GENERATIVE LEARNING FOR THE PROBLEM OF CRITICAL SLOWING DOWN IN LATTICE GROSS NEVEU MODEL

Ankur Singha[1], Dipankar Chakrabarti[1] and Vipul Arora[2]

[1] *Department of Physics, Indian Institute of Technology Kanpur, Kanpur-208016, India.*

[2] *Department of Electrical Engineering, Indian Institute of Technology Kanpur, Kanpur-208016, India*

(Dated: February 15, 2022)

In lattice field theory, Monte Carlo simulation algorithms get highly affected by critical slowing down in the critical region, where autocorrelation time increases rapidly. Hence the cost of generation of lattice configurations near the critical region increases sharply. In this paper, we use a Conditional Generative Adversarial Network (C-GAN) for sampling lattice configurations. We train the C-GAN on the dataset consisting of Hybrid Monte Carlo (HMC) samples in regions away from the critical region, i.e., in the regions where the HMC simulation cost is not so high. Then we use the trained C-GAN model to generate independent samples in the critical region. We perform both interpolation and extrapolation to the critical region. Thus, the overall computational cost is reduced. We test our approach for Gross-Neveu model in 1+1 dimension. We find that the observable distributions obtained from the proposed C-GAN model match with those obtained from HMC simulations, while circumventing the problem of critical slowing down.

Keywords: lattice, GN model, deep learning, GAN, critical slowing down, phase transition

I. INTRODUCTION

Lattice field theory is the most reliable and well established technique to solve quantum field theories nonperturbatively. In this approach the theory is formulated on a discrete space-time lattice to solve numerically. In Monte Carlo(MC) simulation of lattice field theory the efficiency of simulation depends on the algorithm used. Algorithm like Hybrid Monte Carlo (HMC)[1] works well away from the critical points of the lattice theory but when one approaches the critical region the simulation algorithm suffers severe critical slowing down [2][3]. Near the critical point the autocorrelation time increases dramatically and can become larger than the total simulation run time. Therefore we have no control over the statistical uncertainties of calculated observable on the simulated lattice configurations. As an example, in lattice QCD as we approach the continuum limit $a \to 0$ for a fixed physical volume, the computational cost of HMC scales approximately as a^{-z} with $z > 6$[3]. Several methods in gauge theories has been developed [4][5][6] to improve the MC simulations on lattice. Machine Learning(ML), in the mean time has made tremendous advancements and found application in many branches in physics. ML has been applied extensively in many condense matter and statistical physics problems[7–10]. In [11], supervised learning has been adopted to accelerate the MC simulations for statistical physics problems, a self learning MC has been proposed in [12] to reduce the autocorrelation time specially near the critical region by learning an effective Hamiltonian. In recent times some machine learning approaches[13–20] are used to circumvent the problem of diverging autocorrelation time in lattice filed theory and XY model[21] as well. Machine Learning(ML) has been also applied to circumvent the problem of critical slowing down in U(1) gauge theory [17] and parameter regression task in [15]. In this work we explore a system with

fermions viz. the Gross-Neveu model[22]. In this work, following the ML approach we have used Generative Adversarial Network (GAN)[23] conditioned on parameter of the theory to efficiently generate lattice field configurations near critical point. ML based generative models generates uncorrelated samples which is one of the reason for using it as a replacement of MCMC simulation. To the best of our knowledge, GANs have not been applied to any fermionic system. However, normalizing flows have been used for Yukawa model [14]. In Ref[21], C-GANs have been found to be effective for studying phase transitions in XY model.

The critical point of a lattice theory corresponds to a particular value of the parameters (λ) of the theory. Our target is to generate uncorrelated samples from a probability distribution of kind: $P(\Phi|\lambda_{crit}) = \frac{1}{Z}e^{-S(\Phi,\lambda_{crit})}$ where Φ is lattice field, S is the action and Z is the partition function of the theory. The basic idea of our method consists of the following three steps :

1. Generate samples using HMC for λ away from critical region of the lattice theory : $\Phi \sim p(\Phi|\lambda_{noncrit})$.

2. Train the GAN models conditioned on the parameter λ using the data from step 1, i.e., learn the distribution $p(\Phi|\lambda)$.

3. Interpolate the trained generator model near the critical point and generate samples from the C-GAN model.

To implement the above ideas we have used a simple field theory in lattice - the Gross-Neveu model(GN model) in 1+1 dimensions [22]. GN model possesses many properties similar to QCD. It is an asymptotically free theory and re-normalizable in 1+1 dimension. GN model undergoes a spontaneous chiral phase transition and is extensively used in the literature as a toy model for QCD. With Wilson fermion, a parity broken phase (Aoki phase)

emerges on a finite lattice [24]. The Aoki phase structure of GN model with Wilson fermion and staggered fermion with flavored mass term has been investigated in strong coupling limit in [25]. The chiral phase transition of the GN model with minimally doubled Borici-Creutz fermion has been investigated in detail in Ref.[26]. The mass spectrum of GN model has been studied in Refs. [27, 28].

To check the validity of the generative model, we evaluate it in the critical region. For evaluation, we compare the observables calculated from the samples generated by the proposed C-GAN with those from the samples generated by HMC simulations. Since proposed C-GAN model's samples are independent, given λ, it alleviates the critical slowing down problem. Since the lattice constant a changes with parameter of the theory, we must choose the lattice size accordingly so that at critical region we get a lattice of desired physical volume.

II. GROSS-NEVEU MODEL IN 1+1 DIMENSION

A. Continuum Theory:

The Euclidean Lagrangian of GN model in 1+1 dimension is [22]:

$$\mathscr{L} = \sum_{f=1}^{N_f} \overline{\psi}_f(x)(\partial\!\!\!/)\psi_f(x) - \frac{g^2}{2}\Big(\sum_{f=1}^{N_f} \overline{\psi}_f(x)\psi_f(x)\Big)^2 \quad (1)$$

With the help of so called Hubbard-Stratonovich(HS) transformation we can reduce the four fermion part to a term quadratic in the fermion fields and an additional auxiliary bosonic field. The transformation is basically a shifted Gaussian integral.

$$\exp(-\int d^2x[\frac{g^2}{2}(\overline{\psi}_f(x)\psi_f(x))^2]$$
$$= \mathcal{N}[\int \mathscr{D}\sigma(x)exp(-\int d^2x[\frac{N_f}{2\tilde{\lambda}}\sigma^2(x) + \overline{\psi}_f(x)\sigma(x)\psi_f(x)]$$
$$(2)$$

where $\tilde{\lambda} = g^2 N_f$.

The partition function becomes

$$Z = \int \mathscr{D}\overline{\psi}\mathscr{D}\psi\mathscr{D}\sigma e^{-S_\sigma[\overline{\psi},\psi,\sigma]}. \quad (3)$$

The action is given by

$$S_\sigma[\overline{\psi},\psi,\sigma] = \int d^2x[\frac{N_f}{2\tilde{\lambda}}\sigma^2(x) + \overline{\psi}_f D_{GN}(x)\psi_f(x)], \quad (4)$$

where, $D_{GN} = \partial\!\!\!/ + \sigma(x)$
One can show that the σ field and condensate field $\overline{\psi}\psi$

are linked via

$$\langle\overline{\psi}(x)\psi(x)\rangle = \frac{-N_f}{\tilde{\lambda}}\langle\sigma(x)\rangle. \quad (5)$$

Therefore, the average of auxiliary field $\langle\sigma(x)\rangle$ can be referred to as the Chiral Condensate. This $\langle\sigma(x)\rangle$ can be used as an order parameter to study spontaneously chiral symmetry breaking of the GN model. GN model is analytically solvable in the infinite flavor limit:$N_f \to \infty$. Its phase structure has been studied extensively in this limit[29][30]. Inhomogeneous phases of GN model in lattice are also studied for finite number of flavors with proper continuum limit in [31, 32].

