

Reparameterized Sampling for Generative Adversarial Networks

No Author Given

No Institute Given

Abstract. Recently, sampling methods have been successfully applied to enhance the sample quality of Generative Adversarial Networks (GANs). However, in practice, they typically have poor sample efficiency because of the independent proposal sampling from the generator. In this work, we propose REP-GAN, a novel sampling method that allows general dependent proposals by REParameterizing the Markov chains into the latent space of the generator. Theoretically, we show that our reparameterized proposal admits a closed-form Metropolis-Hastings acceptance ratio. Empirically, extensive experiments on synthetic and real datasets demonstrate that our REP-GAN largely improves the sample efficiency and obtains better sample quality simultaneously.

Keywords: Generative Adversarial Networks · Sampling · Markov Chain Monte Carlo · Reparameterization.

1 Introduction

Generative Adversarial Networks (GANs) [8] have achieved a great success on generating realistic images in recent years [11,4]. Unlike previous models that explicitly parameterize the data distribution, GANs rely on an alternative optimization between a generator and a discriminator to learn the data distribution implicitly. However, in practice, samples generated by GANs still suffer from problems such as mode collapse and bad artifacts.

Recently, sampling methods have shown promising results on enhancing the sample quality of GANs by making use of the information in the discriminator. In the alternative training scheme of GANs, the generator only performs a few updates for the inner loop and has not fully utilized the density ratio information estimated by the discriminator. Thus, after GAN training, the sampling methods propose to further utilize this information to bridge the gap between the generative distribution and the data distribution in a fine-grained manner. For example, DRS [2] applies rejection sampling, and MH-GAN [26] adopts Markov chain Monte Carlo (MCMC) sampling for the improved sample quality of GANs. Nevertheless, these methods still suffer a lot from the sample efficiency problem. For example, as will be shown in Section 5, MH-GAN’s average acceptance ratio on CIFAR10 can be lower than 5%, which makes the Markov chains slow to mix. As MH-GAN adopts an *independent* proposal q , i.e., $q(\mathbf{x}'|\mathbf{x}) = q(\mathbf{x}')$, the difference between samples can be so large that the proposal gets rejected easily.

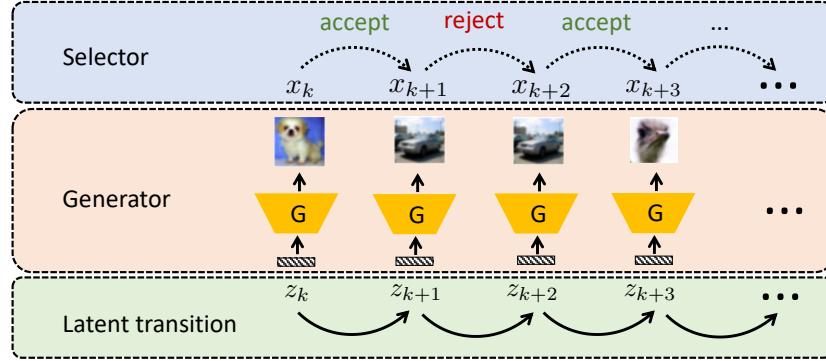


Fig. 1: Illustration of REP-GAN’s reparameterized proposal with two pairing Markov chains, one in the latent space \mathcal{Z} , and the other in the sample space \mathcal{X} .

To address this limitation, we propose to generalize the independent proposal to a general *dependent* proposal $q(\mathbf{x}'|\mathbf{x})$. To the end, the proposed sample can be a refinement of the previous one, which leads to a higher acceptance ratio and better sample quality. We can also balance between the exploration and exploitation of the Markov chains by tuning the step size. However, it is hard to design a proper dependent proposal in the high dimensional sample space \mathcal{X} because the energy landscape could be very complex [18].

Nevertheless, we notice that the generative distribution $p_g(\mathbf{x})$ of GANs is implicitly defined as the push-forward of the latent prior distribution $p_0(\mathbf{z})$, and designing proposals in the low dimensional latent space is generally much easier. Hence, GAN’s latent variable structure motivates us to design a *structured dependent proposal* with two pairing Markov chains, one in the sample space \mathcal{X} and the other in the latent space \mathcal{Z} . As shown in Figure 1, given the current pairing samples $(\mathbf{z}_k, \mathbf{x}_k)$, we draw the next proposal \mathbf{x}' in a bottom-to-up way: 1) drawing a latent proposal \mathbf{z}' following $q(\mathbf{z}'|\mathbf{z}_k)$; 2) pushing it forward through the generator and getting the sample proposal $\mathbf{x}' = G(\mathbf{z}')$; 3) assigning $\mathbf{x}_{k+1} = \mathbf{x}'$ if the proposal \mathbf{x}' is accepted, otherwise rejected $\mathbf{x}_{k+1} = \mathbf{x}_k$. By utilizing the underlying structure of GANs, the proposed reparameterized sampler becomes more efficient in the low-dimensional latent space. We summarize our main contributions as follows:

- We propose a structured dependent proposal of GANs, which reparameterizes the sample-level transition $\mathbf{x} \rightarrow \mathbf{x}'$ into the latent-level $\mathbf{z} \rightarrow \mathbf{z}'$ with two pairing Markov chains. We prove that our reparameterized proposal admits a tractable acceptance criterion.
- Our proposed method, called REP-GAN, serves as a unified framework for the existing sampling methods of GANs. It provides a better balance between exploration and exploitation by the structured dependent proposal, and also corrects the bias of Markov chains by the acceptance-rejection step.

Table 1: Comparison of sampling methods for GANs in terms of three effective sampling mechanisms.

Method	Rejection step	Markov chain	Latent gradient proposal
GAN	✗	✗	✗
DRS [2]	✓	✗	✗
MH-GAN [26]	✓	✓	✗
DDLS [5]	✗	✓	✓
REP-GAN (ours)	✓	✓	✓

- Empirical results demonstrate that REP-GAN achieves better image quality and much higher sample efficiency than the state-of-the-art methods on both synthetic and real datasets.

