Asymptotically Unbiased Generative Neural Sampling

Kim Nicoli
Machine Learning Group, Technische Universität Berlin, 10587 Berlin, Germany

Shinichi Nakajima
Machine Learning Group, Technische Universität Berlin, 10587 Berlin, Germany
Berlin Big Data Center, 10587 Berlin, Germany and
RIKEN Center for AIP, 103-0027, Tokyo, Japan

Nils Strodthoff
Fraunhofer Heinrich Hertz Institute, 10587 Berlin, Germany

Wojciech Samek*
Fraunhofer Heinrich Hertz Institute, 10587 Berlin, Germany
Berlin Big Data Center, 10587 Berlin, Germany and
Berliner Zentrum für Maschinelles Lernen, 10587 Berlin, Germany

Klaus-Robert Müller†
Machine Learning Group, Technische Universität Berlin, 10587 Berlin, Germany
Berlin Big Data Center, 10587 Berlin, Germany
Department of Brain and Cognitive Engineering, Korea University,
Anam-dong, Seongbuk-gu, Seoul 136-713, South Korea
Max-Planck-Institut für Informatik, Saarbrücken, Germany and
Berliner Zentrum für Maschinelles Lernen, 10587 Berlin, Germany

Pan Kessel
Machine Learning Group, Technische Universität Berlin, 10587 Berlin, Germany and
Berlin Big Data Center, 10587 Berlin, Germany
(Dated: October 29, 2019)

We propose a general framework for the estimation of observables with generative neural samplers focusing on modern deep generative neural networks that provide an exact sampling probability. In this framework, we present asymptotically unbiased estimators for generic observables, including those that explicitly depend on the partition function such as free energy or entropy, and derive corresponding variance estimators. We demonstrate their practical applicability by numerical experiments for the 2d Ising model which highlight the superiority over existing methods. Our approach greatly enhances the applicability of generative neural samplers to real-world physical systems.

PACS numbers: 05.10.Ln, 05.20.y

I. INTRODUCTION

Monte-Carlo methods are the workhorses of statistical physics and lattice field theories providing insights in strongly correlated systems from first principles [1, 2]. In spite of their overall success, these approaches come with a number of downsides: Monte-Carlo methods potentially get trapped in local minima that prevent them from exploring the full configuration space [3]. Furthermore, they can suffer from large autocorrelation times — in particular close to criticality thus making them very costly in certain regions of the parameter space. In these regions, observables at physical parameter values can often only be extrapolated from simulations at unphysical parameter values. Last but not least, observables that explicitly depend on the partition function, such as free energy and entropy, can only be evaluated up to an overall constant by "chaining" the results of a considerable number of Monte-Carlo chains [2, 4, 5].

In machine learning, generative neural samplers (GNSs) have shown their remarkable performance in generating realistic samples, capturing complicated probability distributions of real-world data such as images, speech, and text documents, see [6] for an overview. This has inspired the application of GNSs in the context of theoretical physics [7–22].

In this work, we focus on a particularly promising subclass of GNSs. Namely, we will consider deep neural networks q that allow to sample configurations $s \sim q$ from the model and also provide the exact probability $q(s)$ of the sample s. A notable example for this type of GNS are Variational Autoregressive Networks (VANs)[20], which sample from a PixelCNN [23] to estimate observables.

* wojciech.samek@hhi.fraunhofer.de
† klaus-robert.mueller@tu-berlin.de

The main advantage of this class of GNSs is that they can be trained without resorting to Monte-Carlo configurations by minimizing the Kullback–Leibler divergence between the model q and a target (Boltzmann) distribution p. As a result, they represent a truly complementary approach to existing Monte-Carlo methods.

Observables are often estimated by directly sampling from the GNS and then taking the sample mean. However, as we will discuss in detail, this approach suffers from a mismatch of the sampling distribution q and the target distribution p. This mismatch is unavoidable since it cannot be expected that the GNS fully captures the underlying physics. This leads to uncontrolled estimates as both the magnitude and the direction of this bias is in general unknown and scales unfavorably with the system size [16].

In this work, we propose a general framework to avoid this serious problem. Our method applies to any GNS with exact sampling probability. Specifically, we will show that it is possible to define asymptotically unbiased estimators for observables along with their corresponding variance estimators. Notably, our method also allows to directly estimate observables that explicitly depend on the partition function, e.g. entropy and free energy. Our proposal therefore greatly enhances the applicability of GNSs to real-world systems.

The paper is organized as follows: In Section II, we will discuss the proposed asymptotically unbiased estimators for observables along with corresponding variance estimators. We illustrate the practical applicability of our approach for the two-dimensional Ising model in Section III, discuss the applicabilty to other GNSs in Section IV and conclude in Section V. Technical details are presented in several appendices.

II. ASYMPTOTICALLY UNBIASED ESTIMATORS

A. Generative Neural Samplers with Exact Probability (GNSEP)

We will use a particular subclass of GNSs to model the variational distribution q as they can provide the exact probability $q(s)$ of configurations s and also allow sampling from this distribution $s \sim q$. We will henceforth refer to this subclass as generative neural samplers with exact probability (GNSEP). Using these two properties, one can then minimize the inverse Kullback–Leibler divergence between the Boltzmann distribution $p(s) = 1/Z \exp(-\beta H(s))$ and the variational distribution q without relying on Monte-Carlo configurations for training,

$$\begin{aligned} \mathrm{KL}(q|p) &= \sum_s q(s) \ln\left(\frac{q(s)}{p(s)}\right) \\ &= \sum_s q(s)(\ln(q(s)) + \beta H(s)) + \ln Z\,. \end{aligned} \tag{1}$$

This objective can straightforwardly be optimized using gradient decent since the last summand is an irrelevant constant shift. After the optimization is completed, observables (expectation values of an operator $\mathcal{O}$ with respect to the Boltzmann distribution p) are then conventionally estimated by the sample mean

$$\langle \mathcal{O}(s) \rangle_p \approx \frac{1}{N} \sum_{i=1}^{N} \mathcal{O}(s_i) \tag{2}$$

using the neural sampler $s_i \sim q$.

