BOOTSTRAPPING GENERATIVE ADVERSARIAL NETWORKS WITH COUPLING MARKOV CHAINS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recently, sampling methods have been successfully applied to bootstrap the sample quality of GAN. However, in practice, they typically have poor sample efficiency because of the independent proposals. In this work, we propose a novel MCMC sampling method, CMC-GAN, that allows dependent proposals reparameterized through a coupling Markov chain in the latent space. We prove that our structured proposal admits tractable Metropolis-Hastings acceptance criterion. Empirical results show our CMC-GAN has largely improved the average acceptance ratio and obtained better Inception Scores on CIFAR10.

1 INTRODUCTION

Generative Adversarial Networks (GAN) Goodfellow et al. (2014) have achieved great success at generating realistic images in recent years (Karras et al., 2019; Brock et al., 2018). Unlike previous generative models that explicitly parameterize the data distribution, GAN relies on an alternative optimization between a generator and a discriminator to implicitly learn the data distribution.

As is pointed out by Goodfellow et al. (2014), the discriminator of GAN is essentially a density ratio estimator. With an optimal discriminator, the generative distribution is pushed towards the data distribution by minimizing their Jensen-Shannon (JS) divergence. However, in practice, image samples from the learned generator still suffer from problems such as mode collapse and bad artifacts. Various variants have been proposed to alleviate these issues (Lucic et al., 2018).

Recently, sampling methods have shown promising results in boosting the sample quality of GAN by turning waste into treasure. Typically, after training, the discriminator is discarded, and only the generator is left for image generation. Instead, the sampling methods utilize the density ratio information in the discriminator to further bridge the generative and data distributions. Specifically, DRS (Azadi et al., 2019) applies rejection sampling, while MH-GAN (Turner et al., 2019) uses MCMC, which is more scalable for high-dimensional data. They have achieved up to 33% improvement over the original GAN in terms of inception score. Nevertheless, these methods suffer a lot from the sample efficiency problem. For example, as shown in Section 3.2, the average acceptance ratio of MH-GAN can be lower than 5%. Hence the Markov chains need a very long time to mix.

In fact, this is because MH-GAN adopts an *independent* proposal, i.e., $q(\mathbf{x}'|\mathbf{x}) = q(\mathbf{x}')$. Because the proposal is independent of the previous step, the difference between samples can be so large that the proposal gets rejected easily. In view of this limitation, we propose to generalize the independent proposal to a general *dependent* proposal $q(\mathbf{x}'|\mathbf{x})$. In such a way, the proposed sample is close to the previous one, which leads to a higher acceptance ratio and better sample efficiency. We can also balance between the exploration and exploitation of the Markov chains by choice of 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 in practice. Nevertheless, we know that the generative distribution $p_g(\mathbf{x})$ is implicitly defined as the push forward of the latent prior distribution $p_0(\mathbf{z})$, and it is generally much easier to design proposals in the low dimensional latent space $\mathcal{Z}$.

Hence, GAN's latent variable structure motivates us to design a *structured proposal* with two coupling Markov chains, one in the sample space $\mathcal{X}$ and one in the latent space $\mathcal{Z}$. As shown in Figure 1, given the current coupling samples $(\mathbf{z}_k, \mathbf{x}_k)$, we draw the next proposal $\mathbf{x}'$ in an bottom-to-up way. We first draw $\mathbf{z}'$ following the latent proposal distribution $q(\mathbf{z}'|\mathbf{z})$. Then we push it forward through

Table 1: Comparison of sampling methods for GAN. The table lists three efficient mechanisms for the effective sampling of GAN. Our CMC-GAN includes all three mechanisms and takes the other methods as our ablated special cases.

Method	**Rejection step**	**Markov chain**	**Latent gradient proposal**
Baseline	✗	✗	✗
DRS (Azadi et al., 2019)	✓	✗	✗
MH-GAN (Turner et al., 2019)	✓	✓	✗
DDLS (Che et al., 2020)	✗	✓	✓
CMC-GAN (ours)	✓	✓	✓

the generator and get the sample proposal $\mathbf{x}'$. In this way, by utilizing the underlying structure of GAN, the sampling becomes more efficient with a simpler energy landscape in the latent space.