2 Related Work

Although GANs are able to synthesize high-quality images, the minimax nature of GANs makes it quite unstable, which usually results in degraded sample quality. A vast literature has been developed to fix the problems of GANs ever since, including network modules [17], training mechanisms [16] and objectives [1].

Moreover, there is another line of work using sampling methods to improve the sample quality of GANs. DRS [2] firstly proposes to use rejection sampling. MH-GAN [26] instead uses the Metropolis-Hastings (MH) algorithm with an independent proposal. DDLS [5] and DCD [23] apply gradient-based proposals by viewing GAN as an energy-based model. [24] proposes a similar gradient-based method named DOT from the perspective of optimal transport.

Different from them, our REP-GAN introduces a structured dependent proposal through latent reparameterization, and includes all three effective sampling mechanisms, the Markov Chain Monte Carlo method, the acceptance-rejection step, and the latent gradient-based proposal, to further improve the sample efficiency. As shown in Table 1, many existing works are special cases of our REP-GAN.

Our method also belongs to the thread of works that combine MCMC and neural networks for better sample quality. Previously, some works combine variational autoencoders [12] and MCMC to bridge the amortization gap [21,10,14], while others directly learn a neural proposal function for MCMC [22,13,27]. Our work instead reparameterizes the high-dimensional sample-level transition into a simpler low-dimensional latent space via the learned generator network.

3 Background

GANs model the data distribution $p_d(\mathbf{x})$ implicitly with a generator $G : \mathcal{Z} \rightarrow \mathcal{X}$ mapping from a low-dimensional latent space \mathcal{Z} to a high-dimensional sample

space \mathcal{X} ,

$$\mathbf{x} = G(\mathbf{z}), \quad \mathbf{z} \sim p_0(\mathbf{z}), \quad (1)$$

where the sample \mathbf{x} follows the generative distribution $p_g(\mathbf{x})$ and the latent variable \mathbf{z} follows the prior distribution $p_0(\mathbf{z})$, e.g., a standard normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. In GANs, a discriminator $D : \mathcal{X} \rightarrow [0, 1]$ is learned to distinguish samples from $p_d(\mathbf{x})$ and $p_g(\mathbf{x})$ in an adversarial way

$$\min_G \max_D \mathbb{E}_{\mathbf{x} \sim p_d(\mathbf{x})} \log(D(\mathbf{x})) + \mathbb{E}_{\mathbf{z} \sim p_0(\mathbf{z})} \log(1 - D(G(\mathbf{z}))). \quad (2)$$

[8] point out that an optimal discriminator D implies the density ratio between the data and generative distributions

$$D(\mathbf{x}) = \frac{p_d(\mathbf{x})}{p_d(\mathbf{x}) + p_g(\mathbf{x})} \Rightarrow \frac{p_d(\mathbf{x})}{p_g(\mathbf{x})} = \frac{1}{D(\mathbf{x})^{-1} - 1}. \quad (3)$$

Markov Chain Monte Carlo (MCMC) refers to a kind of sampling methods that draw a chain of samples $\mathbf{x}_{1:K} \in \mathcal{X}^K$ from a target distribution $p_t(\mathbf{x})$. We denote the initial distribution as $p_0(\mathbf{x})$ and the proposal distribution as $q(\mathbf{x}'|\mathbf{x}_k)$. With the Metropolis-Hastings (MH) algorithm, we accept the proposal $\mathbf{x}' \sim q(\mathbf{x}'|\mathbf{x}_k)$ with probability

$$\alpha(\mathbf{x}', \mathbf{x}_k) = \min \left(1, \frac{p_t(\mathbf{x}') q(\mathbf{x}_k|\mathbf{x}')}{p_t(\mathbf{x}_k) q(\mathbf{x}'|\mathbf{x}_k)} \right) \in [0, 1]. \quad (4)$$

If \mathbf{x}' is accepted, $\mathbf{x}_{k+1} = \mathbf{x}'$, otherwise $\mathbf{x}_{k+1} = \mathbf{x}_k$. Under mild assumptions, the Markov chain is guaranteed to converge to $p_t(\mathbf{x})$ as $K \rightarrow \infty$. In practice, the sample efficiency of MCMC crucially depends on the proposal distribution to trade off between exploration and exploitation.

4 The Proposed REP-GAN

In this section, we first review MH-GAN and point out the limitations. We then propose our structured dependent proposal to overcome these obstacles, and finally discuss its theoretical properties as well as practical implementations.

4.1 From Independent Proposal to Dependent Proposal

MH-GAN [26] first proposes to improve GAN sampling with MCMC. Specifically, given a perfect discriminator D and a descent (but imperfect) generator G after training, they take the data distribution $p_d(\mathbf{x})$ as the target distribution and use the generator distribution $p_g(\mathbf{x})$ as an independent proposal

$$\mathbf{x}' \sim q(\mathbf{x}'|\mathbf{x}_k) = q(\mathbf{x}') = p_g(\mathbf{x}'). \quad (5)$$

With the MH criterion (Eqn. (4)) and the density ratio (Eqn. (3)), we should accept \mathbf{x}' with probability

$$\alpha_{\text{MH}}(\mathbf{x}', \mathbf{x}_k) = \min \left(1, \frac{p_d(\mathbf{x}') q(\mathbf{x}_k)}{p_d(\mathbf{x}_k) q(\mathbf{x}')} \right) = \min \left(1, \frac{D(\mathbf{x}_k)^{-1} - 1}{D(\mathbf{x}')^{-1} - 1} \right). \quad (6)$$

However, to achieve tractability, MH-GAN adopts an independent proposal $q(\mathbf{x}')$ with poor sample efficiency. As the proposed sample \mathbf{x}' is independent of the current sample \mathbf{x}_k , the difference between the two samples can be so large that it results in a very low acceptance probability. Consequently, samples can be trapped in the same place for a long time, leading to a very slow mixing of the chain.