Various architectures for generative neural samplers are available. Here, we will briefly review the two most popular ones:

a. Normalizing Flows (NFs): Samples from a prior distribution $q_0(z)$, such as a standard normal, are processed by an invertible neural network $f(z)$. The probability of a sample $s = f(z)$ is then given by

$$q(s) = q_0(f^{-1}(s)) \left| \det\left(\frac{\partial f}{\partial z}\right) \right|^{-1} .$$

The architecture of f is chosen such that the inverse and its Jacobian can easily be computed. Notable examples of normalizing flows include NICE[24], RealNVP[25] and GLOW[26]. First physics applications of this framework have been presented in [19] in the context of lattice field theory.

b. Autoregressive Models (AMs): In this case, an ordering $s_1, \ldots, s_N$ of the spins is chosen and the conditional distribution $q(s_i|s_{i-1} \ldots s_1)$ is modeled by a neural network. The joint probability $q(s)$ is then obtained by multiplying the conditionals

$$q(s) = \prod_{i=1}^{N} q(s_i|s_{i-1} \ldots s_1) \tag{3}$$

and one can draw samples from q by autoregressive sampling from the conditionals. State-of-the-art architectures often use convolutional neural networks (with masked filters to ensure that the conditionals only depend on the previous spins in the ordering). Such convolutional architectures were first proposed in the context of image generation with PixelCNN [23, 27] as most prominent example. In [20] these methods were first used for statistical physics applications.

A major drawback of using generative neural samplers is that their estimates are A) (often substantially) biased and B) do not come with reliable error estimates, see Figure 1. Both properties are obviously highly undesirable for physics applications. The main reason for this is that the mean (2) is taken over samples drawn from the sampler q to estimate expectation values with respect to the Boltzmann distribution p. However, it cannot be expected that the sampler q perfectly reproduces the target distribution p. This discrepancy will therefore necessarily result in a systematic error which is often substantial. Furthermore, in all the cases that we are aware of, this error cannot be reliably estimated.

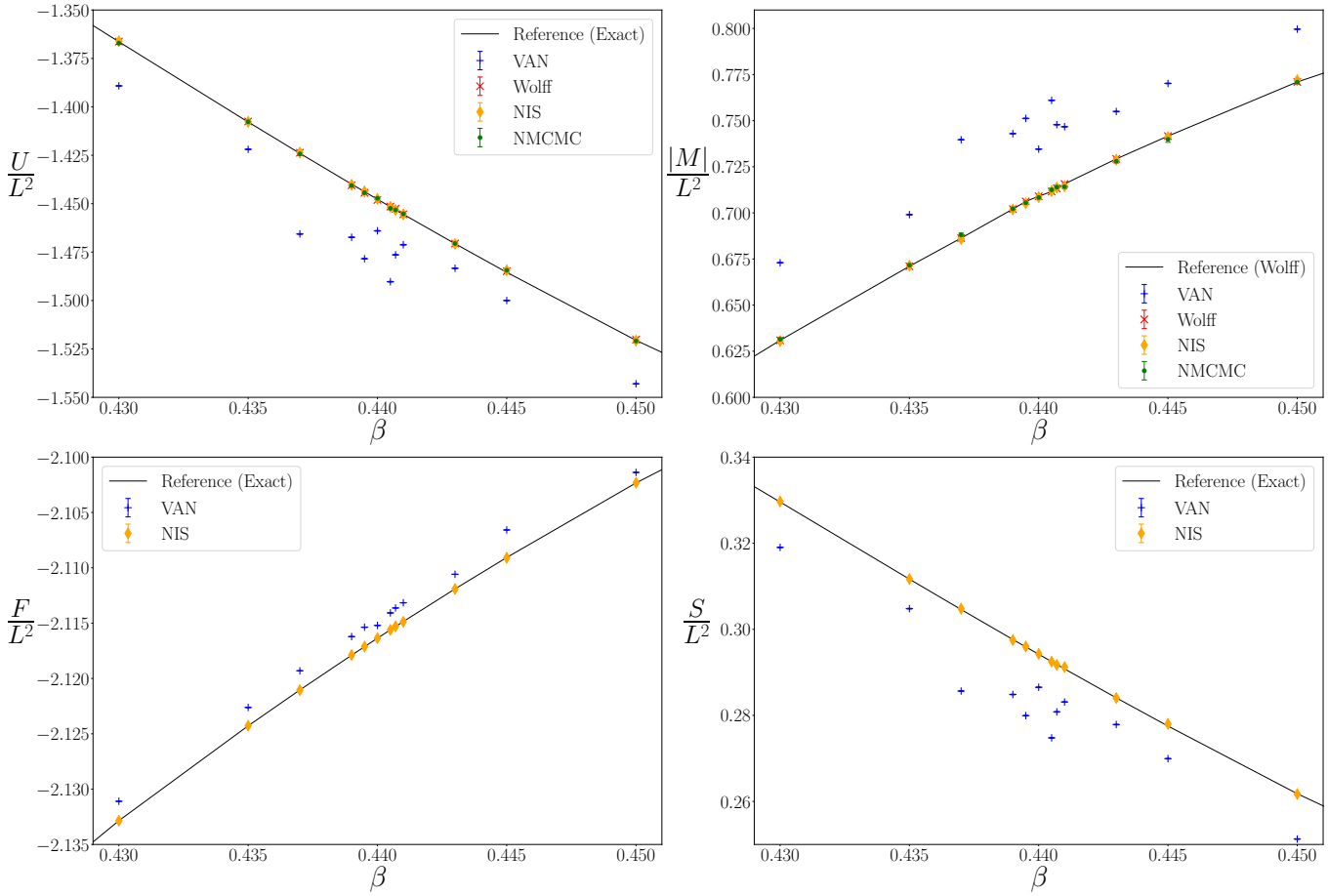

FIG. 1: Estimates for various observables around β_c. Our proposed methods agree with the reference values provided by the exact analytical solutions as well as the Wolff algorithm. VAN deviates significantly.

In order to avoid this serious problem, we propose to use either importance sampling or Markov chain Monte Carlo (MCMC) rejection sampling to obtain asymptotically unbiased estimators. We also derive expressions for the variances of our estimators.

B. Sampling Methods

Here we propose two novel estimators that are asymptotically unbiased and are shown to alleviate the serious issues A) and B) mentioned in the previous section.
Neural MCMC (NMCMC) uses the sampler q as the proposal distribution $p_0(s|s')$ for a Markov Chain. Samples $s \sim p_0(s|s')$ are then accepted with probability

$$\min\left(1, \frac{p_0(s|s')\,p(s')}{p_0(s'|s)\,p(s)}\right) = \min\left(1, \frac{q(s)\,\exp(-\beta H(s'))}{q(s')\,\exp(-\beta H(s))}\right) . \quad (4)$$

We note that the proposal configurations do not depend on the previous elements in the chain. This has two important consequences: Firstly, they can efficiently be sampled in parallel. Secondly, the estimates will typically have very small autocorrelation.