B. Lattice theory:

The action of lattice GN model in the staggered formalism [33] is generally written as

$$S = \sum_{x,y}[\frac{\lambda N_f}{2}\sigma^2(x) + \sum_{f=1}^{f=N_f}\overline{\chi}_f(x)D(x,y)\chi_f(y)] \quad (6)$$

where the coupling constant is inverted to $\lambda = 1/\tilde{\lambda}$ for simulation purpose and $D = D_1 + \Sigma$ with

$$D_1(x,y) = \frac{1}{2}[\delta_{x,y+\hat{1}} - \delta_{x,y-\hat{1}}] + \frac{1}{2}[\delta_{x,y+\hat{2}} - \delta_{x,y-\hat{2}}] \quad (7)$$

$$\Sigma_{xy} = \frac{1}{4}\delta_{xy}[\sigma(x)+\sigma(x-\hat{1})+\sigma(x-\hat{2})+\sigma(x-\hat{1}-\hat{2})] \quad (8)$$

where $\hat{1}$ and $\hat{2}$ are unit vectors in the two directions in 2D.

This theory has discrete chiral symmetry :

$$\chi \to (-1)^{x_1+x_2}\chi, \quad \overline{\chi} \to -(-1)^{x_1+x_2}\overline{\chi}, \quad \sigma(x) = -\sigma(x) \quad (9)$$

Higher N_f value is necessary to match continuum ($N_f \longrightarrow \infty$) results but $N_f = 2$ will serve our purpose in this work. After introducing pseudofermionic[34] method(for $N_f = 2$) action become non-local :

$$S[\sigma,\phi,\lambda] = \sum_{x,y}[\frac{\lambda}{4}\sigma^2(x) + \phi^\dagger(x)(M^{-1})\phi(y)] \quad (10)$$

where $M = D^\dagger D$ and ϕ are pseudofermionic complex field.

Partition function can be written as-

$$Z = \int \mathscr{D}\sigma\mathscr{D}\phi^\dagger\mathscr{D}\phi e^{-S[\phi,\phi^\dagger,\sigma]}. \quad (11)$$

With the action given in Eq.(10) we perform our HMC simulations. In this work we have used the staggered fermion(for details about staggered fermion, see [35]) for lattice simulation.

III. GENERATIVE ADVERSARIAL NETWORK(GAN)

Generative Adversarial Networks (GANs) can be trained to generate samples from a high dimensional probability distribution. A GAN [23] basically consist of two neural networks, namely, the generator and the discriminator, where the generator's prime job is to produce realistic samples and discriminator distinguishes generator's fake samples from the real samples. Notably, GAN learns from the samples from the true distribution, without using the true distribution explicitly. Likewise, it generates samples and does not explicitly tell the probability density.

The Generative model G, parameterized by Θ is a map from a random noise $z \sim p_z(z)$ to $x \sim p_g(G(z, \Theta))$. The training dataset is from a true distribution $x \sim p_{real}(x)$. The discriminator $D(x, \Phi)$ predict whether it is coming from p_g or p_{real}. After completion of the training we expect p_g to be as close as possible to p_{real}. The training process is a two players min-max game where discriminator improves its ability to distinguish generator's fake samples and generator also improves its ability to produce more realistic samples to fool the discriminator as the training continues. In the training process both the generator and discriminator's weights are updated in tandem.

The objective function of GAN is:

$$\min_{G} \max_{D} V(G, D) = E_{x \sim p_{real}(x)}[log D(x, \Phi)]$$
$$+ E_{z \sim p_g(z)}[log(1 - D(G(z, \Theta), \Phi))] \quad (12)$$

Conditional-GAN: If the true dataset has categories or classes, then the original GAN approach has no control over the type or class of output generated by the generator as output depends only on the random noise. But in many situations it become necessary to generate data of a particular type or class. So we want to train a GAN so that it can learn a conditional probability distribution.
In C-GAN [36] we append the random noise with additional information λ, which could be attributes or class labels to produce output $G(\lambda, z, \Theta)$, which is conditioned on λ. We also append λ to the input of discriminator.

The Objective function of C-GAN is:

$$\min_{G} \max_{D} V(\lambda, G, D) = E_{x \sim p_{real}(x)}[log D(\lambda, x, \Phi)]$$
$$+ E_{z \sim p_g(z)}[log(1 - D(\lambda, G(\lambda, z, \Theta), \Phi))]. \quad (13)$$

IV. HMC SIMULATION:

A. HMC Algorithm

HMC algorithm can be use to produce a Markov Chain whose stationary distribution is:

$$P(\sigma, \phi | \lambda) = \frac{1}{Z} e^{-S(\sigma, \phi | \lambda)}, \quad (14)$$

where $S(\sigma, \phi)$ is the lattice action and Z is the partition function defined in Eq.(10) and Eq.(11) respectively. However, this partition function does not represent a classical Hamiltonian system. We can transform it by introducing a canonically conjugate momentum variable $\pi(x)$ into the system. Then it become a Hamiltonian system, where Hamiltonian can be written as:

$$H(\sigma, \pi, \phi) = \frac{1}{2} \sum_x \pi^2(x) + S[\sigma, \phi] \quad (15)$$

In HMC algorithm we solve the Hamiltonian equation in discrete time for $\sigma(x)$ and $\pi(x)$. We can sample pseudofermionic variable ϕ easily by sampling complex vector ξ from $exp(-\xi^\dagger \xi)$ and setting $\phi = D^\dagger \xi$. This ensures that ϕ can be sampled according to the distribution (14) for a given σ. For details of HMC for psedofermion action refer to [35]. The common steps one follows in HMC simulation are:

1. Choose σ_0 configuration from cold-start or hot-start.

2. Choose π from random Gaussian distribution.

3. Choose ξ as Gaussian noise and Evaluate: $\phi = D^\dagger \xi$.

4. MD steps to update σ and π keeping ϕ as background field: Solve Hamiltonian differential equations for some discrete time step τ.

$$\frac{d}{d\tau} \sigma_x(\tau) = \frac{\partial}{\partial \pi_x} H(\pi(\tau), \sigma(\tau), \phi)$$
$$\frac{d}{d\tau} \pi_x(\tau) = -\frac{\partial}{\partial \sigma_x} H(\pi(\tau), \sigma(\tau), \phi)$$

It will generate new configurations $(\sigma_{new}, \pi_{new})$ as the next proposal.

5. Do Metropolis test to accept or reject the new configuration.

6. Return to step 2.
In this way we can generate ensemble of (σ, ϕ) configurations according to the distribution (14).

B. HMC Simulation and Observables

In this work, we simulate for $N_f = 2$, with lattice size=32×32. During HMC simulation, we adjust the MD step-size to keep acceptance rate around $\sim 80\%$ with legitimate autocorrelation time. We left first 500 lattice configurations for thermalization. At each λ values in the range [0.6 - 2.5], we generate 4000 lattice configurations for training dataset and baseline for evaluation of our proposed C-GAN, which is discussed further in section VI.