A natural solution is to take a *dependent* proposal $q(\mathbf{x}'|\mathbf{x}_k)$ that will propose a sample \mathbf{x}' close to the current one \mathbf{x}_k , which is more likely to be accepted. Nevertheless, the problem of such a dependent proposal is that its MH acceptance criterion

$$\alpha_{\text{DEP}}(\mathbf{x}', \mathbf{x}_k) = \min \left(1, \frac{p_d(\mathbf{x}') q(\mathbf{x}_k|\mathbf{x}')}{p_d(\mathbf{x}_k) q(\mathbf{x}'|\mathbf{x}_k)} \right), \quad (7)$$

is generally intractable because the data density $p_d(\mathbf{x})$ is unknown. Besides, it is hard to design a proper dependent proposal $q(\mathbf{x}'|\mathbf{x}_k)$ in the high dimensional sample space \mathcal{X} with complex landscape. These obstacles prevent us from adopting a dependent proposal that is more suitable for MCMC.

4.2 A Tractable Structured Dependent Proposal with Reparameterized Markov Chains

As discussed above, the major difficulty of a general dependent proposal $q(\mathbf{x}'|\mathbf{x}_k)$ is to compute the MH criterion. We show that it can be made tractable by considering an additional pairing Markov chain in the latent space.

As we know, samples of GANs lie in a low-dimensional manifold induced by the push-forward of the latent variable. Suppose that at the k -th step of the Markov chain, we have a GAN sample \mathbf{x}_k with latent \mathbf{z}_k . Instead of drawing a sample \mathbf{x}' directly from a sample-level proposal distribution $q(\mathbf{x}'|\mathbf{x}_k)$, we first draw a latent proposal \mathbf{z}' from a dependent latent proposal distribution $q(\mathbf{z}'|\mathbf{z}_k)$. Afterward, we push the latent \mathbf{z}' forward through the generator and get the output \mathbf{x}' as our sample proposal.

As illustrated in Figure 1, our bottom-to-up proposal relies on the transition reparameterization with two pairing Markov chains in the sample space \mathcal{X} and the latent space \mathcal{Z} . Hence we call it a REP (reparameterized) proposal. Through a learned generator, we transport the transition $\mathbf{x}_k \rightarrow \mathbf{x}'$ in the high dimensional space \mathcal{X} into the low dimensional space \mathcal{Z} , $\mathbf{z}_k \rightarrow \mathbf{z}'$, which enjoys a much better landscape and makes it easier to design proposals in MCMC algorithms. For example, the latent target distribution is nearly standard normal when the generator is nearly perfect. In fact, under mild conditions, the REP proposal distribution $q_{\text{REP}}(\mathbf{x}'|\mathbf{x}_k)$ and the latent proposal distribution $q(\mathbf{z}'|\mathbf{z}_k)$ are tied with the following change of variables [7,3]

$$\log q_{\text{REP}}(\mathbf{x}'|\mathbf{x}_k) = \log q(\mathbf{x}'|\mathbf{z}_k) = \log q(\mathbf{z}'|\mathbf{z}_k) - \frac{1}{2} \log \det J_{\mathbf{z}'}^\top J_{\mathbf{z}'}, \quad (8)$$

where $J_{\mathbf{z}}$ denotes the Jacobian matrix of the push-forward G at \mathbf{z} , i.e., $[J_{\mathbf{z}}]_{ij} = \partial \mathbf{x}_i / \partial \mathbf{z}_j$, $\mathbf{x} = G(\mathbf{z})$.

Nevertheless, it remains unclear whether we can perform the MH test to decide the acceptance of the proposal \mathbf{x}' . Note that a general dependent proposal distribution does not meet a tractable MH acceptance criterion (Eqn. (7)). Perhaps surprisingly, it can be shown that with our structured REP proposal, the MH acceptance criterion is tractable for general latent proposals $q(\mathbf{z}'|\mathbf{z}_k)$.

Theorem 1. *Consider a Markov chain of GAN samples $\mathbf{x}_{1:K}$ with initial distribution $p_g(\mathbf{x})$. For step $k+1$, we accept our REP proposal $\mathbf{x}' \sim q_{\text{REP}}(\mathbf{x}'|\mathbf{x}_k)$ with probability*

$$\alpha_{\text{REP}}(\mathbf{x}', \mathbf{x}_k) = \min \left(1, \frac{p_0(\mathbf{z}')q(\mathbf{z}_k|\mathbf{z}')}{p_0(\mathbf{z}_k)q(\mathbf{z}'|\mathbf{z}_k)} \cdot \frac{D(\mathbf{x}_k)^{-1} - 1}{D(\mathbf{x}')^{-1} - 1} \right), \quad (9)$$

i.e. let $\mathbf{x}_{k+1} = \mathbf{x}'$ if \mathbf{x}' is accepted and $\mathbf{x}_{k+1} = \mathbf{x}_k$ otherwise. Further assume the chain is irreducible, aperiodic and not transient. Then, according to the Metropolis-Hastings algorithm, the stationary distribution of this Markov chain is the data distribution $p_d(\mathbf{x})$ [6].

Proof. Note that similar to Eqn (8), we also have the change of variables between $p_g(\mathbf{x})$ and $p_0(\mathbf{z})$,

$$\log p_g(\mathbf{x})|_{\mathbf{x}=G(\mathbf{z})} = \log p_0(\mathbf{z}) - \frac{1}{2} \log \det J_{\mathbf{z}}^{\top} J_{\mathbf{z}}. \quad (10)$$