Another major contribution of this work is that we first prove that this general structured proposal of GAN admits a closed-form Metropolis-Hastings (MH) acceptance criterion. When the latent Markov chain adopts an independent proposal, our MH formula reduces to that of MH-GAN (Turner et al., 2019) as a special case. When we take a Langevin latent proposal and omits the MH test, our method then reduces to DDLS (Che et al., 2020) as another corner case. As a result, our methods are more sample efficient than MH-GAN because of the dependent proposal, and more unbiased than DDLS Che et al. (2020) with the MH test for bias correction. See Table 1 for a clear comparison. Empirical results also demonstrate our proposed CMC-GAN achieves better image quality and sample efficiency than previous methods.

2 Related Work

Although GAN is able to synthesis high-quality images, the minimax nature of GAN makes it quite unstable and leads to mode collapse. A series of works like DCGAN (Radford et al., 2015) and SNGAN (Miyato et al., 2018) improve the sample quality by devising new architectures. Methods like Unrolled GAN (Metz et al., 2016), PACGAN Lin et al. (2018), try to fix mode collapse with new training mechanisms. Other methods like WGAN Arjovsky et al. (2017), WGAN-GP Gulrajani et al. (2017) instead propose alternative objectives for better training stability.

DRS (Azadi et al., 2019) firstly proposes to use discriminator to improve sample quality by rejection sampling. MH-GAN (Turner et al., 2019) instead uses the Metropolis-Hasting algorithm with an independent MCMC proposal. DDLS Che et al. (2020) and DCD (Song et al., 2020) apply the gradient-based proposal, Langevin Monte Carlo, by viewing GAN as an energy-based model. Tanaka (2019) proposes a similar gradient-based method named DOT from the perspective of optimal transport.

In fact, GAN can be considered as a special kind of variational inference (Hu et al., 2017). The marriage between variational inference and sampling methods has a long history in learning probabilistic models, to list a few, Monte Carlo EM (Levine & Casella, 2001), Contrastive Divergence (Hinton, 2002), etc. In the era of deep learning, Salimans et al. (2015); Hoffman (2017) develop methods to leverage MCMC to improve the posterior of VAE (Kingma & Welling, 2013), while Song et al. (2017); Levy et al. (2017) adopt neural networks for better proposals in MCMC.

3 Background

In this section, we briefly introduce the notations and backgrounds.

3.1 GAN

GAN models the data distribution $p_d(\mathbf{x})$ implicitly with a generator $G : \mathcal{Z} \to \mathcal{X}$ ($|\mathcal{Z}| < |\mathcal{X}|$),

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

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

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

As is pointed out by Goodfellow et al. (2014), an optimal discriminator D implies the density ratio between the data and generative distribution as follows

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

3.2 MCMC

Markov Chain Monte Carlo (MCMC) refers to a kind of sampling methods that draw a chain of samples $\mathbf{x}_{1:K} \in \mathcal{X}^K$ that marginally comes 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 Metropolis-Hastings algorithm, we accept the proposal $\mathbf{x}' \sim q(\mathbf{x}'|\mathbf{x}_k)$ with probability

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

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

4 METHOD

For GAN, typically, the discriminator is thrown away after training, and only the generator is kept for generating samples. The idea behind sampling methods is to utilize the density ratio information in the discriminator to further bootstrap the generative distribution. In this section, we first review MH-GAN and point out the limitations. We then propose our structured proposal to overcome these obstacles, and discuss its theoretical properties as well as practical implementations.