The quantity: $\bar{\sigma} = \frac{1}{N} \sum_x \sigma(x)$, which is measured in a single lattice configuration can be use to study the phase transition as it's ensemble average has direct relation to Chiral Condensate $\langle \bar{\psi}\psi \rangle$. However, there is a problem with quantity $\bar{\sigma}$ as its ensemble average $\langle \bar{\sigma} \rangle$ vanishes even for $\lambda \lesssim \lambda_{crit}$ i.e. even for broken phase close to the critical point. This can be seen from Fig.1 which is calculated near critical point where configurations fluctuates between two minimum and hence average $\langle \bar{\sigma} \rangle$ nearly vanishes. This is due to the ability of configurations to make tunnel from one minimum to the other. So instead of using $\langle \bar{\sigma} \rangle$, we choose $\langle |\bar{\sigma}| \rangle$ as our order parameter which is a suitable observable to study phase transition. One more observable of importance is susceptibility. The two observables can be defined as

$$\langle |\bar{\sigma}| \rangle = \frac{1}{N} \langle | \sum_x \sigma(x) | \rangle; \quad \chi = N[\langle \bar{\sigma}^2 \rangle - \langle |\bar{\sigma}| \rangle^2] \quad (16)$$

where N is the lattice volume.
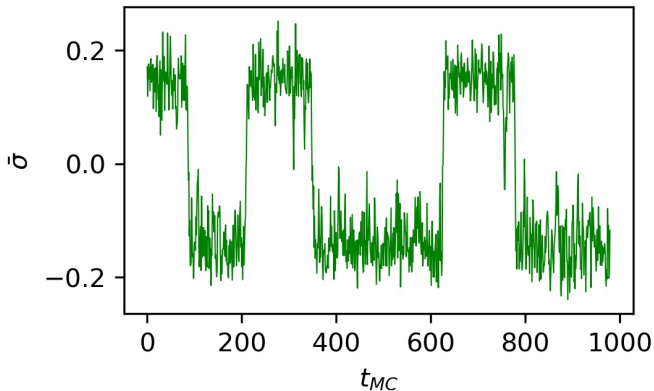
Figure 1. Fluctuations of $\bar{\sigma}$ values for lattice configurations during HMC simulation in between two minima at $\lambda \lesssim \lambda_{crit}$.

V. PROPOSED METHOD

In HMC simulation we sample σ, ϕ field according to the the distribution given in Eq.(14) i.e. $\sigma, \phi \sim P(\sigma, \phi | \lambda)$. We want the C-GAN to learn the marginal distribution of σ field i.e $p(\sigma | \lambda)$. So we discard HMC generated pseudofermionic ϕ samples and only consider σ samples which now represent the marginal distribution of σ from the joint distribution in Eq. (14). Let the samples from the C-GAN represent an implicit distribution $\hat{p}(\sigma | \lambda)$. Our target is to train the C-GAN so that $\hat{p}(\sigma | \lambda)$ approximates the true distribution $p(\sigma | \lambda)$. To address the problem of critical slowing down, we train the C-GAN model for λ values sampled in non-critical region, where the autocorrelation time is much smaller comparing to the critical region. Hence generation of training dataset is not affected by critical slowing down. Then we use the trained C-GAN model to generate samples near critical λ. Since C-GAN model generates independent samples, hence our method can produce uncorrelated lattice configurations in the critical region.

Vanilla C-GAN trained over the HMC samples fails to learn the distribution reliably. The learning is made efficient as well as robust by incorporating into the C-GAN model the information of symmetries and constraints in the theory. Also, transforming the samples so as to reduce the imbalance in their values improves learning. We discuss these in detail in the following subsections.

A. Translation Symmetry

Due to translation symmetry in GN model lattices, C-GAN generator made of dense layers fails to learn the true distribution properly. Convolutional kernels allow translational invariance in the lattices. Hence, using convolutional layers in the generator allows the learning to take place efficiently.

B. Transformation of σ field:

Since the observables of GN model can be calculated from $|\bar{\sigma}|$, hence for training the C-GAN we transformed the lattice configurations such that each configuration has $\bar{\sigma} > 0$. This will reduce degrees of freedom for the C-GAN model which will help in exploring the distribution space more efficiently. For that purpose, we select a particular configuration and if found $\bar{\sigma} < 0$ then we apply a transformation:

$$\sigma(x) = -\sigma(x), \quad \forall x. \quad (17)$$

For training purpose of C-GAN, we apply natural log transformation to the HMC generated samples as follows:

$$\sigma_i'(x) = ln(\sigma_i(x) + c), \quad \forall x, i \quad (18)$$

where i represent a single lattice configuration from the ensemble and c is a constant such that the sum inside the logarithm become positive. This transformation(18) become necessary for stable training of C-GAN as it balances data values and reduces the dynamical range of the $\sigma(x)$ field.

For efficient training we apply the Min-Max scaling to the above transformed data to bring into a range [-1,1].

C.   Periodic Boundary Condition:

During generation of configurations by HMC, we apply periodic boundary condition i.e. we replace $\sigma(i,j)$ by $\sigma'(i,j) := \sigma((i)_N, (j)_N)$, where $(i)_N$ represents i modulo N. In order to learn the periodicity by the C-GAN model we apply periodic padding to the all layers of generator and initial two layers of discriminator.

VI.   NUMERICAL EXPERIMENT & RESULTS

A.   Dataset:

Our training dataset is consisting of 10 ensembles, each of which has 4000 lattice configurations corresponding to 10 different λ values generated by HMC simulation. Λ_{tr} is the set of λ on which we train the C-GAN model. It includes λ values away from the critical region. Assuming $\lambda_{crit} \sim 1.5$, $\Lambda_{tr} = \{0.6, 0.8, 1.0, 1.2, 1.3, 1.8, 2.0, 2.2, 2.3, 2.5\}$. For inference and evaluation of the proposed C-GAN model we use Λ_{ts} which includes λ values in the critical region too, $\Lambda_{ts} = \{0.6, 0.8, 1.0, 1.3, 1.5, 1.6, 1.7, 2.0, 2.3, 2.5\}$.

B.   C-GAN Model Architecture and Training :

Input to the generator model consist of a 4×4 i.i.d Gaussian noise with zero mean and unit variance , stacked together with 4×4 matrix containing all entries as λ. In generator model three 2D Transposed convolutional layers are used for up-sampling to 32×32 final lattice configuration. We use kernel of sizes (3,3) & (4,4) and strides (1,1) & (2,2) for generator model. In discriminator the input is a grid of 32×32 σ samples, concatenated with a 32×32 channel with the λ value repeated in all the cells. It has three 2D convolutional layers with Tanh activation function followed by a dense layer with Sigmoid activation. We use kernel of sizes (4,4) and strides (2,2),(1,1) for discriminator.The detail architecture of generator and discriminator model is given in the appendix. We add periodic padding to the all layers of generator model and only two initial layers of discriminator model to learn the periodicity in the lattice configuration.

Once the training of C-GAN is over, we use the generator model to generate two ensembles each consist of 20000 configurations for Λ_{tr} and Λ_{ts} respectively. In both cases we evaluate our C-GAN model by comparing observables calculated on the above two set with those calculated from the HMC generated ensembles. The observables used for this purpose are: $\langle |\bar\sigma| \rangle$ and χ as defined in Eq.(16).