According to [6], the assumptions that the chain is irreducible, aperiodic, and not transient make sure that the chain has a unique stationary distribution, and the MH algorithm ensures that this stationary distribution equals to the target distribution $p_d(\mathbf{x})$. Thus we only need to show that the MH criterion in Eqn. (9) holds. Together with Eqn. (3), (7) and (8), we have

$$\begin{aligned} \alpha_{\text{REP}}(\mathbf{x}', \mathbf{x}_k) &= \frac{p_d(\mathbf{x}')q(\mathbf{x}_k|\mathbf{x}')}{p_d(\mathbf{x}_k)q(\mathbf{x}'|\mathbf{x}_k)} = \frac{p_d(\mathbf{x}')q(\mathbf{z}_k|\mathbf{z}')(\det J_{\mathbf{z}_k}^{\top} J_{\mathbf{z}_k})^{-\frac{1}{2}}p_g(\mathbf{x}_k)p_g(\mathbf{x}')}{p_d(\mathbf{x}_k)q(\mathbf{z}'|\mathbf{z}_k)(\det J_{\mathbf{z}'}^{\top} J_{\mathbf{z}'})^{-\frac{1}{2}}p_g(\mathbf{x}')p_g(\mathbf{x}_k)} \\ &= \frac{q(\mathbf{z}_k|\mathbf{z}')(\det J_{\mathbf{z}_k}^{\top} J_{\mathbf{z}_k})^{-\frac{1}{2}}p_0(\mathbf{z}')(\det J_{\mathbf{z}'}^{\top} J_{\mathbf{z}'})^{-\frac{1}{2}}(D(\mathbf{x}_k)^{-1} - 1)}{q(\mathbf{z}'|\mathbf{z}_k)(\det J_{\mathbf{z}'}^{\top} J_{\mathbf{z}'})^{-\frac{1}{2}}p_0(\mathbf{z}_k)(\det J_{\mathbf{z}_k}^{\top} J_{\mathbf{z}_k})^{-\frac{1}{2}}(D(\mathbf{x}')^{-1} - 1)} \\ &= \frac{p_0(\mathbf{z}')q(\mathbf{z}_k|\mathbf{z}')(\det J_{\mathbf{z}_k}^{\top} J_{\mathbf{z}_k})^{-\frac{1}{2}}}{p_0(\mathbf{z}_k)q(\mathbf{z}'|\mathbf{z}_k)(\det J_{\mathbf{z}'}^{\top} J_{\mathbf{z}'})^{-\frac{1}{2}}} \cdot \frac{(D(\mathbf{x}_k)^{-1} - 1)}{(D(\mathbf{x}')^{-1} - 1)}. \end{aligned} \quad (11)$$

Hence the proof is completed.

The theorem above demonstrates the following favorable properties of our method:

- The discriminator score ratio is the same as $\alpha_{\text{MH}}(\mathbf{x}', \mathbf{x}_k)$, but MH-GAN is restricted to a specific independent proposal. Our method instead works for any latent proposal $q(\mathbf{z}'|\mathbf{z}_k)$. When we take $q(\mathbf{z}'|\mathbf{z}_k) = p_0(\mathbf{z}')$, our method reduces to MH-GAN.

- Compared to $\alpha_{\text{DEP}}(\mathbf{x}', \mathbf{x}_k)$ of a general dependent proposal (Eqn. (7)), the unknown data distributions terms are successfully cancelled in the reparameterized acceptance criterion.
- The reparameterized MH acceptance criterion becomes tractable as it only involves the latent priors, the latent proposal distributions, and the discriminator scores.

Combining the REP proposal $q_{\text{REP}}(\mathbf{x}'|\mathbf{x}_k)$ and its tractable MH criterion $\alpha_{\text{REP}}(\mathbf{x}', \mathbf{x}_k)$, we have developed a novel sampling method for GANs, coined as REP-GAN. See Appendix 1 for a detailed description. Moreover, our method can serve as a general approximate inference technique for Bayesian models by bridging MCMC and GANs. Previous works [15,25,9] also propose to avoid the bad geometry of a complex probability measure by reparameterizing the Markov transitions into a simpler measure. However, these methods are limited to explicit invertible mappings without dimensionality reduction. In our work, we first show that it is also tractable to conduct such model-based reparameterization with implicit models like GANs.

4.3 A Practical Implementation

REP-GAN enables us to utilize the vast literature of existing MCMC algorithms [18] to design dependent proposals for GANs. We take Langevin Monte Carlo (LMC) as an example. As an Euler-Maruyama discretization of the Langevin dynamics, LMC updates the Markov chain with

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{\tau}{2} \nabla_{\mathbf{x}} \log p_t(\mathbf{x}_k) + \sqrt{\tau} \cdot \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad (12)$$

for a target distribution $p_t(\mathbf{x})$. Compared to MH-GAN, LMC utilizes the gradient information to explore the energy landscape more efficiently. However, if we directly take the (unknown) data distribution $p_d(\mathbf{x})$ as the target distribution $p_t(\mathbf{x})$, LMC does not meet a tractable update rule.

As discussed above, the reparameterization of REP-GAN makes it easier to design transitions in the low-dimensional latent space. Hence, we instead propose to use LMC for the latent Markov chain. We assume that the data distribution also lies in the low-dimensional manifold induced by the generator, i.e., $\text{Supp}(p_d) \subset \text{Im}(G)$. This implies that the data distribution $p_d(\mathbf{x})$ also has a pairing distribution in the latent space, denoted as $p_t(\mathbf{z})$. They are tied with the change of variables

$$\log p_d(\mathbf{x})|_{\mathbf{x}=G(\mathbf{z})} = \log p_t(\mathbf{z}) - \frac{1}{2} \log \det J_{\mathbf{z}}^{\top} J_{\mathbf{z}}, \quad (13)$$

Taking $p_t(\mathbf{z})$ as the (unknown) target distribution of the latent Markov chain, we have the following Latent LMC (L2MC) proposal

$$\begin{aligned}
\mathbf{z}' &= \mathbf{z}_k + \frac{\tau}{2} \nabla_{\mathbf{z}} \log p_t(\mathbf{z}_k) + \sqrt{\tau} \cdot \boldsymbol{\varepsilon} \\
&= \mathbf{z}_k + \frac{\tau}{2} \nabla_{\mathbf{z}} \log \frac{p_t(\mathbf{z}_k)}{p_0(\mathbf{z}_k)} \left(\det J_{\mathbf{z}_k}^\top J_{\mathbf{z}_k} \right)^{-\frac{1}{2}} + \frac{\tau}{2} \nabla_{\mathbf{z}} \log p_0(\mathbf{z}_k) + \sqrt{\tau} \cdot \boldsymbol{\varepsilon} \\
&= \mathbf{z} + \frac{\tau}{2} \nabla_{\mathbf{z}} \log \frac{p_d(\mathbf{x}_k)}{p_g(\mathbf{x}_k)} + \frac{\tau}{2} \nabla_{\mathbf{z}} \log p_0(\mathbf{z}_k) + \sqrt{\tau} \cdot \boldsymbol{\varepsilon} \\
&= \mathbf{z} - \frac{\tau}{2} \nabla_{\mathbf{z}} \log (D^{-1}(\mathbf{x}_k) - 1) + \frac{\tau}{2} \nabla_{\mathbf{z}} \log p_0(\mathbf{z}_k) + \sqrt{\tau} \cdot \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),
\end{aligned} \tag{14}$$