4.1 FROM INDEPENDENT PROPOSAL TO DEPENDENT PROPOSAL

MH-GAN (Turner et al., 2019) 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

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

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

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

Because $\alpha_{\mathrm{MH}}(\mathbf{x}', \mathbf{x}_k)$ only involves the discriminator scores, it is computationally tractable. However, to achieve the tractability, MH-GAN adopts an independent proposal $q(\mathbf{x}')$ with poor sample efficiency. Because 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 to the sample efficiency problem of MH-GAN is to take a *dependent* proposal $q(\mathbf{x}'|\mathbf{x})$ that will propose a $\mathbf{x}'$ close to the current sample $\mathbf{x}_k$. Such proposals are more likely to be accepted. Nevertheless, the problem of such a dependent proposal is that its MH acceptance criterion $\alpha_{\mathrm{DEP}}(\mathbf{x}', \mathbf{x}_k)$ is generally intractable:

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

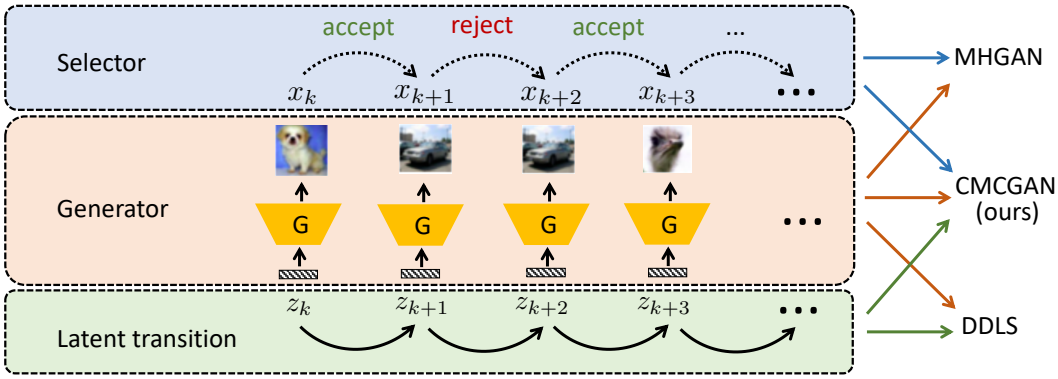

Figure 1: Illustration of MCMC transitions with our CMC-GAN's structured proposal with two coupling Markov chains in the latent space $\mathcal{Z}$ and the sample space $\mathcal{X}$. In comparison, MH-GAN (Turner et al., 2019) and DDLS (Che et al., 2020) only include part of our three components.

First, we do not have access to the explicit data density $p_d(\mathbf{x})$. And second, as the sample space $\mathcal{X}$ is usually high-dimensional with complex energy landscape, it is also hard to design a proper dependent proposal $q(\mathbf{x}'|\mathbf{x}_k)$ with explicit density (see Section 4.4). This obstacle holds us back from adopting a dependent proposal which is more suitable for MCMC.

4.2 A Tractable Structured Proposal with Coupling Markov Chains

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

As we know, samples of GAN lie in a low-dimensional manifold induced by the push-forward of the latent variable. Suppose at the step k of the Markov chain, we have a GAN sample $\mathbf{x}$ with latent $\mathbf{z}$. Instead of proposing a new sample $\mathbf{x}'$ directly from a sample-level proposal $q(\mathbf{x}'|\mathbf{x})$, we first propose a latent $\mathbf{z}'$ from the current latent $\mathbf{z}$ following a latent proposal distribution $q(\mathbf{z}'|\mathbf{z})$, which is much easier to design in the low dimensional latent space $\mathcal{Z}$. Afterwards, we push the latent $\mathbf{z}'$ forward through the generator and get the output $\mathbf{x}'$ as our sample proposal. In summary we have

$$1)\, \mathbf{x}_k = G(\mathbf{z}_k);\;\; 2)\, \mathbf{z}' \sim q(\mathbf{z}'|\mathbf{z});\;\; 3)\, \mathbf{x}' = G(\mathbf{z}'). \tag{8}$$