C.   Results

1.   Testing on Λ_{tr} Set:

We do the analysis on λ_{tr} set to confirm that the C-GAN model has correctly learned the training data distribution. In this ensemble we calculate the $\bar\sigma$ for each lattice configuration then plot the histogram of $|\bar\sigma|$ as shown in Fig.2. Different peaks in the histogram roughly corresponds to different λ values. The histogram generated from the proposed C-GAN model and HMC overlaps quite well. It indicates that our proposed distribution $\hat{p}(\sigma|\lambda)$ represented by C-GAN approximates the true distribution for the λ_{tr} set. Also in Fig.3, we plot ensemble averaged $\langle |\bar\sigma| \rangle$ for Λ_{tr} set. Here we take ensemble average $\langle |\bar\sigma| \rangle$ for each λ separately and then plot $\langle |\bar\sigma| \rangle$ vs λ. It shows that the observables are matching well for both C-GAN and HMC ensembles for Λ_{tr} i.e. the set used during training.
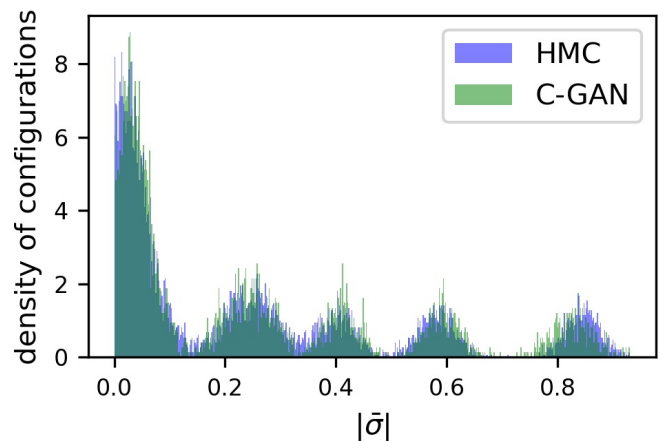
Figure 2. Histogram of $|\bar\sigma|$ for λ_{tr} set, estimated from samples obtained via HMC and C-GAN.
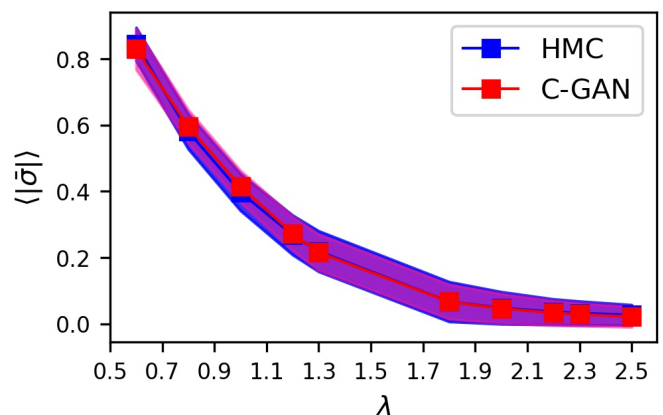
Figure 3. Mean $\langle |\bar\sigma| \rangle$ and standard deviation on λ_{tr} set, estimated from samples obtained via HMC and C-GAN.
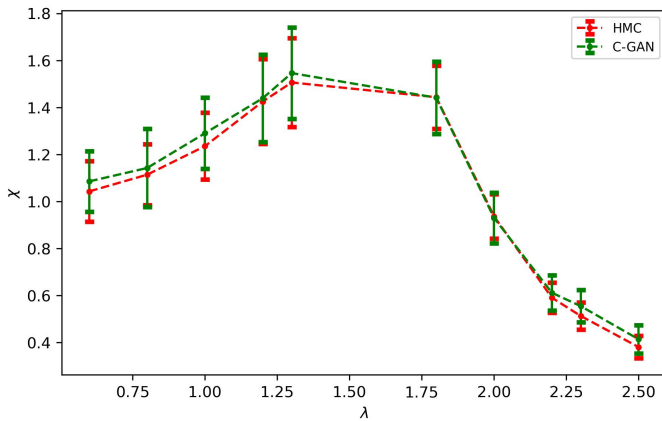
Figure 4. Susceptibility and its standard deviation on λ_{tr} set, estimated from 8000 samples with bin size of 100.

2. *Testing on* Λ_{ts} *Set:*

Since our main goal is to generate lattice ensembles in the critical region we must evaluate our C-GAN model in Λ_{ts}. We asses the performance of the proposed C-GAN in terms of being able to produce observables matching with those obtained from true distributions(i.e., generated by HMC).

Mean $\langle|\bar{\sigma}|\rangle$ **:** In Fig.5 we can observe that histogram of $|\bar{\sigma}|$ matches quite well with the true histogram obtained via HMC samples even for Λ_{ts}. Different peaks at high $|\bar{\sigma}|$ values roughly represent different λ values. However, there are no distinct peaks visible near low $|\bar{\sigma}|$ values as the peaks gets overlapped. We also present the histograms of $|\bar{\sigma}|$ in Figs.(6 & 7) for $\lambda \in \{1.5, 1.6\}$ which are in the critical region where we didn't train the model. In Fig.8 we present the results for the mean $\langle\bar{\sigma}\rangle$ in the critical region. We can see that the phase transition behaviour is described very well by the generator model.

Susceptibility χ **:** We show the susceptibility values obtained from HMC configurations as well as those obtained from C-GAN in Fig.4 for non-critical data set. One can observe that the peak coincides for both HMC and C-GAN. The same plot for critical dataset is shown in Fig.9. We have found that in critical region both mean $\langle\bar{\sigma}\rangle$ and susceptibility agree quite well with the HMC results even without training in that region. This gives a good indication that the trained model can reproduce the second order phase transition in the GN lattice model.

In Fig.10 we show the autocorrelation time generated from HMC simulation with unit trajectory length in MD step, while keeping acceptance rate $\approx 80\%$. We see that near the phase transition point the the autocorrelation time increases sharply. However, during sampling from the C-GAN model we starts with a random Gaussian noise vector to generate lattice configurations. Therefore, the lattice configurations generated by the C-GAN model are independent of each other, which will solve the critical slowing down problem. In this way we can generate