where $\mathbf{x}_k = G(\mathbf{z}_k)$. As we can see, L2MC is made tractable by our structured dependent proposal with pairing Markov chains. DDLS [5] proposes a similar Langevin proposal by formalizing GANs as an implicit energy-based model, while here we provide a straightforward derivation through reparameterization. Our major difference to DDLS is that REP-GAN also includes a tractable MH correction step (Eqn. (9)), which accounts for the numerical errors introduced by the discretization and ensures that detailed balance holds.

We give a detailed description of the algorithm procedure of our REP-GAN in Algorithm 1.

4.4 Extension to WGAN

Our method can also be extended to other kinds of GAN, like Wasserstein GAN (WGAN) [1]. The WGAN objective is

$$\min_G \max_D \mathbb{E}_{\mathbf{x} \sim p_d(\mathbf{x})}[D(\mathbf{x})] - \mathbb{E}_{\mathbf{x} \sim p_g(\mathbf{x})}[D(\mathbf{x})], \tag{15}$$

where $D : \mathcal{X} \rightarrow \mathbb{R}$ is restricted to be a Lipschitz function. Under certain conditions, WGAN also implies an approximate estimation of the density ratio [5],

$$D(\mathbf{x}) \approx \log \frac{p_d(\mathbf{x})}{p_g(\mathbf{x})} + \text{const} \Rightarrow \frac{p_d(\mathbf{x})}{p_g(\mathbf{x})} \approx \exp(D(\mathbf{x})) \cdot \text{const}. \tag{16}$$

Following the same derivations as in Eqn. (11) and (14), we will have the WGAN version of REP-GAN. Specifically, with $\mathbf{x}_k = G(\mathbf{z}_k)$, the L2MC proposal follows

$$\mathbf{z}' = \mathbf{z}_k + \frac{\tau}{2} \nabla_{\mathbf{z}} D(\mathbf{x}_k) + \frac{\tau}{2} \nabla_{\mathbf{z}} \log p_0(\mathbf{z}_k) + \sqrt{\tau} \cdot \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{17}$$

and the MH acceptance criterion is

$$\alpha_{REP-W}(\mathbf{x}', \mathbf{x}_k) = \min \left(1, \frac{q(\mathbf{z}_k | \mathbf{z}') p_0(\mathbf{z}')}{q(\mathbf{z}' | \mathbf{z}_k) p_0(\mathbf{z}_k)} \cdot \frac{\exp(D(\mathbf{x}'))}{\exp(D(\mathbf{x}_k))} \right). \tag{18}$$

Algorithm 1 GAN sampling with Reparameterized Markov chains (REP-GAN)

Input: trained GAN with (calibrated) discriminator D and generator G , Markov chain length K , latent prior distribution $p_0(\mathbf{z})$, latent proposal distribution $q(\mathbf{z}'|\mathbf{z}_k)$;

Output: an improved GAN sample \mathbf{x}_K ;

```

Draw an initial sample  $\mathbf{x}_1$ : 1) draw initial latent  $\mathbf{z}_1 \sim p_0(\mathbf{z})$  and 2) push forward
 $\mathbf{x}_1 = G(\mathbf{z}_1)$ ;
for each step  $k \in [1, K - 1]$  do
    Draw a REP proposal  $\mathbf{x}' \sim q_{\text{REP}}(\mathbf{x}'|\mathbf{x}_k)$ : 1) draw a latent proposal  $\mathbf{z}' \sim q(\mathbf{z}'|\mathbf{z}_k)$ ,
    and 2) push forward  $\mathbf{x}' = G(\mathbf{z}')$ ;
    Calculate the MH acceptance criterion  $\alpha_{\text{REP}}(\mathbf{x}_k, \mathbf{x}')$  following Eqn. (9);
    Decide the acceptance of  $\mathbf{x}'$  with probability  $\alpha_{\text{REP}}(\mathbf{x}_k, \mathbf{x}')$ ;
    if  $\mathbf{x}'$  is accepted then
        Let  $\mathbf{x}_{k+1} = \mathbf{x}'$ ,  $\mathbf{z}_{k+1} = \mathbf{z}'$ 
    else
        Let  $\mathbf{x}_{k+1} = \mathbf{x}_k$ ,  $\mathbf{z}_{k+1} = \mathbf{z}_k$ 
    end if
end for
```

5 Experiments

We evaluate our method on two synthetic datasets and two real-world image datasets as follows.

5.1 Manifold Dataset

Following DOT [24] and DDLS [5], we apply REP-GAN to the Swiss Roll dataset, where data samples lie on a Swiss roll manifold in the two-dimensional space. We construct the dataset by scikit-learn with 100,000 samples, and train a WGAN as in [24], where both the generator and discriminator are fully connected neural networks with leaky ReLU nonlinearities. We optimize the model using the Adam optimizer, with $\alpha = 0.0001$, $\beta_1 = 0.5$, $\beta_2 = 0.9$. After training, we draw 1,000 samples with different sampling methods. Following previous practice, we initialize a Markov chain with a GAN sample, run it for $K = 100$ steps, and collect the last example for evaluation.