Neural Importance Sampling (NIS) provides an estimator by

$$\langle \mathcal{O}(s) \rangle_p \approx \sum_i w_i\, \mathcal{O}(s_i) \qquad \text{with} \qquad s_i \sim q\,, \quad (5)$$

where $w_i = \frac{\hat{w}_i}{\sum_i \hat{w}_i}$ for $\hat{w}_i = \frac{e^{-\beta H(s_i)}}{q(s_i)}$ is the importance weight. It is important to stress that we can obtain the configurations s_i by iid sampling from q. This is in stark contrast to related reweighting techniques in the context of MCMC sampling [1].

We assume that the output probabilities of the neural sampler q are bounded within $[\epsilon, 1-\epsilon]$ for small $\epsilon > 0$. In practice, this can easily be ensured by rescaling and shifting the output probability of the model as explained in Appendix B.

It then follows from standard textbook arguments that these two sampling methods provide asymptotically unbiased estimators. For convenience, we briefly recall these arguments in Appendix B.

We note that our asymptotic unbiased sampling methods have the interesting positive side effect that they allow for *transfer across parameter space*, a property they share with conventional MCMC approaches [28]. For example, we can use a neural sampler trained at inverse

temperature β' to estimate physical observable at a different target temperature $\beta \neq \beta'$, as shown later in Section III. In some cases, this can result in a significant reduction of runtime, as we will demonstrate in Section III.

C. Asymptotically Unbiased Estimators

For operators $\mathcal{O}(s)$ which do not explicitly depend on the partition function, such as internal energy $\mathcal{O}_U(s) = H(s)$ or absolute magnetization $\mathcal{O}_{|M|}(s) = \sum_i |s_i|$, both NIS and NMCMC provide asymptotically unbiased estimators as explained in the last section.

However, generative neural samplers are often also used for operators $\mathcal{O}(s, Z)$ explicitly involving the partition function Z. Examples for such quantities include

$$\mathcal{O}_F(s, Z) = -\frac{1}{\beta}\ln(Z)\,, \tag{6}$$

$$\mathcal{O}_S(s, Z) = \beta\, H(s) + \ln Z\,, \tag{7}$$

which can be used to estimate the free energy $F = \frac{-1}{\beta}\ln(Z) = \mathbb{E}_p[\mathcal{O}_F]$ and the entropy $S = -\mathbb{E}_p[\ln p] = \mathbb{E}_p[\mathcal{O}_S]$ respectively. Since the Kullback-Leibler divergence is greater or equal to zero, it follows from the optimization objective (1) that

$$F_q = \frac{1}{\beta}\sum_s q(s)(\ln(q(s)) + \beta H(s)) \geq -\frac{1}{\beta}\ln(Z) = F\,. \tag{8}$$

Therefore, the variational free energy F_q provides an upper bound on the free energy F and is thus often used as its estimate. Similarly, one frequently estimates the entropy $S = -\mathbb{E}_p(\ln p)$ by simply using the variational distribution q instead of p. Both estimators however typically come with substantial biases which are hard to quantify. This effect gets particularly pronounced close to the critical temperature.

Crucially, neural importance sampling also provides asymptotically unbiased estimators for $\mathbb{E}_p[\mathcal{O}(s, Z)]$ by

$$\hat{\mathcal{O}}_N = \frac{\frac{1}{N}\sum_{i=1}^N \mathcal{O}(s_i, \hat{Z}_N)\,\hat{w}(s_i)}{\hat{Z}_N} \quad \text{with} \quad s_i \sim q\,, \tag{9}$$

where the partition function Z is estimated by

$$\hat{Z}_N = \frac{1}{N}\sum_{i=1}^N \hat{w}_i\,. \tag{10}$$

In the next section, we will derive the variances of these estimators. Using these results, the errors of such observables can systematically be assessed.

D. Variance Estimators

In the following, we focus on observables of the form

$$\mathcal{O}(s, Z) = g(s) + h(Z)\,, \tag{11}$$

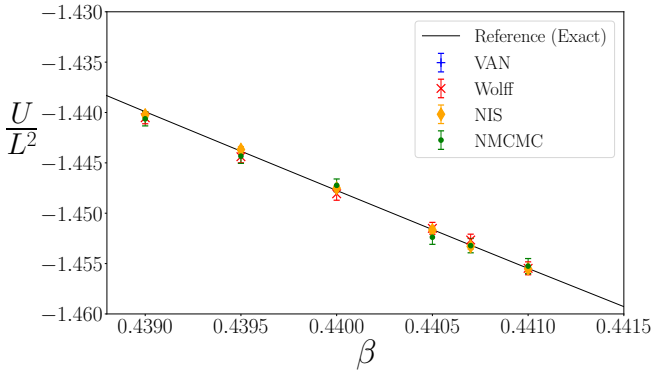

FIG. 2: Zoom around critical $\beta_c \approx 0.4407$ showing the internal energy per lattice site from Fig. 1 (e.g. 16×16 lattice). We took internal energy as a reference example, same considerations hold for the other observables. The estimates from our proposed method match with the exact reference. VAN is not shown because it is out of range as one can see from Figure 1.

that include the estimators for internal energy, magnetization but most notably also for free energy (6) and entropy (7). As just mentioned, expectation values of these operators can be estimated using (9).

Let us assume that h is differentiable at the true value of the partition function Z. Then, as shown in Appendix C, the variance of the estimator for large N is given by

$$\sigma^2_{\hat{\mathcal{O}}_N} = \frac{\boldsymbol{\psi}^T \mathbb{E}_q[\boldsymbol{\phi}\boldsymbol{\phi}^T]\boldsymbol{\psi}}{N} + o_P(N^{-1})\,, \tag{12}$$

where

$$\boldsymbol{\phi} = \begin{pmatrix} g\hat{w} - \mathbb{E}_p[g]Z \\ \hat{w} - Z \end{pmatrix}, \quad \boldsymbol{\psi} = \begin{pmatrix} 1/Z \\ -\mathbb{E}_p[g]/Z + h'(Z) \end{pmatrix}. \tag{13}$$

Note that $\mathbb{E}_p[g]$ can be estimated by

$$\frac{\frac{1}{N}\sum_{i=1}^N g(s_i)\hat{w}(s_i)}{\hat{Z}_N} \tag{14}$$

and Z can be estimated by (10), respectively.

