu) = \frac{1}{N} \sum_{i=1}^{N} (O(\phi_i) - f(\phi_i) - \mu)^2 . \tag{10}$$

In this way, when overfitting occurs, the auxiliary parameter μ may absorb the sample average of O, $\bar{O}$, ensuring that $\bar{f}$ remains close to zero. Note that during training using gradient descent, the auxiliary parameter μ is updated along the network parameters w. However, since μ is not used for evaluating observables, the new observable $\tilde{O} = O - f$ remains unbiased.

Therefore, in this proceeding, neural networks are optimized by minimizing the loss function in Eq. (10), with the *ADAM* optimizer [8] with exponentially decaying learning rate from 10^{-3} to 10^{-6}.

3. Scalar field theory

In this section, we will utilize $1 + 1$ dimensional scalar field theory, which has served as a testbed for addressing signal-to-noise problems [1, 10, 11].

On $L_0 \times L_1$ lattice, the lattice action is written as

$$S = \sum_{x,\mu} \left(\frac{(\phi_{x+\mu} - \phi_x)^2}{2} + \frac{m^2}{2} \phi_x^2 + \frac{\lambda}{4!} \phi_x^4 \right), \tag{11}$$

where $\mu = 0, 1$ denotes the direction and and $x = (x_0, x_1)$. In this work, both small and large coupling cases are considered, with $\lambda = 0.5, 24.0$, respectively. The observable considered here is the two-point correlator at momentum $p = 0$:

$$O(t) = \frac{1}{L_0} \sum_{y_0} \left[\left(\frac{1}{L_1} \sum_{x_1} \phi_{y_0+t, x_1} \right) \left(\frac{1}{L_1} \sum_{y_1} \phi_{y_0, y_1} \right) \right]. \tag{12}$$

Fig. 1 illustrates the training history of control variates on a 20×20 lattice for the correlator at the midpoint, $t = L_0/2$, where the signal-to-noise ratio is the worst. The results show that with zero hidden layer, corresponding to linear transformations, the control variates method still provides an improvement for the small coupling case, as discussed in [1], while it does not perform well for the large coupling case. However, by adding hidden layers, the improvement is in the order of hundreds.
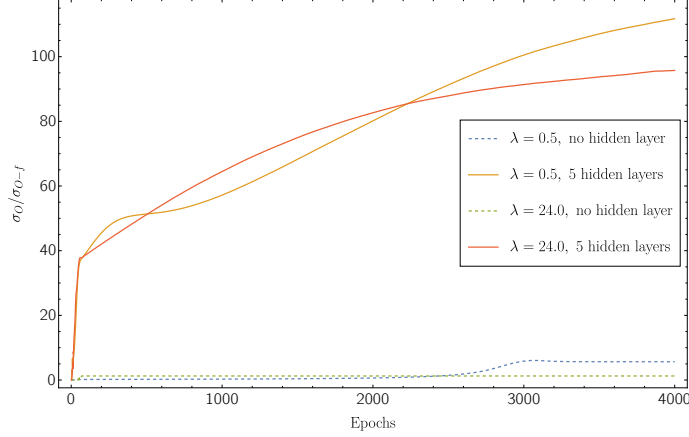
Figure 1: Training histories of the small ($\lambda = 0.5$) and large ($\lambda = 24.0$) couplings on 20×20 lattice with $m^2 = 0.1$ are shown. The figure displays the improvement of standard deviation using control variates relative to the raw observabale ($\sigma_{\mathrm{Raw}}/\sigma_{\mathrm{CV}}$). The dashed line corresponds to the zero hidden layers (linear transformation), while the solid line represents the result with 5 hidden layers, each containing 4 neurons. 10^4 configurations are used for training the network, and 10^3 configurations are employed to estimate the standard deivation.

3.1 Transfer learning

In this subsection, we discuss the improvement in the estimation of mass using a 40×10 lattice with $m^2 = 0.01$ and $\lambda = 0.1$. Note that L_2 regularization is implemented and the loss function is given by:

$$L(w, \mu) = \frac{1}{N} \sum_{i=1}^{N} \left(O(\phi_i) - f(\phi_i) - \mu \right)^2 + \delta \sum_w w^2, \tag{13}$$

where δ denotes the regularization strength.