uncorrelated samples near critical region at the cost of generation of samples by HMC at the non critical region.
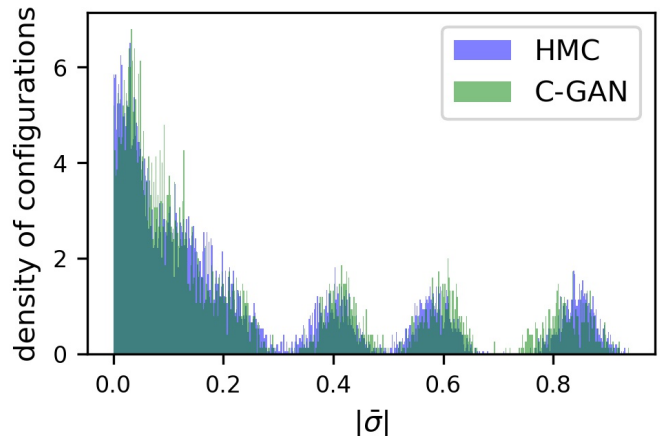
Figure 5. Histogram of $|\bar{\sigma}|$ for Λ_{ts} set, estimated from samples obtained via HMC and C-GAN.
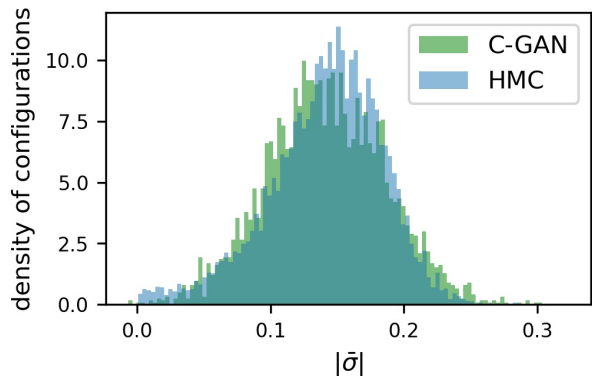
Figure 6. Histogram of $|\bar{\sigma}|$ at $\lambda = 1.5 \in \Lambda_{ts}$: HMC and C-GAN histograms overlaps quite well
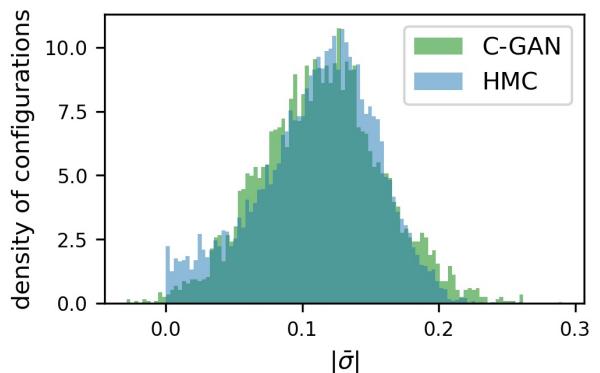
Figure 7. Histogram of $|\bar{\sigma}|$ at $\lambda = 1.6 \in \Lambda_{ts}$: HMC and C-GAN histograms overlaps quite well
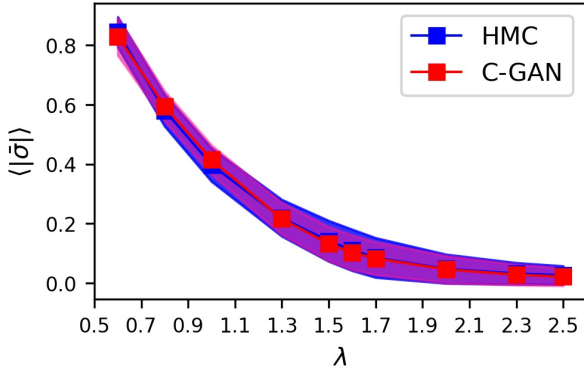
Figure 8. Mean $\langle|\bar{\sigma}|\rangle$ and standard deviation for Λ_{ts} set, estimated from samples obtained via HMC and C-GAN.
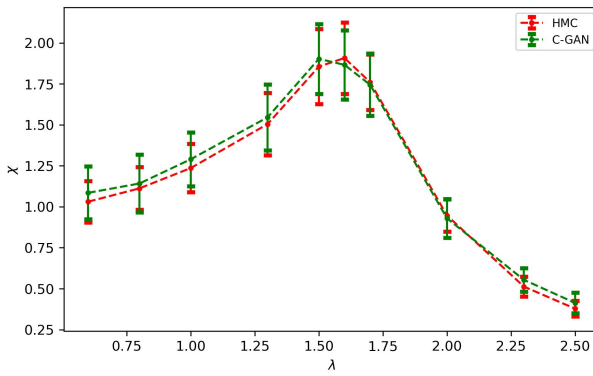
Figure 9. Susceptibility and its standard deviation on λ_{ts} set, estimated from 8000 samples with bin size of 100.
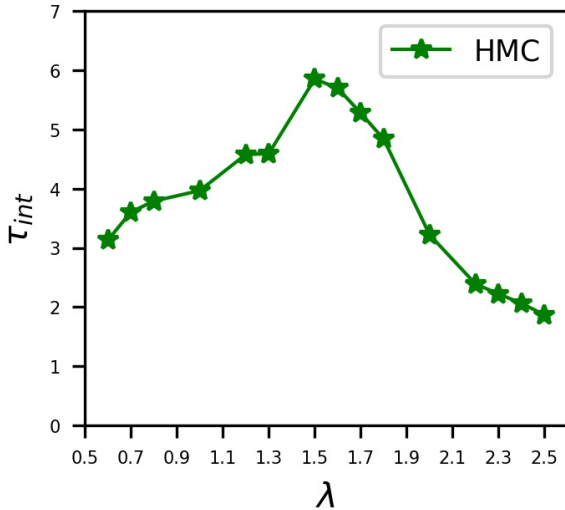
Figure 10. Integrated Autocorrelation time estimated from HMC simulation with MD trajectory length, $\tau = 1$.

D. Numerical experiment with data from a single phase

We also train the C-GAN model using HMC generated dataset consiting of λ values only from one single phase. This experiment is necessary to check our model's utility in lattice gauge theory where extrapolation to critical point is necessary from one direction of parameter space.
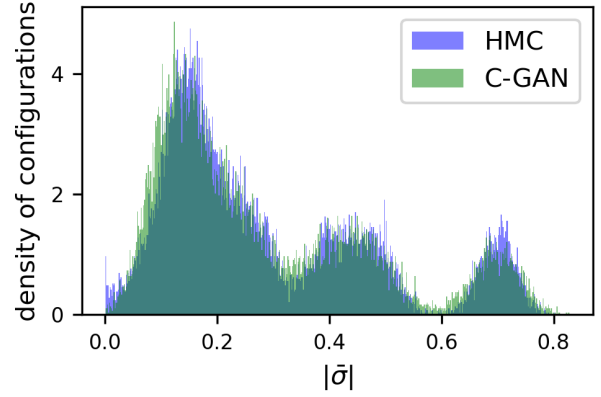
Figure 11. Histogram of $|\bar{\sigma}|$ forΛ_{1ph}^{tr} set, estimated from samples obtained via HMC and C-GAN.
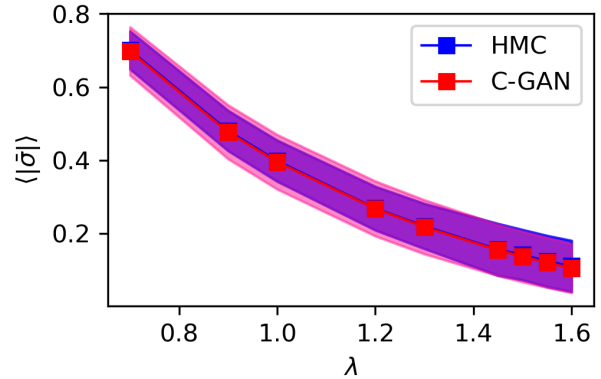
Figure 12. Mean $\langle|\bar{\sigma}|\rangle$ and standard deviation for Λ_{1ph}^{ts} set, estimated from samples obtained via HMC and C-GAN.

The training set of λ values are $\lambda_{1ph}^{tr} = \{0.4$,0.6, 0.7 ,0.8 ,0.9 ,1.0 ,1.1 ,1.2 ,1.25 ,1.3 ,1.35 ,1.4$\}$ taken from the broken phase and the test set of λ values are $\lambda_{1ph}^{ts} = \{0.7$,0.9, ,1.0 ,1.2, 1.3 ,1.45 ,1.5 ,1.55, 1.6$\}$. We extrapolate the C-GAN model to critical region of λ values 1.45,1.5 ,1.55 and 1.6.