For operators with $h \equiv 0$, it is well-known [29] that Eq. (12) reduces to

$$\sigma^2_{\hat{\mathcal{O}}_N} = \frac{\mathrm{Var}_p(g)}{N_{\text{eff}}} + o_P(N^{-1})\,, \tag{15}$$

where we have defined the effective sampling size

$$N_{\text{eff}} = \frac{N}{\mathbb{E}_q\left[w^2\right]}\,. \tag{16}$$

Note that the effective sampling size does not depend on the particular form of g. It is however important to stress that for observables with $h \neq 0$, the error cannot

be estimated in terms of effective sampling size but one has to use (12). While this expression is more lengthy, it can be easily estimated. Therefore, neural importance sampling allows us to reliably estimate the variances of physical observables — in particular observables with explicit dependence on the partition function. This is in stark contrast to the usual GNS approach.

It is also worth stressing that MCMC sampling does not allow to directly estimate those observables which explicitly involve the partition function. For completeness, we also note that a similar well-known effective sampling size can be defined for MCMC

$$N_{\text{eff}} = \frac{N}{2\,\tau_{int,\mathcal{O}}}\,, \tag{17}$$

where $\tau_{int,\mathcal{O}}$ is the integrated auto-correlation time of the operator $\mathcal{O}$, see [1, 30] for more details.

III. EXPERIMENTS

We will now demonstrate the effectiveness of our method on the example of the two-dimensional Ising model with vanishing external magnetic field. This model has an exact solution and therefore provides us with a ground truth to compare to. The Hamiltonian of the Ising model is given by

$$H(s) = -J \sum_{\langle i,j \rangle} s_i\, s_j\,, \tag{18}$$

where J is the coupling constant and the sum runs over all neighbouring pairs of lattice sites. The corresponding Boltzmann distribution is then given by

$$p(s) = \frac{1}{Z} \exp(-\beta H(s))\,, \tag{19}$$

with partition function $Z = \sum_s \exp(-\beta H(s))$. For simplicity, we will absorb the coupling constant J in β in the following. Here, we will only consider the ferromagnetic case for which $J > 0$ and the model undergoes a second-order phase transition at $\beta_c \approx 0.4407$ in the infinite volume limit.

In addition to the exact solution by Onsager for the infinite volume case [31], there also exists an analytical solution for finite lattices [32], which we review in Appendix A and use for reference values. An exact partition function for the case of vanishing magnetic field is not enough to derive expressions for some observables, such as magnetization. For these observables, we obtain reference values by using the Wolff MCMC clustering algorithm [33].

A. Unbiased Estimators for the Ising Model

For discrete sampling spaces, autoregressive algorithms are the preferred choice as normalizing flows are designed for continuous ones [34]. It is nonetheless important to stress that our proposed method applies irrespective of the particular choice for the sampler.

We use the standard VAN architecture for the GNS. For training, we closely follow the procedure described in the original publication [20]. More detailed information about hyperparameter choices can be found in Appendix D. We use VANs, trained for a 16×16 lattice at various temperatures around the critical point, to estimate a number of observables. The errors for neural importance sampling are determined as explained in Section II D. For Wolff and Neural MCMC, we estimate the autocorrelation time as described in [30].

Figure 1 summarizes the main results of our experiments in terms of estimates for internal energy, absolute magnetization, entropy and free energy around the critical regime. NMCMC and NIS agree with the reference values while VAN deviates significantly. We note that this effect is also present for observables with explicit dependence on the partition function, i.e. for entropy and free energy.

All estimates in Figure 1 deviate from the reference value in the same direction. Whereas this is expected for the free energy (for which the true value is a lower bound) also for the other observables the trained GNSs seem to favor a certain direction of approaching the true value. However, as we show in Appendix E, this trend holds only on average and is not a systematic effect.

In Figure 3, we track the evolution of the estimates for the four observables under consideration during training. This figure clearly demonstrates that our proposed method leads to accurate predictions even at earlier stages of the training process. This is particularly important because the overall runtime for GNS estimates is heavily dominated by the training.

Table I summarizes results for 24×24 lattice. For this larger lattice, the systematic error of VAN is even more pronounced and the estimated values do not even qualitatively agree with the reference values. Our modified sampling techniques, on the other hand, lead to fully compatible results.

Lastly, our proposed methods allow for transfer across parameter space, as explained in Section II B. In Figure 6, we summarize a few transfer runs. Models are trained at decreasing β value and are used to predict the energy at β_c. All predicted values agree with the reference within error bars. As the difference in temperature between model and target increases, the variance grows as well — as was to be expected. In practice, this limits the difference between model and target inverse temperature. Nevertheless, we can use models trained at a single β value to predict other values in a non-trivial neighbourhood of the model β. This allows to more finely probe parameter space at only minimal additional computational costs.

Sampler (24x24)	U/L^2	$\lvert M\rvert/L^2$	S/L^2	F/L^2
VAN	-1.50583 (0.00010)	0.78293 (0.00008)	0.26505 (0.00004)	-2.107250 (0.000001)
NIS	**-1.43472** (0.02154)	**0.672** (0.03)	**0.29885** (0.00717)	**-2.11284** (0.00075)
NMCMC	**-1.44950** (0.00673)	**0.680** (0.04)	-	-
Reference	-1.44025	0.6777 (0.0006)	0.29611	-2.11215

TABLE I: Comparison of VAN, NMCMC and NIS on a 24×24 lattice trained at β_c. Entropy and Free Energy cannot be directly estimated using Monte Carlo approaches. Bold numbers denote estimates which are compatible with ground truth within one standard deviation. Standard deviations are in parentheses.

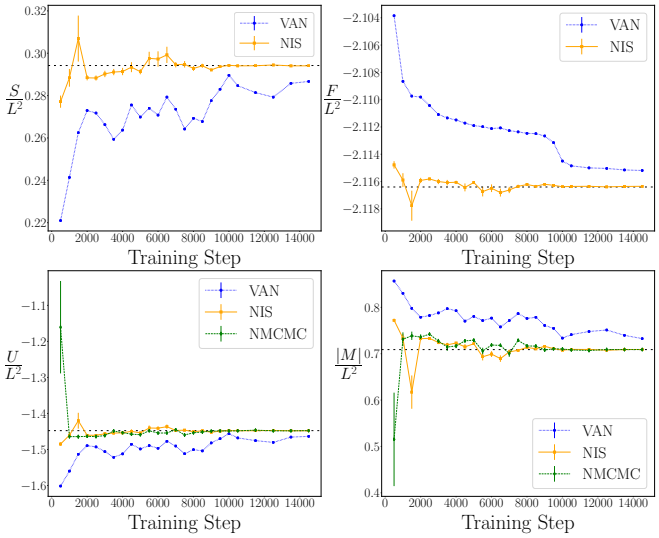

FIG. 3: Estimation of observables during training for a single run on a 16×16 lattice. The modified sampling procedure leads to accurate predictions at significantly earlier stages of training.