In the large t limit, the correlator should follow the exponential fit:

$$C(t) = B \left(\mathrm{e}^{-mt} + \mathrm{e}^{m(L_0 - t)} \right). \tag{14}$$

Fig. 2 compares three correlator fits using different estimators: the naive observable, the improved observable with a control variate at $t = L_0/2 = 20$, and the improved estimator at all points using control variates. The fitting results are presented in Table 1. Note that a control variate at $t = 20$ can be applied to other t-values if it helps reduce the variance. Therefore, it is utilized near $t = 20$, as shown in the right panel of Fig. 2.

Note that for the improved estimator at all points using control variates, the transfer learning technique is employed. In general, the transfer learning method [9] leverages learned network

	B	m	χ^2/dof
Raw	0.000112(18)	0.1935(40)	0.55
CV at $t = 20$	0.0001191(51)	0.1920(13)	0.29
Transfer learning	0.0001096(7)	0.1938(2)	0.91

Table 1: The fitted values of correlators in Fig. 2.
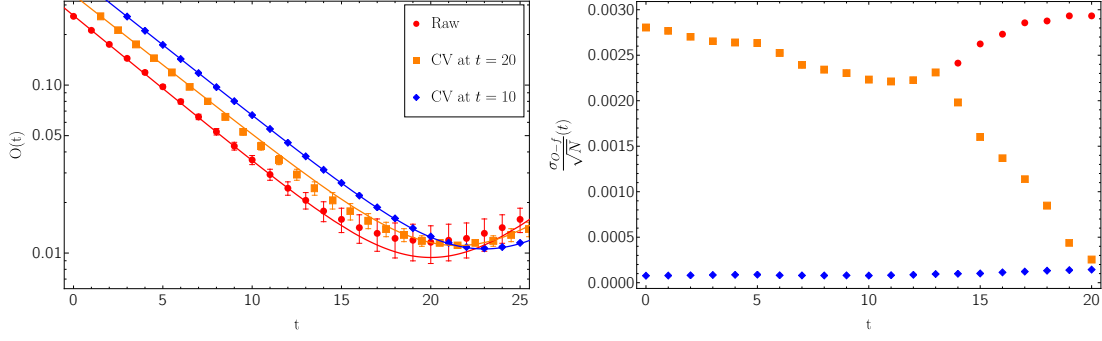
Figure 2: Correlation functions with $m^2 = 0.01$ and $\lambda = 0.1$ on 40×10 lattice are shown. The raw result and the result with control variates are shown. A total of 2×10^3 samples are used, with 10^3 samples reserved for training the control variate, and the entire samples used for estimating the observables. The left plot shows the correlation functions along with their fitting curves, as described in Eq. (14), with the results horizontally shifted for better visualization. The right plot displays the errors presented in the left plot.

parameters to optimize other networks. In our case, for example, if a network is found for the control variate at $t = 10$, the network at $t = 10$ can be used to initialize networks for control variates at $t = 9$ and $t = 11$. This approach significantly reduces the training time. Moreover, the right panel of Fig. 2 shows that the control variate trained with the transfer learning technique at $t = 20$ outperforms the control variates trained from random initialization.

4. Abelian gauge theory

In this section, we consider 2-dimensional $U(1)$ gauge theory. The action is written as

$$S = \beta \sum_x \cos(\theta_x^P), \tag{15}$$

where θ^P denotes the plaquette angle, $\theta_1^L + \theta_2^L - \theta_3^L - \theta_4^L$ in terms of link angles. The observable is the Wilson loop:

$$O(A) = \prod_{x \in A} e^{i\theta_x}. \tag{16}$$

Note that the Wilson loop is not averaged over various shapes or locations with the same size. While this is one of the simplest methods to reduce errors, it is avoided here to preserve the factorization property of a single Wilson loop.

With the open boundary condition and maximal tree gauge fixing, one can rewrite the integration measure of the partition function in terms of plaquettes:

$$Z = \int D\theta^p \exp\left(\beta \sum_x \cos(\theta_x^P)\right), \tag{17}$$

which have been used to study signal-to-noise problems [11] and sign problems [12].