As illustrated in Figure 1, our bottom-to-up proposal relies on the interplay between two coupling Markov chains, one in the sample space $\mathcal{X}$, and one in the latent space $\mathcal{Z}$. Hence we call it a CMC (Coupling Markov Chain) proposal $q_{\mathrm{CMC}}(\mathbf{x}'|\mathbf{x})$. In fact, under mild conditions, the CMC prposal distribution $q_{\mathrm{CMC}}(\mathbf{x}'|\mathbf{x}_k)$ and the latent proposal distribution $q(\mathbf{z}'|\mathbf{z}_k)$ are tied with the change of variables as follows (Gemici et al., 2016; Ben-Israel, 1999)

$$\log q_{\mathrm{CMC}}(\mathbf{x}'|\mathbf{x}) = \log q(\mathbf{x}'|\mathbf{z}_k) = \log q(\mathbf{z}'|\mathbf{z}) - \frac{1}{2}\log\det J_{\mathbf{z}'}^{\top} J_{\mathbf{z}'}, \tag{9}$$

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

Nevertheless, it remains unclear whether we can perform the MH test to decide the acceptance of the proposal $\mathbf{x}'$. Recall that a general dependent proposal distribution does not admit a tractable MH acceptance criterion (Eqn. 7). Perhaps surprisingly, another favorable property of our CMC proposal is that its MH acceptance criterion is tractable for a general latent proposal $q(\mathbf{z}'|\mathbf{z})$.

Theorem 1. *Consider a Markov chain of GAN samples $\mathbf{x}_{1:K}$ with initial distribution $p_0(\mathbf{x})$, For step $k+1$, we accept our CMC proposal $\mathbf{x}' \sim q_{\mathrm{CMC}}(\mathbf{x}'|\mathbf{x}_k)$ with probability*

$$\alpha_{\mathrm{CMC}}\left(\mathbf{x}', \mathbf{x}_k\right) = \min\left(1, \frac{p_0(\mathbf{z}')q(\mathbf{z}_k|\mathbf{z}')}{p_0(\mathbf{z}_k)q(\mathbf{z}'|\mathbf{z}_k)} \cdot \frac{D(\mathbf{x}_k)^{-1} - 1}{D(\mathbf{x}')^{-1} - 1}\right), \tag{10}$$

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 and aperiodic. Then, according to the Metropolis-Hastings algorithm, the stationary distribution of this Markov chain is the data distribution $p_d(\mathbf{x})$.

Proof. Note that similar to Eqn 9, we also have the change of variables between $p_g(\mathbf{x})$ and $p_0(\mathbf{z})$,

$$\log p_g(\mathbf{x}) = \log p_0(\mathbf{z}) - \frac{1}{2}\log\det J_{\mathbf{z}}^{\top} J_{\mathbf{z}}. \tag{11}$$

We only need to prove Eqn. 10 holds. Together with Eqn. 3, 7, 9, we have

$$\begin{aligned}
\alpha_{\mathrm{CMC}}(\mathbf{x}', \mathbf{x}_k) &= \frac{p_d\left(\mathbf{x}'\right) q\left(\mathbf{x}_k|\mathbf{x}'\right)}{p_d\left(\mathbf{x}_k\right) q\left(\mathbf{x}'|\mathbf{x}_k\right)} = \frac{\cancel{p_d\left(\mathbf{x}'\right)} q(\mathbf{z}_k|\mathbf{z}')\left(\det J_{\mathbf{z}_k}^{\top} J_{\mathbf{z}_k}\right)^{-\frac{1}{2}} \cancel{p_g(\mathbf{x}_k)} p_g(\mathbf{x}')}{\cancel{p_d\left(\mathbf{x}_k\right)} q(\mathbf{z}'|\mathbf{z}_k)\left(\det J_{\mathbf{z}'}^{\top} J_{\mathbf{z}'}\right)^{-\frac{1}{2}} \cancel{p_g(\mathbf{x}')}} \\
&= \frac{q(\mathbf{z}_k|\mathbf{z}')\cancel{\left(\det J_{\mathbf{z}_k}^{\top} J_{\mathbf{z}_k}\right)^{-\frac{1}{2}}} p_0(\mathbf{z}')\cancel{\left(\det J_{\mathbf{z}'}^{\top} J_{\mathbf{z}'}\right)^{-\frac{1}{2}}}(D(\mathbf{x}_k)^{-1}-1)}{q(\mathbf{z}'|\mathbf{z}_k)\cancel{\left(\det J_{\mathbf{z}'}^{\top} J_{\mathbf{z}'}\right)^{-\frac{1}{2}}} p_0(\mathbf{z}_k)\cancel{\left(\det J_{\mathbf{z}_k}^{\top} J_{\mathbf{z}_k}\right)^{-\frac{1}{2}}}(D(\mathbf{x}')^{-1}-1)} \\
&= \frac{p_0(\mathbf{z}')q(\mathbf{z}_k|\mathbf{z}')(D(\mathbf{x}_k)^{-1}-1)}{p_0(\mathbf{z}_k)q(\mathbf{z}'|\mathbf{z}_k)(D(\mathbf{x}')^{-1}-1)}.
\end{aligned} \tag{12}$$