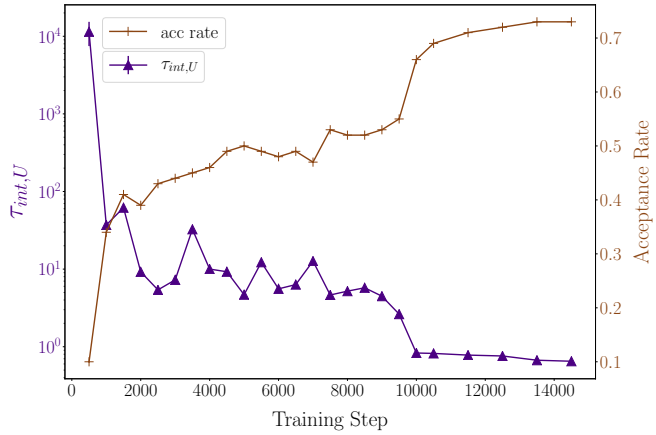

FIG. 4: Evolution of the acceptance rate (right) and the integrated autocorrelation time of the internal energy $\tau_{int,U}$ (left) during training. NMCMC runs were preformed on a 16×16 lattice at β_c.

Observable	Metropolis	NMCMC
$\tau_{int,U}$	4.0415	0.8317
$\tau_{int,\lvert M\rvert}$	7.8510	1.3331

TABLE II: Neural MCMC instead of Metropolis leads to a significant reduction of integrated autocorrelation times τ_{int} for a 16×16 lattice at β_c. The neural sampler was trained over ten thousands steps and the acceptance rate was 69 percent.

B. Neural MCMC

NMCMC obtains a proposal configuration by independent and identically distributed sampling from the sampler q. This can result in a significantly reduced integrated autocorrelation time $\tau_{int,\mathcal{O}}$ for the observables $\langle\mathcal{O}\rangle$. For this reduction, it is not required to perfectly train the sampler. It is however required that the sampler is sufficiently well-trained such that the proposal configuration is accepted with relatively high probability, as illustrated in Figure 4. Table II demonstrates a significant reduction in integrated autocorrelation τ_{int}, as defined in (17), for two observables at β_c on a 16×16 lattice.

In NMCMC, the proposal configuration $s \sim p_0(s|s') = q(s)$ is independent of the previous configuration s' in the chain. This is in stark contrast to the Metropolis algorithm for which the proposal configuration is obtained by a local update of the previous configuration. As a result, NMCMC is less likely to stay confined in (the neighbourhood of) an energy minimum of the configuration space. This is demonstrated in Figure 5 which shows the magnetization histograms for Metropolis and Neural MCMC.

Since the Ising model has a discrete $\mathbb{Z}_2$-symmetry, we expect a two-moded distribution. In constrast to the Metropolis algorithm, NMCMC indeed shows such a behaviour.

IV. APPLICABILITY TO OTHER SAMPLERS

We note that our approach can in parts be applied to other generative models see Table III which summarizes the applicability of neural MCMC (NMCMC) sampling and neural importance sampling (NIS). Namely, when the employed GNS provides an unnormalized sampling probability, i.e., the exact probability multiplied by a constant, then NMCMC and NIS can again correct the

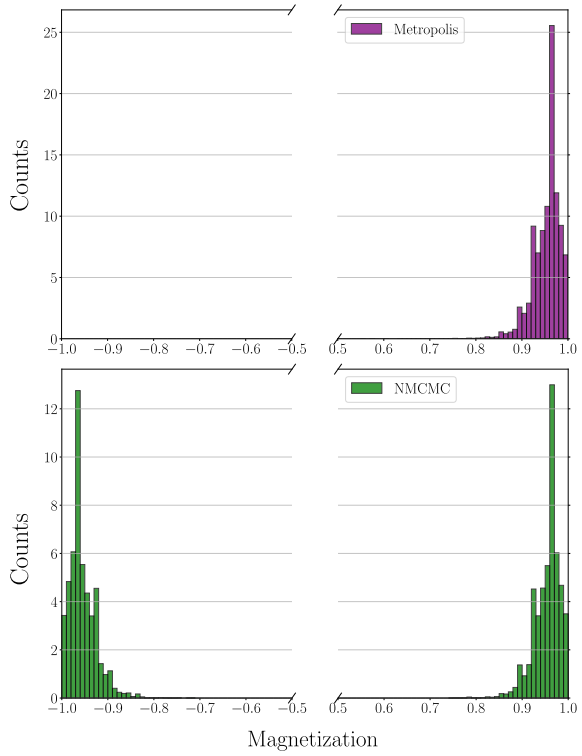

FIG. 5: Histogram for the magnetization of the system at $\beta = 0.55$. While the Metropolis algorithm is only able to capture one of the two modes of the distribution, NMCMC is able to cover both.

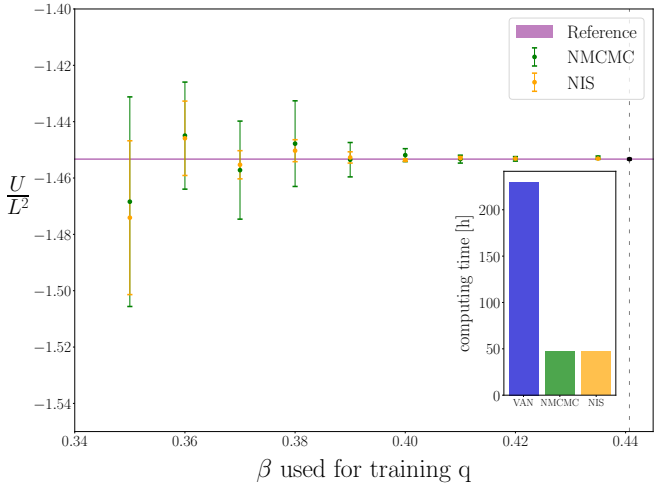

FIG. 6: Samplers q are trained at increasingly lower β values and used to predict the internal energy U/L^2 at the critical coupling β_c. All results agree with the reference values within error bars. The variance of the estimators increase as the difference between model and target temperature gets larger. Transfer runs for NMCMC and NIS allow to only train one model which leads to significant speed up since runtime is dominated by training, as illustrated in the inset.

learned sampler q leading to asymptotically unbiased estimators. However, the applicability is limited to the observables that do *not* explicitly depend on the partition function, i.e., $h \equiv 0$ in (11).