As shown in Figure 2, with appropriate step size ($\tau = 0.01$), the gradient-based methods (DDLS and REP-GAN) outperform independent proposals (DRS and MH-GAN) by a large margin, while DDLS is more discontinuous on shape compared to REP-GAN. In DDLS, when the step size becomes too large ($\tau = 0.1, 1$), the numerical error of the Langevin dynamics becomes so large that the chain either collapses or diverges. In contrast, those bad proposals are rejected by the MH correction steps of REP-GAN, which prevents the misbehavior of the Markov chain.

5.2 Multi-modal Dataset

As GANs are known to suffer from the mode collapse problem, we also compare different GAN sampling methods in terms of modeling multi-modal distributions.

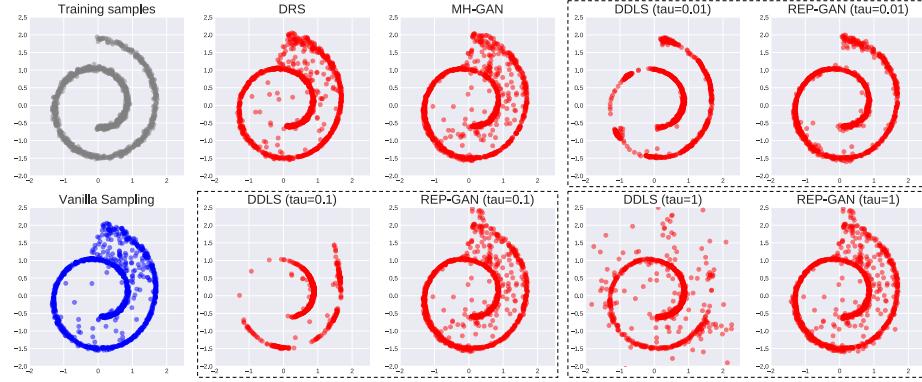


Fig. 2: Visualization of samples with different sampling methods on the Swiss Roll dataset. Here τ denotes the Langevin step size in Eqn. (17).

Specifically, we consider the 25-Gaussians dataset that is widely discussed in previous work [2,26,5]. The dataset is generated by a mixture of twenty-five two-dimensional isotropic Gaussian distributions with variance 0.01, and means separated by 1, arranged in a grid. We train a small GAN with the standard WGAN-GP objective following the setup in [24]. After training, we draw 1,000 samples with different sampling methods.

As shown in Figure 3, compared to MH-GAN, the gradient-based methods (DDLS and ours) produce much better samples close to the data distribution with proper step size ($\tau = 0.01$). Comparing DDLS and our REP-GAN, we can notice that DDLS tends to concentrate so much on the mode centers that its standard deviation can be even smaller than the data distribution. Instead, our method preserves more sample diversity while concentrating on the mode centers. This difference becomes more obvious as the step size τ becomes larger. When $\tau = 0.1$, as marked with blue circles, DDLS samples become so concentrated that some modes are even missed. When $\tau = 1$, DDLS samples diverge far beyond the 5×5 grid. In comparison, our REP-GAN is more stable because the MH correction steps account for the numerical errors caused by large τ .

These distinctions also become even more obvious when we scale to more modes. As shown in Figure 4, we also compare them w.r.t. mixture of Gaussians with 9×9 and 13×13 modes, respectively. Under the more challenging scenarios, we can see that the gradient-based methods still consistently outperforms MH-GAN. Besides, our REP-GAN has a more clear advantage over DDLS. Specifically, for 9×9 modes, our REP-GAN produces samples that are less noisy, while preserving all the modes. For 9×9 modes, DDLS makes a critical mistake that it drops

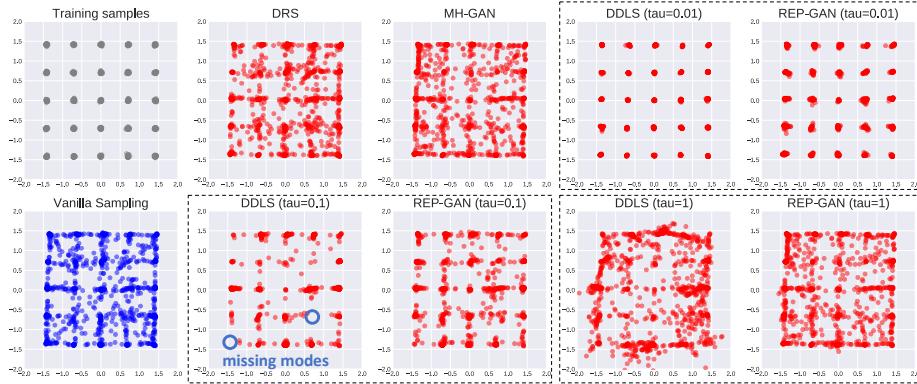


Fig. 3: Visualization of samples with different sampling methods on the 25-Gaussians dataset. Here τ denotes the Langevin step size in Eqn. (17).

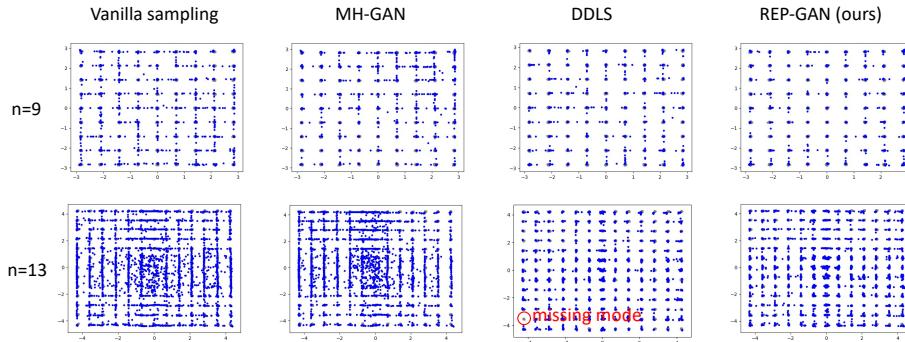


Fig. 4: Visualization of the mixture-of-Gaussian experiments with 9x9 (1st row) and 13x13 (2nd row) modes with proper step size $\tau = 0.01$. True data points are shown in grey (in background), and generated points are shown in blue.

one of the modes. As discussed above, we believe this is because DDLS has a bias towards regions with high probability, while ignoring the diversity of the distribution. In comparison, REP-GAN effectively prevents such bias by the MH correction steps.