The results are shown in figure 11-15. We observe that the histogram and mean $\langle\bar{\sigma}\rangle$ matches quite well with HMC results for λ_{1ph}^{ts}, where critical points are included. Also in figure 12,13,14 we have shown the individual histogram of $|\bar{\sigma}|$ for λ=1.5 ,1.55, 1.6. We found that for the critical λ

values observables does't differ either we train the model with one single phase or consider both the phases.
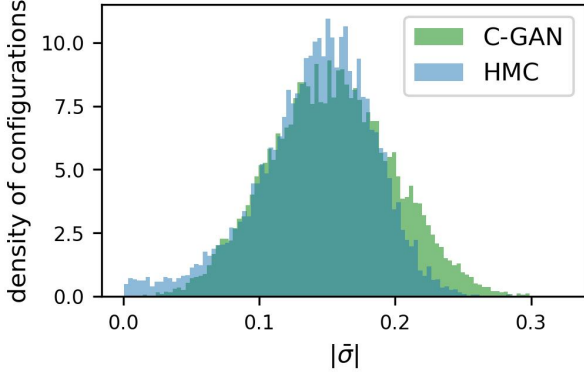
Figure 13. Histogram of $|\bar{\sigma}|$ at $\lambda = 1.5 \in \Lambda_{1ph}^{ts}$: HMC and C-GAN histograms overlaps quite well
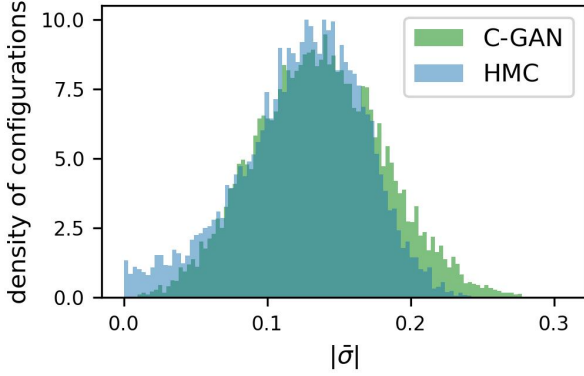
Figure 14. Histogram of $|\bar{\sigma}|$ at $\lambda = 1.55 \in \Lambda_{1ph}^{ts}$: HMC and C-GAN histograms overlaps quite well
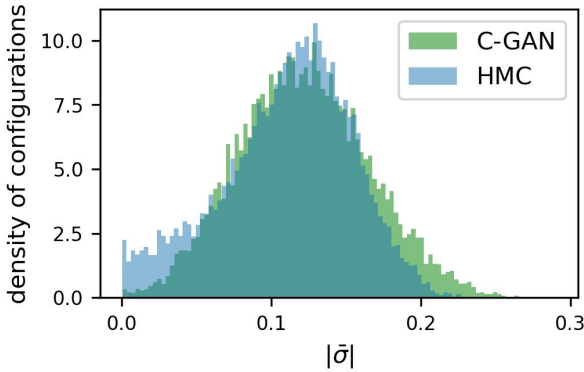
Figure 15. Histogram of $|\bar{\sigma}|$ at $\lambda = 1.6 \in \Lambda_{1ph}^{ts}$: HMC and C-GAN histograms overlaps quite well.

E.   ABLATION ANALYSIS:

We perform ablation analysis to see the effect of certain key component of the proposed method on its performance.

Transformation of σ field: We find that log transformation (18) is one of the crucial component for the training of C-GAN model. On removing it ,the training loss becomes high and the observables do not agree well with the HMC observables which can be seen from Figs.16-18. There is a large deviation of mean of $|\bar{\sigma}|$ compared to HMC results in both critical and non-critical regions as shown in Fig.16. It is observed that susceptibility is too sensitive to log transformation as shown in Fig.17. It is also seen that without Min-Max scaling the C-GAN model is unable to learn different modes corresponding to different λ values.