Hence the proof is completed. □

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

- The discriminator score ratio is the same as $\alpha_{\mathrm{MH}}(\mathbf{x}', \mathbf{x}_k)$, but MH-GAN is restricted to a specific independent proposal. Our method instead works for any latent proposal $q(\mathbf{z}'|\mathbf{z})$. When we take $q(\mathbf{z}'|\mathbf{z}) = p_0(\mathbf{z}')$, our method reduces to MH-GAN as a special case.
- Compared to $\alpha_{DEP}(\mathbf{x}', \mathbf{x}_k)$ of a general dependent proposal, the unknown data distributions terms are canceled in the reparameterized acceptance criterion.
- Although we adopt a dependent proposal, the unknown Jacobian terms are successfully canceled. The reparameterized MH acceptance criterion becomes tractable as it only involves the latent priors, the latent proposal distribution, and the discriminator scores.

Combining the CMC proposal $q_{\mathrm{CMC}}(\mathbf{x}'|\mathbf{x})$ and its tractable MH acceptance criterion $\alpha_{CMC}(\mathbf{x}', \mathbf{x}_k)$, we have developed a novel sampling method for bootstrapping GAN, coined as CMC-GAN.

4.3 Transition Reparameterization

According to the law of the unconscious statistician, our Markov Chain in the sample space $\mathcal{X}$ can be reparameterized as a latent Markov Chain followed by push-forward at each step. Specifically, given a coupling pair $(\mathbf{z}, \mathbf{x})$ in the two chains, the transition $\mathbf{x} \to \mathbf{x}'$ following $q(\mathbf{x}'|\mathbf{x})$ is equivalent to that we firstly transit from $\mathbf{z}$ to $\mathbf{z}'$ in the latent chain following $q(\mathbf{z}'|\mathbf{z})$, and then push it forward and get $\mathbf{x}' = G(\mathbf{z}')$ through the generator G.

In this way, through a learned generator, we transport the transition in high dimensional space $\mathcal{X}$ into low dimensional space $\mathcal{Z}$, which enjoys much better geometry and makes it easier to design proposals in MCMC algorithms. Previous works (Titsias, 2017; Hoffman et al., 2019) also develop similar sampling methods with invertible models that have explicit density, like flows (Dinh et al., 2016; Kingma & Dhariwal, 2018). In this work, we first prove that it is also tractable to conduct measure transport (Marzouk et al., 2016) for implicit models like GAN.

4.4 Latent Langevin Monte Carlo Proposal

CMC-GAN enables us to utilize the vast literature of existing MCMC algorithms to design proposals Neal et al. (2010). We take Langevin Monte Carlo (LMC) as an example. As an Euler-Maruyama discretization of the Langevin dynamics, LMC updates the Markov chain with

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{\alpha}{2}\nabla_{\mathbf{x}} \log p_t(\mathbf{x}_k) + \sqrt{\alpha}\cdot\boldsymbol{\varepsilon}, \;\; \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{13}$$

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 admit a tractable update rule.