5.3 Real-world Image Dataset

Following MH-GAN [26], we conduct experiments on two real-world image datasets, CIFAR-10 and CelebA, for two models, DCGAN [19] and WGAN [1]. Following the conventional evaluation protocol, we initialize each Markov chain with a GAN sample, run it for 640 steps, and take the last sample for

Table 2: Inception Scores of different sampling methods on CIFAR-10 and CelebA, with the DCGAN and WGAN backbones.

Method	CIFAR-10		CelebA	
	DCGAN	WGAN	DCGAN	WGAN
GAN	3.219	3.740	2.332	2.788
DRS [2]	3.073	3.137	2.869	2.861
MH-GAN [26]	3.225	3.851	3.106	2.889
DDLS [5]	3.152	3.547	2.534	2.862
REP-GAN (ours)	3.541	4.035	2.686	2.943

Table 3: Average Inception Score (a) and acceptance ratio (b) vs. training epochs with DCGAN on CIFAR-10.

(a) Inception Score (mean \pm std)					
Epoch	20	21	22	23	24
GAN	2.482 ± 0.027	3.836 ± 0.046	3.154 ± 0.014	3.383 ± 0.046	3.219 ± 0.036
MH-GAN	2.356 ± 0.023	3.891 ± 0.040	3.278 ± 0.033	3.458 ± 0.029	3.225 ± 0.029
DDLS	2.419 ± 0.021	3.332 ± 0.025	2.996 ± 0.035	3.255 ± 0.045	3.152 ± 0.028
REP-GAN	$\mathbf{2.487} \pm 0.019$	$\mathbf{3.954} \pm 0.046$	$\mathbf{3.294} \pm 0.030$	$\mathbf{3.534} \pm 0.035$	$\mathbf{3.541} \pm 0.038$

(b) Average Acceptance Ratio (mean \pm std)					
Epoch	20	21	22	23	24
MH-GAN	0.028 ± 0.143	0.053 ± 0.188	0.060 ± 0.199	0.021 ± 0.126	0.027 ± 0.141
REP-GAN	$\mathbf{0.435} \pm 0.384$	$\mathbf{0.350} \pm 0.380$	$\mathbf{0.287} \pm 0.365$	$\mathbf{0.208} \pm 0.335$	$\mathbf{0.471} \pm 0.384$

evaluation. We collect 50,000 samples to evaluate the Inception Score¹ [20]. The step size τ of our L2MC proposal is 0.01 on CIFAR-10 and 0.1 on CelebA. We calibrate the discriminator with Logistic Regression as in [26].

From Table 2, we can see our method outperforms the state-of-the-art sampling methods in most cases. In Table 3, we also present the average Inception Score and acceptance ratio during the training process. As shown in Table 3a, our REP-GAN can still outperform previous sampling methods consistently and significantly. Besides, in Table 3b, we find that the average acceptance ratio of MH-GAN is lower than 0.05 in most cases, which is extremely low. While with our reparameterized dependent proposal, REP-GAN achieves an acceptance ratio around 0.4, which is known to be a relatively good tradeoff for MCMC algorithms.

¹ For fair comparison, our training and evaluation follows the the official code of MH-GAN [26]: <https://github.com/uber-research/metropolis-hastings-gans>

Table 4: Ablation study of our REP-GAN with Inception Scores (IS) and acceptance ratios on CIFAR-10 with two backbone models, DCGAN and WGAN.

Method	DCGAN		WGAN	
	Accept Ratio	IS	Accept Ratio	IS
REP-GAN	0.447 ± 0.384	3.541 ± 0.038	0.205 ± 0.330	4.035 ± 0.036
REP-GAN w/o REP proposal	0.027 ± 0.141	3.225 ± 0.029	0.027 ± 0.141	3.851 ± 0.044
REP-GAN w/o MH rejection	-	3.152 ± 0.028	-	3.547 ± 0.029

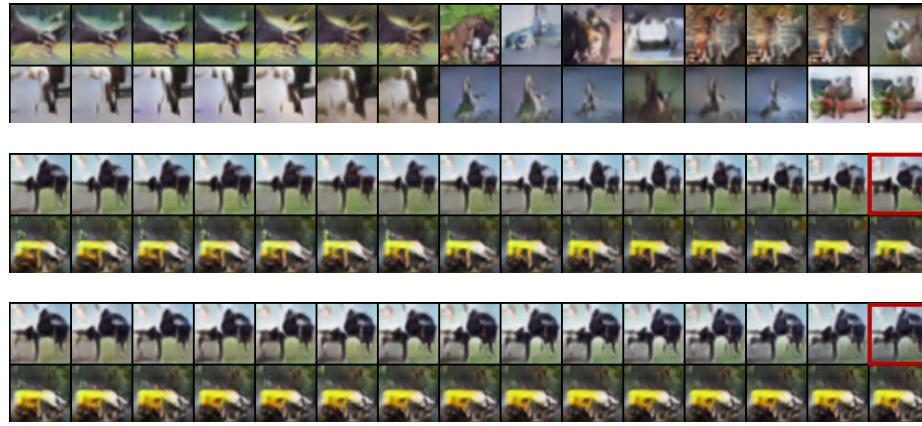


Fig. 5: The first 15 steps of two Markov chains with the same initial samples, generated by MH-GAN (top), DDLS (middle), and REP-GAN (bottom).

5.4 Algorithmic Analysis

Ablation Study We conduct an ablation study of the proposed sampling algorithm, REP-GAN, and the results are shown in Table 4. We can see that without our proposed reparameterized (REP) proposal, the acceptance ratio is very small (with an independent proposal instead). Consequently, the sample quality degrades significantly. Also, we can find that the MH correction step also matters a lot, without which the sample quality of Langevin sampling becomes even worse than the independent proposal. The ablation study shows the necessity of both REP proposal and MH rejection steps in the design of our REP-GAN.