If the employed GNS allows us to approximate the (normalized or unnormalized) sampling probability, one can apply our approach by using the approximate probability for q. The bias can then be reduced if the gap between the target distribution and the sampling distribution is larger than the approximation error to the sampling probability. However, then the estimator may not be asymptotically unbiased.

V. CONCLUSION AND OUTLOOK

In this work, we presented a novel approach for the unbiased estimation of observables with well-defined variance estimates from generative neural samplers that provides the exact sampling probability (GNSEP). Most notably, this includes also observables that explicitly depend on the partition function such as free energy or entropy. The practical applicability of the approach is demonstrated for the two-dimensional Ising model, stressing the importance of unbiased estimators compared to biased estimators from the literature.

In summary, the methods proposed in this paper not only lead to theoretical guarantees but are also of great practical relevance. They are applicable for a large class of generative samplers, easy to implement, and often lead to a significant reduction in runtime. We therefore firmly believe that they will play a crucial role in the promising application of generative models to challenging problems of theoretical physics.

ACKNOWLEDGMENTS

This work was supported by the German Ministry for Education and Research as Berlin Big Data Center (01IS18025A) and Berlin Center for Machine Learning (01IS18037I). This work is also supported by the Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (No. 2017-0-001779) and by the DFG (EXC 2046/1, Project-ID 390685689).

Accessible sampling probability	NMCMC, NIS($h \equiv 0$)	NIS($h \neq 0$)	GNSs
exact, normalized	✓	✓	AM, NF
exact, unnormalized	✓	✗	–
approximate, normalized	(✓)	(✓)	VAE
approximate, unnormalized	(✓)	✗	RBM
none	✗	✗	GAN

TABLE III: Applicability of NMCMC and NIS to various GNSs. h refers to the term explicitly depending on the partition function Z of the observable (11). General Adversarial Networks (GANs) do not provide sampling probabilites and therefore can not be used for our method. Restricted Boltzmann Machines (RBMs) only provide approximate and unnormalized sampling probability and therefore do not lead to asymptotically unbiased estimators using our methods. Because of the lack of normalization, observables with explicit dependence on the partition function cannot be estimated. Variational Autoencoders (VAEs) provide approximate sampling probability. Our method can therefore be applied but does not lead to asymptotic guarantees. The cases of Normalizing Flows (NFs) and Autoregressive Models (AMs) were discussed at length before. The applicability is summarized in the table using the following notation: ✓: the estimator is asymptotically unbiased; (✓): applicable but the estimator is still biased; ✗: not applicable.

Appendix A: Partition function of the finite-size Ising model

In this appendix, we review the exact solution for the partition function of the finite-size Ising model [32]. For an $L \times L$ lattice, the partition function is given by

$$Z = \frac{1}{2}(2\sinh(2\beta))^{L^2/2} \sum_{i=i}^{4} Z_i \,, \tag{A1}$$

where we have used the definitions

$$\begin{aligned} Z_1 &= \prod_{r=0}^{L-1} 2\cosh(\tfrac{1}{2}L\gamma_{2r+1})\,, \quad Z_2 = \prod_{r=0}^{L-1} 2\sinh(\tfrac{1}{2}L\gamma_{2r+1})\,, \\ Z_3 &= \prod_{r=0}^{L-1} 2\cosh(\tfrac{1}{2}L\gamma_{2r})\,, \qquad Z_4 = \prod_{r=0}^{L-1} 2\sinh(\tfrac{1}{2}L\gamma_{2r})\,, \end{aligned} \tag{A2}$$

with the coefficients

$$\begin{aligned} \gamma_0 &= 2\beta + \ln\tanh\beta\,, \\ \gamma_r &= \ln(c_r + \sqrt{c_r^2 - 1}) \qquad \text{for} \qquad r > 0\,, \end{aligned} \tag{A3}$$

and $c_r = \cosh 2\beta \coth 2\beta - \cos(r\pi/L)$. From this expression for the partition function, one can easily obtain analytical expressions for the free energy and entropy.

Appendix B: Proof of asymptotic Unbiasedness

In this section, we will give a review of the relevant arguments establishing that the NIS and NMCMC estimators are asymptotically unbiased.

For reasons that will become obvious soon, it is advantageous to re-interpret the original network output $q' \in [0,1]$ as the probability $q \in [\epsilon, 1-\epsilon]$ by the following mapping:

$$q = \left(q' - \frac{1}{2}\right)(1-2\epsilon) + \frac{1}{2}. \tag{B1}$$

Due to the rescaling discussed above, we can assume that the support of the sampling distribution q contains the support of the target distribution p. This property is ensured since the sampler q takes values in $q \in [\epsilon, 1-\epsilon]$.

1. Neural Importance Sampling

Importance sampling with respect to q, i.e.

$$\begin{aligned} &\mathbb{E}_p[\mathcal{O}(s)] \approx \sum_{i=1}^{N} w_i\, \mathcal{O}(s_i)\,, \text{ where} \\ &s_i \sim q(s)\,,\, w_i = \frac{\hat{w}_i}{\sum_i \hat{w}_i}\,,\, \hat{w}_i = \frac{e^{-\beta H(s_i)}}{q(s_i)}\,, \end{aligned} \tag{B2}$$

is an asymptotically unbiased estimator of the expectation value $\langle \mathcal{O}(s) \rangle$ because

$$\begin{aligned} \mathbb{E}_p[\mathcal{O}(s)] &= \sum_s p(s)\mathcal{O}(s) = \sum_s q(s)\,\frac{p(s)}{q(s)}\,\mathcal{O}(s) \\ &= \frac{1}{Z}\sum_s q(s)\,\underbrace{\frac{\exp(-\beta H(s))}{q(s)}}_{=\hat{w}(s)}\,\mathcal{O}(s) \\ &= \frac{1}{ZN}\sum_{i=1}^{N} \hat{w}(s_i)\mathcal{O}(s_i) + o_P(1)\,, \end{aligned} \tag{B3}$$

where $s_i \sim q$. The partition function Z can be similarly determined

$$\begin{aligned} Z &= \sum_s \exp(-\beta H(s)) \\ &= \sum_s q(s)\frac{\exp(-\beta H(s))}{q(s)} = \frac{1}{N}\sum_{i=1}^{N} \hat{w}(s_i) + o_P(1)\,. \end{aligned} \tag{B4}$$