By using the factorization property of the action and the observable, one can construct a control variate as following:

$$f = \prod_{x \in A} e^{i\theta_x^P} - \prod_{x \in A} \left(e^{i\theta_x^P} - f_1(\theta_x^P)\right) \quad \text{where} \quad f_1(\theta^P) = \frac{dg}{d\theta^P} - g\frac{dS_1}{d\theta^P}. \tag{18}$$
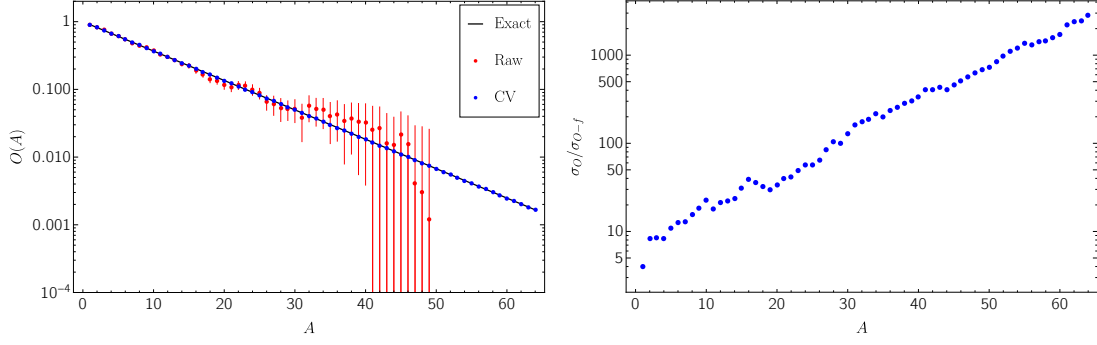
Figure 3: Wilson loops on 8×8 lattice, with the coupling $\beta = 5.555$ are shown. The left panel represents the expectation values of Wilson loops with 1σ error bars, while the right plot illustrates the improvement in standard deviation achieved using control variates, expressed as the ratio $(\sigma_{\text{Raw}}/\sigma_{\text{CV}})$.

Here, $S_1(\theta^P) = \beta(1 - \cos \theta^P)$ is the action for a single plaquette. Note that since the observable in Eq. (16) is complex while its average is real, f_1 must generally be complex. Consequently, g is parametrized by two networks, g_r and g_i, such that $g(\theta^P) = g_r(\theta^P) + i g_i(\theta^P)$.

Fig. 3 illustrates the error reduction achieved using neural control variates. Since two dimensional gauge theories are exactly solvable, the exact expectation values, $(I_1(\beta)/I_0(\beta))^A$, are plotted for comparison. For each area, the two neural networks for g have 4 hidden layers with 16 neurons each. To ensure that f respects the periodic boundary condition to satisfy the Stein's identity, Eq. (2), the input of g is is constructed from trigonometric functions of the plaquette angles θ_P. Fig. 3 demonstrates that while the raw observable suffers from the signal-to-noise problem as the area increases, this issue is mitigated by the control variates, resulting in exponential error reduction, as shown in the right panel.

5. Perfect control variates and constructions of control variates

One of the key properties of the control variates method is its potential to completely eliminate errors by identifying the perfect control variate f_P:

$$f_P = O - \langle O \rangle. \tag{19}$$

Therefore, to evaluate the effectiveness of control variates in reducing variance, it is essential to determine whether the construction includes the perfect control variate or to assess how closely the constructed control variate approximates the ideal one.

There are multiple ways to construct control variates using the Stein's method. However, given that most theories respect translational symmetry, two general constructions can be considered. The first construction is

$$f(\phi) = \sum_i \left(\frac{\partial g}{\partial \phi_i} - g \frac{\partial S}{\partial \phi_i} \right) \quad \text{where } g : \mathbb{R}^V \to \mathbb{R}, \tag{20}$$

Note that the scalar function g must be translationally invariant to impose translational symmetry on the control variate f. Another construction is using a vector-valued function g, as shown in

Eq. (5). Note that g must be translationally covariant for the control variate f to be translationally invariant, as discussed in Section 2.1.

Ref. [3] discussed that Eq. (5), the construction of control variates with a vector-valued function g, can generate any possible control variates through topological arguments. However, the topological argument does not address whether Eq. (20), the construction with a scalar function, can generate every possible control variate. If this is the case, it would be easier to find better control variates, as it would allow for a more restricted search space.