Markov Chain Visualization In Figure 5, we demonstrate two Markov chains sampled with different methods. We can see that MH-GAN is often trapped in the same place because of the independent proposals. DDLS and REP-GAN instead gradually refine the samples with gradient steps. In addition, compared to the gradient-based methods, we can see that the MH rejection steps of REP-GAN help avoid some bad artifacts in the images. For example, in the camel-like images marked in red, the body of the camel is separated in the sample of DDLS (middle) while it is not in the sample of REP-GAN (bottom). Note that, the evaluation

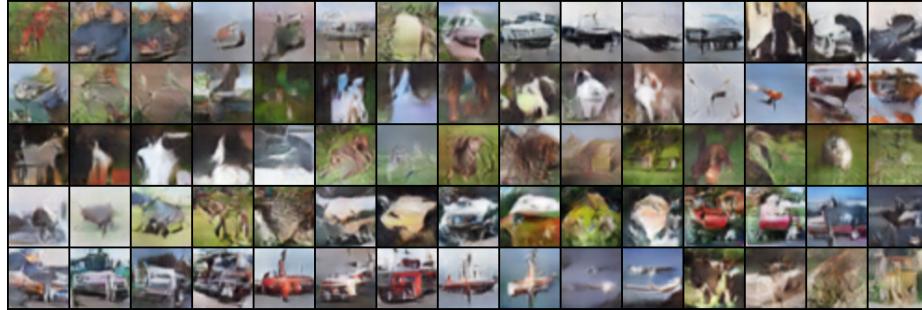


Fig. 6: Visualization of 5 Markov chains of our REP proposals (i.e., REP-GAN without the MH rejection steps) with a large step size ($\tau = 1$).

protocol only needs the last step of the chain, thus we prefer a small step size that finetunes the initial samples for better sample quality. As shown in Figure 6, our REP proposal can also produce very diverse images with a large step size.

Computation Overhead We also compare the computation cost of the gradient-based sampling methods, DDLS and REP-GAN. They take 88.94s and 88.85s, respectively, hence the difference is negligible. Without the MH-step, our method takes 87.62s, meaning that the additional MH-step only costs 1.4% computation overhead, which is also negligible, but it brings a significant improvement of sample quality as shown in Table 4.

6 Conclusion

In this paper, we have proposed a novel method, REP-GAN, to improve the sampling of GAN. We devise a structured dependent proposal that reparameterizes the sample-level transition of GAN into the latent-level transition. More importantly, we first prove that this general proposal admits a tractable MH criterion. Experiments show our method does not only improve sample efficiency but also demonstrate state-of-the-art sample quality on benchmark datasets over existing sampling methods.

References

1. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein GAN. ICML (2017)
2. Azadi, S., Olsson, C., Darrell, T., Goodfellow, I., Odena, A.: Discriminator rejection sampling. ICLR (2019)
3. Ben-Israel, A.: The change-of-variables formula using matrix volume. SIAM Journal on Matrix Analysis and Applications **21**(1), 300–312 (1999)
4. Brock, A., Donahue, J., Simonyan, K.: Large scale GAN training for high fidelity natural image synthesis. ICLR (2019)
5. Che, T., Zhang, R., Sohl-Dickstein, J., Larochelle, H., Paull, L., Cao, Y., Bengio, Y.: Your GAN is secretly an energy-based model and you should use discriminator driven latent sampling. ICML (2020)
6. Gelman, A., Carlin, J.B., Stern, H.S., Dunson, D.B., Vehtari, A., Rubin, D.B.: Bayesian data analysis. CRC press (2013)
7. Gemici, M.C., Rezende, D., Mohamed, S.: Normalizing flows on Riemannian manifolds. arXiv preprint arXiv:1611.02304 (2016)
8. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. NeurIPS (2014)
9. Hoffman, M., Sountsov, P., Dillon, J.V., Langmore, I., Tran, D., Vasudevan, S.: Neutralizing bad geometry in Hamiltonian Monte Carlo using neural transport. arXiv preprint arXiv:1903.03704 (2019)
10. Hoffman, M.D.: Learning deep latent gaussian models with markov chain monte carlo. In: ICML (2017)
11. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. NeurIPS (2019)
12. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. ICLR (2014)
13. Levy, D., Hoffman, M.D., Sohl-Dickstein, J.: Generalizing Hamiltonian Monte Carlo with neural networks. ICLR (2018)
14. Li, Y., Turner, R.E., Liu, Q.: Approximate inference with amortised mcmc. arXiv preprint arXiv:1702.08343 (2017)
15. Marzouk, Y., Moseley, T., Parno, M., Spantini, A.: An introduction to sampling via measure transport. arXiv preprint arXiv:1602.05023 (2016)
16. Metz, L., Poole, B., Pfau, D., Sohl-Dickstein, J.: Unrolled generative adversarial networks. ICLR (2017)
17. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. ICLR (2018)
18. Neal, R.M., et al.: MCMC using Hamiltonian dynamics. Handbook of markov chain monte carlo **54**, 113–162 (2010)
19. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)
20. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training GANs. NeurIPS (2016)
21. Salimans, T., Kingma, D., Welling, M.: Markov chain monte carlo and variational inference: Bridging the gap. In: ICML (2015)
22. Song, J., Zhao, S., Ermon, S.: A-NICE-MC: Adversarial training for MCMC. NeurIPS (2017)
23. Song, Y., Ye, Q., Xu, M., Liu, T.Y.: Discriminator contrastive divergence: semi-amortized generative modeling by exploring energy of the discriminator. arXiv preprint arXiv:2004.01704 (2020)

24. Tanaka, A.: Discriminator optimal transport. NeurIPS (2019)
25. Titsias, M.K.: Learning model reparametrizations: implicit variational inference by fitting MCMC distributions. arXiv preprint arXiv:1708.01529 (2017)
26. Turner, R., Hung, J., Saatci, Y., Yosinski, J.: Metropolis-Hastings generative adversarial networks. ICML (2019)
27. Wang, T., Wu, Y., Moore, D., Russell, S.J.: Meta-learning mcmc proposals. In: NeurIPS (2018)