Combining the previous equations, we obtain

$$\langle \mathcal{O}(s) \rangle_p = \sum_{i=1}^{N} w_i \, \mathcal{O}(s_i) + o_P(1) \quad \text{with} \quad w_i = \frac{\hat{w}_i}{\sum_i \hat{w}_i} \,. \tag{B5}$$

2. Neural MCMC

The sampler q can be used as a trial distribution $p_0(s'|s) = q(s')$ for a Markov-Chain which uses the following acceptance probability in its Metropolis step

$$\begin{aligned} p_a(s'|s) &= \min\left(1, \frac{p_0(s|s')p(s')}{p_0(s'|s)p(s)}\right) \\ &= \min\left(1, \frac{q(s)\,\exp(-\beta H(s'))}{q(s')\,\exp(-\beta H(s))}\right). \end{aligned} \tag{B6}$$

This fulfills the detailed balance condition

$$p_t(s'|s)\exp(-\beta H(s)) = p_t(s|s')\exp(-\beta H(s)) \tag{B7}$$

because the total transition probability is given by $p_t(s'|s) = q(s')\,p_a(s'|s)$ and therefore

$$\begin{aligned} &p_t(s'|s)\exp(-\beta H(s)) \\ &= q(s')\min\left(1, \frac{q(s)\,\exp(-\beta H(s'))}{q(s')\,\exp(-\beta H(s))}\right)\exp(-\beta H(s)) \\ &= \min\left\{q(s')\exp(-\beta H(s)),\, q(s)\exp(-\beta H(s'))\right\} \\ &= p_t(s|s')\exp(-\beta H(s'))\,, \end{aligned} \tag{B8}$$

where we have used the fact that the min operator is symmetric and that all factors are strictly positive. The latter property is ensured by the fact that $q(s) \in [\epsilon, 1-\epsilon]$.

Appendix C: Variance Estimators

As explained in the main text, we estimate observables of the form

$$\mathcal{O} = g(s) + h(Z) \tag{C1}$$

by the samples $s_i \sim q$ with $i = 1 \dots N$ using

$$\hat{\mathcal{O}}_N = \frac{\frac{1}{N}\sum_{i=1}^{N} \mathcal{O}(s_i, \hat{Z}_N)\,\hat{w}(s_i)}{\hat{Z}_N} \,. \tag{C2}$$

By the definition of $\hat{Z}_N$, see (10), this is equivalent to

$$\hat{\mathcal{O}}_N = \frac{\frac{1}{N}\sum_{i=1}^{N} g(s_i)\hat{w}(s_i)}{\hat{Z}_N} + h(\hat{Z}_N). \tag{C3}$$

Let

$$\boldsymbol{\phi}_N = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{\phi}(s_i) \qquad \text{for} \qquad \boldsymbol{\phi}(s) = \begin{pmatrix} g(s)\hat{w}(s) \\ \hat{w}(s) \end{pmatrix}. \tag{C4}$$

Observable	Estimated Std	Sample Std
Entropy	0.00023	0.00025
Free En.	0.00002	0.00002

TABLE IV: Comparison of standard deviation estimated as in Section II D to sample standard deviation over ten runs. Experiments are done at an inverse temperature $\beta = 0.44$.

Then, the central limit theorem implies that

$$\boldsymbol{\phi}_N \xrightarrow{D} \mathcal{N}\left(\boldsymbol{\phi}^*, \frac{1}{N}\boldsymbol{\Sigma}\right), \tag{C5}$$

where

$$\begin{aligned} \boldsymbol{\phi}^* &= \mathbb{E}_q\left[\boldsymbol{\phi}\right] = \begin{pmatrix} \mathbb{E}_p[g]\,Z \\ Z \end{pmatrix}, \\ \boldsymbol{\Sigma} &= \mathbb{E}_q\left[\boldsymbol{\phi}\boldsymbol{\phi}^T\right] - \mathbb{E}_q[\boldsymbol{\phi}]\mathbb{E}_q[\boldsymbol{\phi}^T]. \end{aligned} \tag{C6}$$

Since the estimator (C2) can be written as a smooth function f of $\boldsymbol{\phi}_N$ as

$$f(\boldsymbol{\phi}_N) := \frac{(\boldsymbol{\phi}_N)_1}{(\boldsymbol{\phi}_N)_2} + h\big((\boldsymbol{\phi}_N)_2\big) = \hat{\mathcal{O}}_N, \tag{C7}$$

its variance can be written as

$$\sigma^2_{\hat{\mathcal{O}}_N} = \frac{1}{N}\boldsymbol{\psi}^T\,\boldsymbol{\Sigma}\,\boldsymbol{\psi} + o_P(N^{-1}) \tag{C8}$$

with

$$\begin{aligned} \boldsymbol{\psi} = \nabla f(\boldsymbol{\phi}^*) &= \left.\begin{pmatrix} 1/(\boldsymbol{\phi}_N)_2 \\ -(\boldsymbol{\phi}_N)_1/(\boldsymbol{\phi}_N)_2^2 + h'\big((\boldsymbol{\phi}_N)_2\big) \end{pmatrix}\right|_{\boldsymbol{\phi}^*} \\ &= \begin{pmatrix} 1/Z \\ -\mathbb{E}_p[g]/Z + h'\left(Z\right) \end{pmatrix}. \end{aligned} \tag{C9}$$

In Table IV, we numerically verify that our estimated standard deviation is consistent with the sample standard deviation over ten runs.

Appendix D: Experimental Details

In this appendix, we provide an overview of the setup used for the experiments presented in this manuscript.

1. Model Training

Unless reported differently, all the models were trained for a 16×16 lattice for a total of 10000 steps. The model trained on a 24×24 lattice required 15000 steps until convergence. Our model use the VAN architecture with residual connections (see [20] for details on this architecture). The networks are six convolutional layers deep (with a half-kernel size of three) and has a width size of

64. A batch size of 2000 and a learning rate of 10^{-4} were chosen. No learning rate schedulers were deployed in our training. For each model, we applied *β-annealing* to the *target* β_t using the following annealing schedule

$$\beta = \beta_t(1 - 0.998^{N_{step}}) \tag{D1}$$

where N_{step} is the total number of training steps. We summarize the used setup in Table V. Training a sampler for a 16×16 lattice takes approximately 24 hours of computing time on two Tesla P100 GPUs with 16GB each.