Unfortunately, the example in Section (4) demonstrates that Eq. (20) might not generally include the perfect control variate. Consider the 2-area Wilson loop, where its plaquette angles are parametrized by x and y. In this case, finding the perfect control variate corresponds to solving the following differential equation:

$$\frac{\partial g}{\partial x} + \frac{\partial g}{\partial y} - \beta \sin(x)g - \beta \sin(y)g = \exp(i(x+y)) - \left(\frac{I_1(\beta)}{I_0(\beta)}\right)^2. \tag{21}$$

One can easily find its general solution by changing variables to $x + y$ and $x - y$:

$$g(x, y) = e^{-2\beta \cos\left(\frac{x+y}{2}\right)\cos\left(\frac{x-y}{2}\right)} \left(\int^{\frac{x+y}{2}} du\, e^{2\beta \cos(u)\cos\left(\frac{x-y}{2}\right)} \left(e^{2iu} - \left(\frac{I_1(\beta)}{I_0(\beta)}\right)^2 \right) + C\left(\frac{x-y}{2}\right) \right), \tag{22}$$

where the function C depends on the initial or boundary condition. It is easy to verify that this general solution does not have a periodic solution.

6. Conclusion and outlook

In this work, we showed that the control variates method offers a promising way to mitigate signal-to-noise problems and it can be naturally combined with neural networks. We applied this method to $1 + 1$ dimensional scalar field theory and two dimensional $U(1)$ gauge theory to illustrate its potential for reducing signal-to-noise problems. Symmetries were imposed on the control variates to improve neural network performance.

In Section 5, we discussed that the control variates derived from vector-valued functions are universal, whereas those from scalar-valued functions are not. However, finding effective control variates through training remains challenging, similar to the way solutions to differential equations are proven to exist, but their construction remains a separate challenge. Using the Wilson loop example (Eq. (18)), we showed that good control variates can be identified by leveraging the properties of the theory. Future studies should explore other constructions, while considering specific theories and observables for better training. Lastly, the application of the control variates method to more complex theories remains an important direction, and we are currently working on extending this approach to three-dimensional gauge theories.

Acknowledgments

References

[1] T. Bhattacharya, S. Lawrence, and J.-S. Yoo, *Control variates for lattice field theory*, Phys. Rev. D **109**, (2024) 3, L031505 [2307.14950].

[2] C. Stein, *A bound for the error in the normal approximation to the distribution of a sum of dependent random variables*, Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability (Univ. California, Berkeley, Calif., 1970/1971), Vol. II: Probability theory, 583–602.

[3] S. Lawrence, *Schwinger-Dyson control variates for lattice fermions*, 2404.10707.

[4] P. F. Bedaque, and H. Oh., *Leveraging neural control variates for enhanced precision in lattice field theory*, Phys. Rev. D **109**, (2024) 094519 [2312.08228].

[5] K. Hornik, M. Stinchcombe, and H. White, *Multilayer feedforward networks are universal approximators*, Neural Networks **2**, 359 (1989).

[6] G. Cybenko, *Approximation by superpositions of a sigmoidal function*, Math. Control Signal Systems **2**, 303–314 (1989).

[7] R. Wan, M. Zhong, H. Xiong, and Z. Zhu, *Neural control variates for Monte Carlo variance reduction*, In: Brefeld, U., Fromont, E., Hotho, A., Knobbe, A., Maathuis, M., Robardet, C. (eds) Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2019. Lecture Notes in Computer Science, **11907**. Springer, Cham.

[8] D. Kingma, and J. Ba, *Adam: A Method for Stochastic Optimization*, in International Conference on Learning Representations (ICLR) (San Diega, CA, USA, 2015).

[9] S. Bozinovski, and A. Fulgosi, *The influence of pattern similarity and transfer learning on the base perceptron training (original in Croatian)*, Proceedings of Symposium Informatica 3-121-5, Bled (1976).

[10] S. Lawrence, H. Oh., and Y. Yamauchi, *Lattice scalar field theory at complex coupling*, Phys. Rev. D **106**, (2022) 114503 [2205.12303].

[11] W. Detmold, G. Kanwar, M. L. Wagman, and N. C. Warrington, *Path integral contour deformations for noisy observables*, Phys. Rev. D **102**, (2020) 014514 [2003.05914].

[12] S. Lawrence, and Y. Yamauchi, *Lattice scalar field theory at complex coupling*, Phys. Rev. D **110**, (2024) 014508 [2311.13002].