Periodic Boundary Condition: We have noticed that the C-GAN performs well using periodic padding in both discriminator and generator as seen from Figs.5-7. But when we remove periodicity from both discriminator and generator, C-GAN fails to reproduce the HMC results. Figs.19 & 20 show the disagreements between C-GAN and HMC results for $\langle|\bar{\sigma}|\rangle$, susceptibility χ respectively and Fig.21 compares the histogram of $|\bar{\sigma}|$ without periodicity in C-GAN for $\lambda = 1.5$. Likewise, not applying periodic padding to the generator and applying only to the discriminator, also degrades the performance.
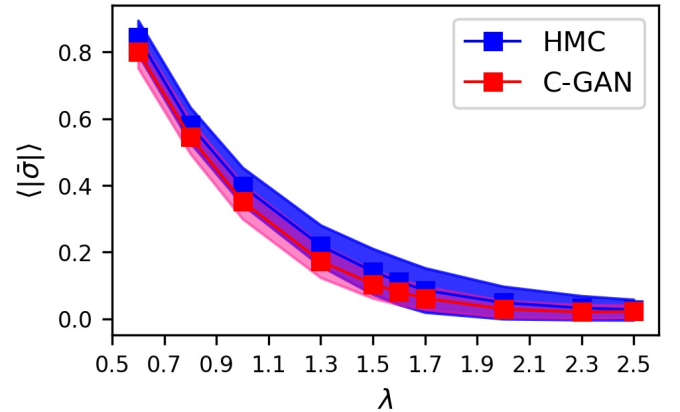
Figure 16. Ablation for log transformation: Mean $\langle|\bar{\sigma}|\rangle$ and standard deviation on Λ_{ts} set without log transformation
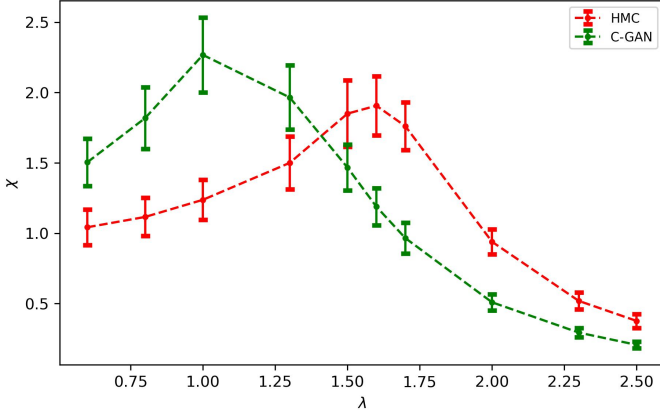
Figure 17. Ablation for log transformation: Susceptibility on Λ_{ts} set without log transformation.
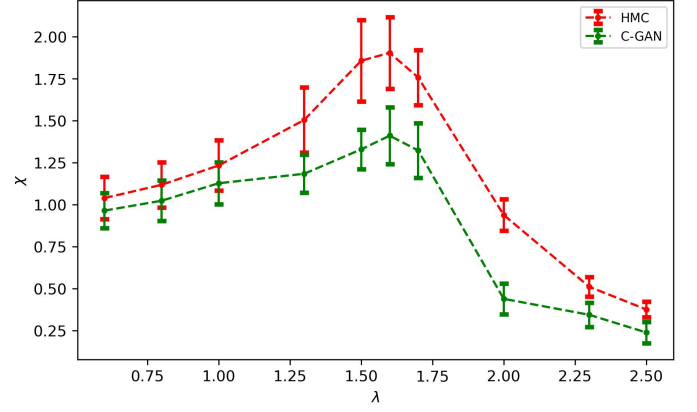
Figure 20. Ablation for periodic padding: Susceptibility on Λ_{ts} set without periodic padding in both generator and discriminator.
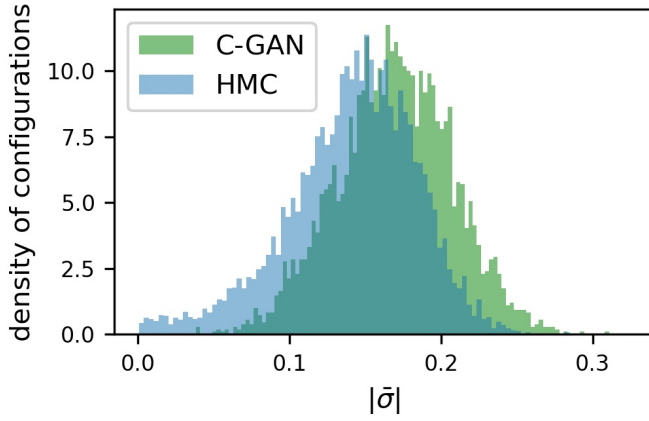
Figure 18. Ablation for log transformation: Histogram of $|\bar{\sigma}|$ at $\lambda = 1.5 \in \Lambda_{ts}$ set without log transformation
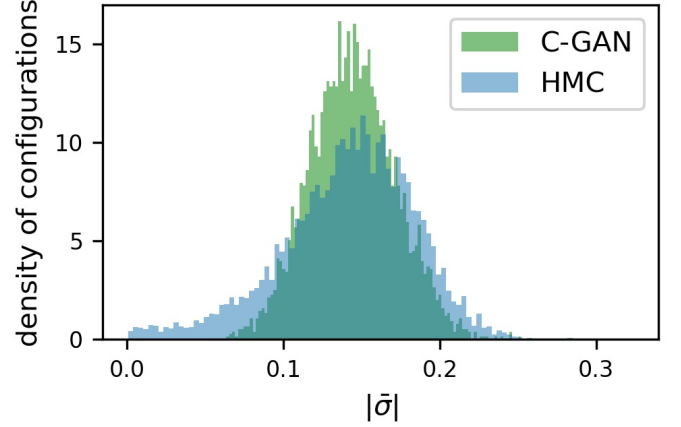
Figure 21. Ablation for periodic padding: Histogram of $|\bar{\sigma}|$ at $\lambda = 1.5 \in \Lambda_{ts}$ set without periodic padding in both generator and discriminator
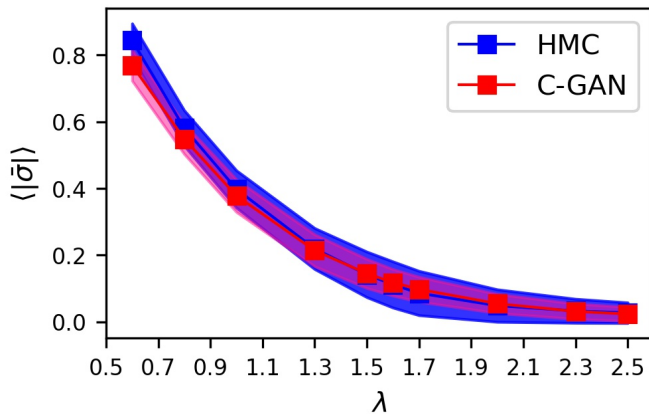
Figure 19. Ablation for periodic padding: Mean $\langle |\bar{\sigma}| \rangle$ and standard deviation on Λ_{ts} set without periodic padding in both generator and discriminator

VII. SUMMARY & CONCLUSION:

MCMC methods are generally used to generate lattices as they give theoretical guarantees on validity of samples. In this work, we use GANs which don't give theoretical guarantees but the empirical results show that they are able to efficiently interpolate as well as extrapolate to critical regions. In lattice field theory, the cost of generation of lattice configurations by MCMC methods is severely affected by critical slowing down as the lattice parameters are tuned towards the critical region. At the critical point the cost of HMC simulation diverges for theories like QCD due to the diverging autocorrelation time. Therefore, generation of configurations in lattice field theory in the critical region is a challenging task. This paper proposes to use HMC generated configurations for GN model away from the critical region and trained a C-GAN to generate lattice configuration near critical point. With

HMC data in non-critical region, we train the C-GAN model conditioned with parameter λ. For evaluation of the proposed C-GAN model at critical region we compare few observables on the samples generated from both C-GAN and HMC. We found a good matching between the results of HMC and our C-GAN model and also observed that phase transition can be very well reproduced by the generative approach. Since the C-GAN model in the critical region gives correct critical behaviour, we can infer that our generative model is a good interpolator in the critical region. Since C-GAN generates independent configurations, there is no correlation in the samples generated by the C-GAN model, thereby avoiding the critical slowing down problem. In this work we evaluate our proposed C-GAN model by comparing observables with HMC samples. We could also use our C-GAN model distribution $\hat{p}(\sigma|\lambda)$ as proposal distribution to construct a Markov chain as done in [13]. However, to construct such a Markov chain we must know the proposal density explicitly which is not available for GANs. Rather in this work we have accepted all the samples generated by the C-GAN model and thus having a vanishing autocorrelation.

Although the problem of critical slowing down is not as severe for GN model in 1+1 dimensions but building and testing the C-GAN in the GN model establishes its applicability in the lattice formulation of fermionic system. In this work, we dealt with only fermionic fields without any gauge interaction. Extending our work to lattice gauge theory and QCD will be an interesting as well as challenging task.

Acknowledgments

Thanks to Prof. William Detmold for helpful discussions, and to SPARC for facilitating his visit at IIT Kanpur.

[1] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth, Physics letters B **195**, 216 (1987).

[2] U. Wolff, Nucl. Phys. B Proc. Suppl. **17**, 93 (1990).

[3] S. Schaefer, R. Sommer, and F. Virotta (ALPHA), Nucl. Phys. B **845**, 93 (2011), arXiv:1009.5228 [hep-lat].

[4] A. Ramos, PoS **LATTICE2012**, 193 (2012), arXiv:1212.3800 [hep-lat].

[5] A. S. Gambhir and K. Orginos, "Improved sampling algorithms in lattice qcd," (2015), arXiv:1506.06118 [hep-lat].

[6] M. G. Endres, R. C. Brower, W. Detmold, K. Orginos, and A. V. Pochinsky, Physical Review D **92** (2015), 10.1103/physrevd.92.114516.

[7] G. Torlai and R. G. Melko, Phys. Rev. B **94**, 165134 (2016).

[8] G. Carleo and M. Troyer, Science **355**, 602 (2017).

[9] L. Wang, Phys. Rev. B **94**, 195105 (2016).

[10] P. Zhang, H. Shen, and H. Zhai, Phys. Rev. Lett. **120**, 066401 (2018).

[11] L. Huang and L. Wang, Phys. Rev. B **95**, 035105 (2017).

[12] J. Liu, Y. Qi, Z. Y. Meng, and L. Fu, Phys. Rev. B **95**, 041101 (2017).

[13] M. S. Albergo, G. Kanwar, and P. E. Shanahan, Phys. Rev. D **100**, 034515 (2019).

[14] 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," (2021), arXiv:2106.05934 [hep-lat].