2. Neural Monte Carlo and Neural Important Sampling

In Neural MCMC, we use a chain of 500k steps. Conversely to standard MCMC methods, such as Metropolis, no equilibrium steps are required since we sample from an already trained proposal distribution. In Neural Importance Sampling, batches of 1000 samples are drawn 500 times. Both sampling methods were performed on a Tesla P100 GPU and their runtime is approximately an hour in the case of a 16×16.

Appendix E: Direction of Bias

In this appendix, we demonstrate that the direction of the bias depends on the random initialization of the network. In order to illustrate this fact, we trained five models at $\beta = 0.45$ for a 8×8 lattice using the same hyperparameter setup. We compare the estimate of the energy with an exact reference value of $U/L^2 = 1.54439$. Table VI summarizes the results. Values which overestimate the ground truth are in bold. This shows that the trend of under- or overestimating, observed from Fig 1, holds only on average and is not a systematic effect.

Sampler	Depth	Width	Batch	lr	Steps	β Ann.
PixelCNN	6	64	$2 \cdot 10^3$	10^{-4}	10^4	0.998

TABLE V: Summary of the hyperparameters setup used for training our samplers.

Obs	Model 1	Model 2	Model 3	Model 4	Model 5
U/L^2	**-1.5407**	-1.5461	**-1.5364**	**-1.5438**	**-1.5421**
$\lvert M\rvert/L^2$	**0.8089**	**0.8114**	0.8059	**0.8098**	0.8070
S/L^2	0.258889	**0.260271**	0.257843	**0.262209**	**0.261478**

TABLE VI: Internal energy per lattice site on an 8×8 lattice. Values which overestimate the ground truth $U/L^2 = -1.54439$ are in bold. Same holds for Entropy, ground truth $S/L^2 = 0.25898$. The second row shows absolute magnetization with ground truth $|M|/L^2 = 0.8083$. In this case, estimates which overestimate the ground truth are in bold.

[1] C. Gattringer and C. Lang, *Quantum chromodynamics on the lattice: an introductory presentation*, Vol. 788 (Springer, 2009).
[2] M. Newman and G. Barkema, *Monte carlo methods in statistical physics* (Oxford University Press, 1999).
[3] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, Science **220**, 671 (1983).
[4] C. M. Bishop, *Pattern recognition and machine learning* (springer, 2006).
[5] S. Nakajima, K. Watanabe, and M. Sugiyama, *Variational Bayesian Learning Theory* (Cambridge University Press, 2019).
[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning* (MIT press, 2016).
[7] G. Torlai and R. G. Melko, Phys. Rev. B **94**, 165134 (2016), arXiv:1606.02718 [cond-mat.stat-mech].
[8] A. Morningstar and R. G. Melko, Journal of Machine Learning Research **18**, 163:1 (2017), arXiv:1708.04622 [cond-mat.dis-nn].
[9] Z. Liu, S. P. Rodrigues, and W. Cai, (2017), arXiv:1710.04987 [cond-mat.dis-nn].
[10] L. Huang and L. Wang, Physical Review B **95**, 035105 (2017).
[11] S.-H. Li and L. Wang, Physical review letters **121**, 260601 (2018).
[12] M. Koch-Janusz and Z. Ringel, Nature Physics **14**, 578 (2018).
[13] J. M. Urban and J. M. Pawlowski, (2018), arXiv:1811.03533 [hep-lat].
[14] K. Zhou, G. Endrődi, L.-G. Pang, and H. Stöcker, Physical Review D **100**, 011501 (2019).
[15] M. Mustafa, D. Bard, W. Bhimji, Z. Lukić, R. Al-Rfou, and J. M. Kratochvil, Computational Astrophysics and Cosmology **6**, 1 (2019).
[16] K. Nicoli, P. Kessel, N. Strodthoff, W. Samek, K.-R. Müller, and S. Nakajima, (2019), arXiv:1903.11048 [cond-mat.stat-mech].
[17] H.-Y. Hu, S.-H. Li, L. Wang, and Y.-Z. You, (2019), arXiv:1903.00804 [cond-mat.dis-nn].
[18] L. Yang, Z. Leng, G. Yu, A. Patel, W.-J. Hu, and H. Pu, (2019), arXiv:1905.10730 [cond-mat.str-el].
[19] M. S. Albergo, G. Kanwar, and P. E. Shanahan, Phys. Rev. **D100**, 034515 (2019), arXiv:1904.12072 [hep-lat].

[20] D. Wu, L. Wang, and P. Zhang, Physical review letters **122**, 080602 (2019).
[21] O. Sharir, Y. Levine, N. Wies, G. Carleo, and A. Shashua, (2019), arXiv:1902.04057 [cond-mat.dis-nn].
[22] F. Noé, S. Olsson, J. Köhler, and H. Wu, Science **365** (2019).
[23] A. V. Oord, N. Kalchbrenner, and K. Kavukcuoglu, in *ICML*, Vol. 48 (2016) pp. 1747–1756, arXiv:1601.06759 [cs.CV].
[24] L. Dinh, D. Krueger, and Y. Bengio, in *ICLR Workshop* (2015) arXiv:1410.8516 [cs.LG].
[25] L. Dinh, J. Sohl-Dickstein, and S. Bengio, in *ICLR* (2017) arXiv:1605.08803 [cs.LG].
[26] D. P. Kingma and P. Dhariwal, in *NIPS* (2018) pp. 10215–10224, arXiv:1807.03039 [stat.ML].
[27] T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma, in *ICLR*, Vol. 33 (2017) arXiv:1701.05517 [cs.LG].
[28] M. Newman and G. Barkema, in *Monte carlo methods in statistical physics* (Oxford University Press, New York, USA, 1999) Chap. 8, pp. 211–216.
[29] A. Kong, University of Chicago, Dept. of Statistics, Tech. Rep **348** (1992).
[30] U. Wolff, A. Collaboration, *et al.*, Computer Physics Communications **156**, 143 (2004).
[31] L. Onsager, Physical Review **65**, 117 (1944).
[32] A. E. Ferdinand and M. E. Fisher, Physical Review **185**, 832 (1969).
[33] U. Wolff, Physics Letters B **228**, 379 (1989).
[34] However, [35, 36] present a recent attempt to apply normalizing flows to discrete sampling spaces.
[35] D. Tran, K. Vafa, K. K. Agrawal, L. Dinh, and B. Poole, ICLR Workshop Paper (2019), arXiv:1905.10347 [cs.LG].
[36] E. Hoogeboom, J. W. Peters, R. v. d. Berg, and M. Welling, (2019), arXiv:1905.07376 [cs.LG].