As is discussed above, the reparameterization of CMC-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., $\mathrm{Supp}\,(p_d) \subset \mathrm{Im}(G)$. This implies that the data distribution $p_d(\mathbf{x})$ also has a coupling distribution in the latent space, denoted as $p_t(\mathbf{z})$. They are tied with the change of variables

$$\log p_d(\mathbf{x}) = \log p_t(\mathbf{z}) - \frac{1}{2}\log\det\, J_{\mathbf{z}}^\top J_{\mathbf{z}}, \tag{14}$$

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

$$\begin{aligned}
\mathbf{z}' &= \mathbf{z} + \frac{\alpha}{2}\nabla_{\mathbf{z}}\log p_t(\mathbf{z}) + \sqrt{\alpha}\cdot\boldsymbol{\varepsilon} \\
&= \mathbf{z} + \frac{\alpha}{2}\nabla_{\mathbf{z}}\log\frac{p_t(\mathbf{z})\left(\det J_{\mathbf{z}}^\top J_{\mathbf{z}}\right)^{-\frac{1}{2}}}{p_0(\mathbf{z})\left(\det J_{\mathbf{z}}^\top J_{\mathbf{z}}\right)^{-\frac{1}{2}}} + \frac{\alpha}{2}\nabla_{\mathbf{z}}\log p_0(\mathbf{z}) + \sqrt{\alpha}\cdot\boldsymbol{\varepsilon} \\
&= \mathbf{z} + \frac{\alpha}{2}\nabla_{\mathbf{z}}\log\frac{p_d(\mathbf{x})}{p_g(\mathbf{x})} + \frac{\alpha}{2}\nabla_{\mathbf{z}}\log p_0(\mathbf{z}) + \sqrt{\alpha}\cdot\boldsymbol{\varepsilon} \\
&= \mathbf{z} - \frac{\alpha}{2}\nabla_{\mathbf{z}}\log(D^{-1}(\mathbf{x}) - 1) + \frac{\alpha}{2}\nabla_{\mathbf{z}}\log p_0(\mathbf{z}) + \sqrt{\alpha}\cdot\boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I}),
\end{aligned} \tag{15}$$

where $\mathbf{x} = G(\mathbf{z})$. As we can see, L2MC is made tractable by our structured proposal with coupling Markov chains. In DDLS (Che et al., 2020), the authors propose the same algorithm by formalizing GAN as an implicit energy-based model, while here we provide a straightforward derivation. Our major difference to DDLS is that CMC-GAN also includes a tractable MH correction step (Eqn. 10), which accounts for the numerical errors introduced by the discretization and ensures that detailed balance holds. Experiments show it is crucial for the final performance.

4.5 Extended to WGAN

In fact, our method can also be extended to other kinds of GAN, like Wasserstein GAN (WGAN) (Arjovsky et al., 2017). The WGAN objective is

$$\min_G \max_D\ \mathbb{E}_{p_d}[D(x)] - \mathbb{E}_{p_g}[D(x)] \tag{16}$$

where $D : \mathcal{X} \to \mathbb{R}$ is restricted to be a K-Lipschitz function. Under certain conditions, WGAN also implies an approximate estimation of the density ratio (Che et al., 2020),

$$D(\mathbf{x}) \approx \log\frac{p_d(\mathbf{x})}{p_g(\mathbf{x})} + c \quad\Rightarrow\quad \frac{p_d(\mathbf{x})}{p_g(\mathbf{x})} \approx \exp(D(x))\cdot e^{-c}, \tag{17}$$

where c is a constant. Following the same derivations as in Eqn. 12 & 15, we will have the WGAN version of CMC-GAN. Specifically, the Langevin CMC proposal follows

$$\mathbf{z}' = \mathbf{z} + \alpha\nabla_{\mathbf{z}}D(\mathbf{x}) + \alpha\nabla_{\mathbf{z}}\log p_0(\mathbf{z}) + \eta\cdot\boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I}), \tag{18}$$

and the MH acceptance criterion is

$$\alpha_{CMC-W}\left(\mathbf{x}', \mathbf{x}_k\right) = \min\left(1,\ \frac{q(\mathbf{z}_k|\mathbf{z}')p_0(\mathbf{z}')}{q(\mathbf{z}'|\mathbf{z}_k)p_0(\mathbf{z}_k)}\cdot\frac{D(\mathbf{x}')}{D(\mathbf{x}_k)}\right). \tag{19}$$

5 Experiments

We show our empirical results on a synthetic dataset and a real image dataset.