[15] P. E. Shanahan, D. Trewartha, and W. Detmold, Physical Review D **97**, 094506 (2018).

[16] D. C. Hackett, C.-C. Hsieh, M. S. Albergo, D. Boyda, J.-W. Chen, K.-F. Chen, K. Cranmer, G. Kanwar, and P. E. Shanahan, (2021), arXiv:2107.00734 [hep-lat].

[17] G. Kanwar, M. S. Albergo, D. Boyda, K. Cranmer, D. C. Hackett, S. Racanière, D. J. Rezende, and P. E. Shanahan, Physical Review Letters **125** (2020), 10.1103/physrevlett.125.121601.

[18] J. M. Pawlowski and J. M. Urban, Machine Learning: Science and Technology **1**, 045011 (2020).

[19] M. S. Albergo, D. Boyda, D. C. Hackett, G. Kanwar, K. Cranmer, S. Racanière, D. J. Rezende, and P. E. Shanahan, "Introduction to normalizing flows for lattice field theory," (2021), arXiv:2101.08176 [hep-lat].

[20] D. Boyda, G. Kanwar, S. Racanière, D. J. Rezende, M. S. Albergo, K. Cranmer, D. C. Hackett, and P. E. Shanahan, Physical Review D **103** (2021), 10.1103/physrevd.103.074504.

[21] J. Singh, M. Scheurer, and V. Arora, SciPost Physics **11** (2021), 10.21468/scipostphys.11.2.043.

[22] D. J. Gross and A. Neveu, Phys. Rev. D **10**, 3235 (1974).

[23] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," (2014), arXiv:1406.2661 [stat.ML].

[24] S. Aoki, Phys. Rev. D **30**, 2653 (1984).

[25] M. Creutz, T. Kimura, and T. Misumi, Phys. Rev. D **83**, 094506 (2011), arXiv:1101.4239 [hep-lat].

[26] J. Goswami, D. Chakrabarti, and S. Basak, Phys. Rev. D **91**, 014507 (2015), arXiv:1409.7999 [hep-lat].

[27] J. Danzer and C. Gattringer, PoS **LATTICE2007**, 092 (2007), arXiv:0710.1711 [hep-lat].

[28] J. Goswami, D. Chakrabarti, and S. Basak, Int. J. Mod. Phys. A **32**, 1750059 (2017), arXiv:1511.07678 [hep-lat].

[29] M. Thies and K. Urlichs, Physical Review D **67** (2003), 10.1103/physrevd.67.125015.

[30] O. Schnetz, M. Thies, and K. Urlichs, Annals of Physics **314**, 425–447 (2004).

[31] L. Pannullo, J. Lenz, M. Wagner, B. Wellegehausen, and A. Wipf, Acta Physica Polonica B Proceedings Supplement **13**, 127 (2020).

[32] J. Lenz, L. Pannullo, M. Wagner, B. Wellegehausen, and A. Wipf, Phys. Rev. D **101**, 094512 (2020), arXiv:2004.00295 [hep-lat].

[33] P. De Forcrand and U. Wenger, PoS **LAT2006**, 152 (2006), arXiv:hep-lat/0610117.

[34] C. Gattringer and C. B. Lang, Quantum chromodynamics on the lattice, Vol. 788 (Springer, Berlin, 2010).

[35] H. J. Rothe, Lattice gauge theories: An Introduction, Vol. 43 (1992).

[36] M. Mirza and S. Osindero, "Conditional generative adversarial nets," (2014), arXiv:1411.1784 [cs.LG].

APPENDIX:
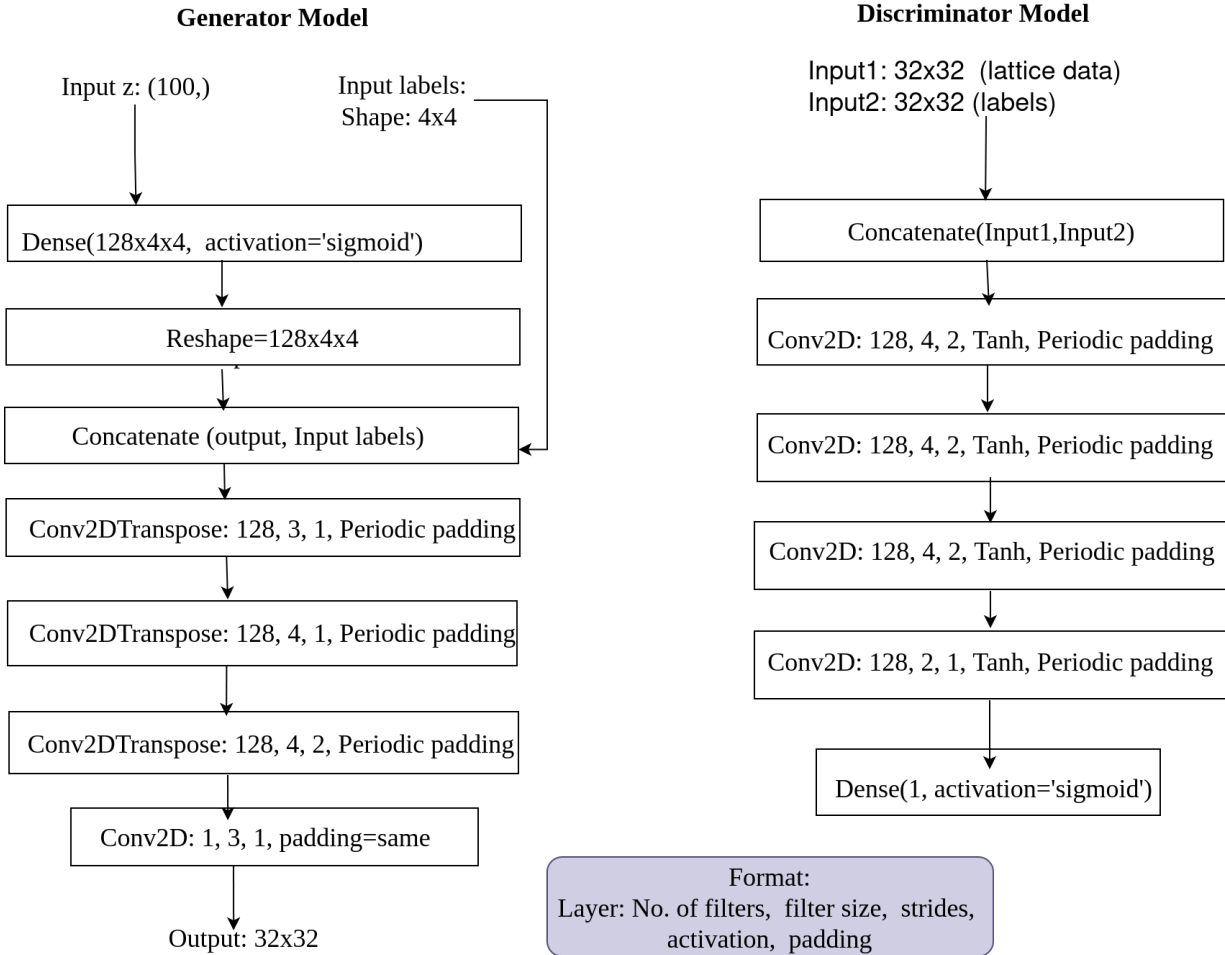
Figure 22. **Architecture of C-GAN models: Generator and Discriminator model**