5.1 Synthetic Data

Following the practice in Tanaka (2019); Che et al. (2020), we apply CMC-GAN to the synthetic Swiss Roll dataset, where data samples lie on a Swiss roll manifold in the two-dimensional space. We construct the dataset with scikit-learn and train a WGAN as in Tanaka (2019). Afterward, we sample 5,000 samples from the model with different sampling methods and visualize them in Figure 2. We can see that our CMC-GAN has better sample quality and numerical stability.

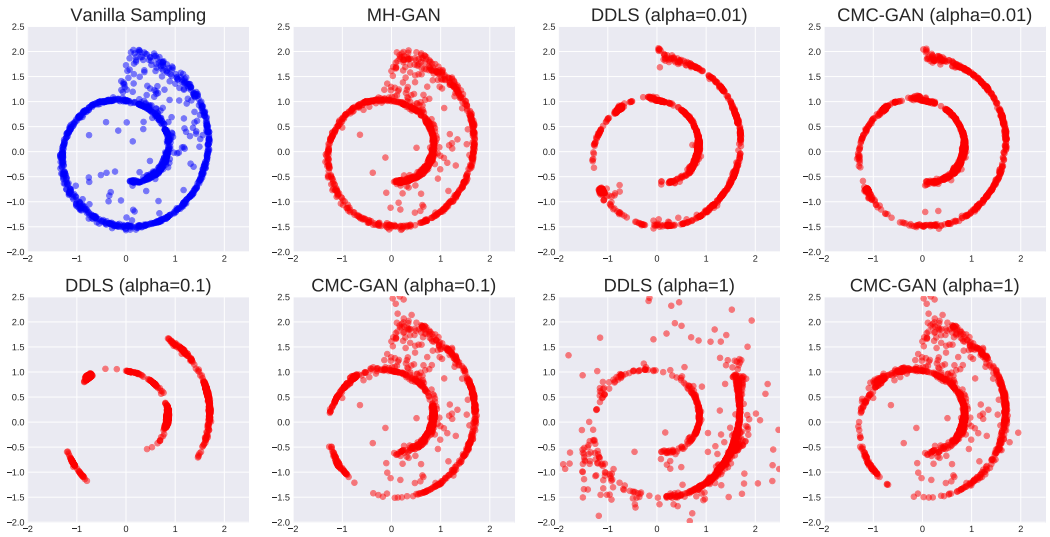

Figure 2: Visualization of samples obtained by different sampling methods for the Swiss Roll dataset. Here alpha denotes the step size α in Eqn. 18. We can see that with appropriate step size, the gradient-based methods, DDLS and CMC-GAN, outperform MH-GAN by a large margin. When the step size becomes too large, such as 0.1 and 1, the numerical error of the Langevin dynamics becomes so large that the chain either collapses or diverges. In contrast, the MH correction steps of CMC-GAN effectively prevent such behaviors by rejecting the bad samples.

Table 2: Inception scores and acceptance ratios on CIFAR10 for DCGAN and WGAN. Computed with 50,000 samples per checkpoint, and averaged over five adjacent checkpoints.

Method	DCGAN		WGAN	
	Acceptance	Inception Score	Acceptance	Inception Score
Baseline	-	3.769	-	3.557
MH-GAN (Turner et al., 2019)	0.033	3.735	0.084	3.509
DDLS (Che et al., 2020)	-	3.312	-	3.500
CMC-GAN (ours)	**0.363**	**3.820**	**0.447**	**3.598**

5.2 CIFAR10

Following the common practice (Azadi et al., 2019; Turner et al., 2019), we carry out experiments on a real image dataset, CIFAR10, for two well-known GANs, DCGAN (Radford et al., 2015) and WGAN (Arjovsky et al., 2017). We utilize the official implementation of MH-GAN[1]. and train models with the same hyperparameters. We compare the three MCMC methods, MH-GAN, DDLS, and our CMC-GAN. For a fair comparison, each Markov chain starts from a GAN sample and runs for 640 steps (Turner et al., 2019). The step size of the Langevin proposal is kept as 0.01 (Che et al., 2020). We calibrate the discriminator scores with Logistic Regression following Turner et al. (2019).

As Figure 3 shows, the training of GAN is very unstable, and the performance varies a lot from epoch to epoch. For a fair evaluation of the sampling methods, we average the reported metrics over five adjacent checkpoints with the best vanilla Inception Scores on average. We believe in such cases, the discriminator is more likely to be nearly optimal, which is the common assumption of the sampling methods. For each checkpoint, we draw 50,000 samples to compute the Inception scores and acceptance ratios. The average Inception score and acceptance ratio are shown in Table 2. We can see our method is the only one that consistently improves over the baseline.

[1] `https://github.com/uber-research/metropolis-hastings-GAN`

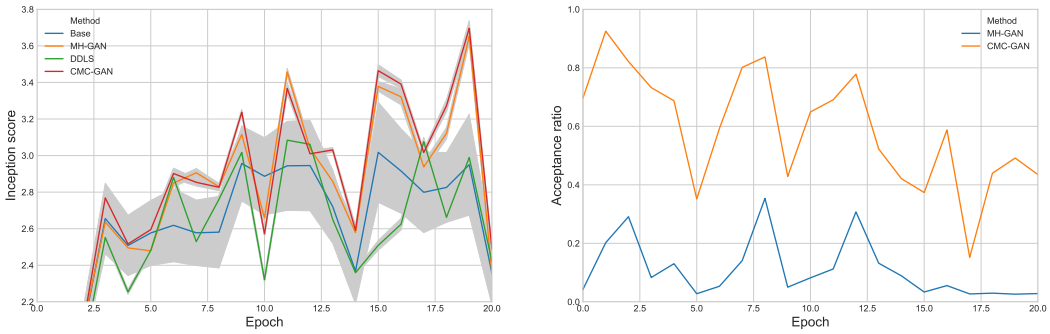

Figure 3: Average IS (Inception Score) (left) and acceptance ratio (right) v.s. training epochs with DCGAN on CIFAR-10. The standard deviation of IS is shown with the shaded error bar. (left) CMC-GAN outperforms DDLS by a large margin, showing the MH correction steps also matter on a real dataset. Also, CMC-GAN has higher IS than MH-GAN in most cases, showing the advantage of our structured proposal. (right) The average acceptance ratio of MH-GAN is lower than 0.2 in most cases, leading to poor sample efficiency. CMC-GAN, instead, has an acceptance ratio of around 0.6, which is known to be a good tradeoff between exploration and exploitation for MCMC algorithms.

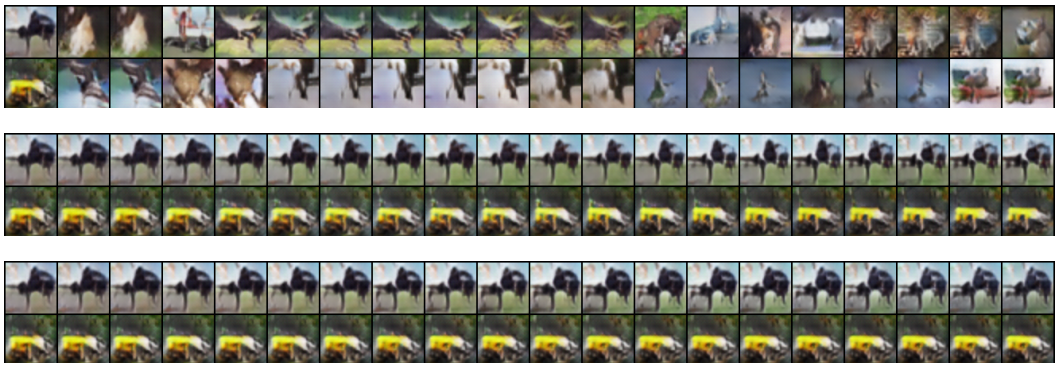

Figure 4: Visualization of the $0 - 20$ steps of the Markov chains of MH-GAN (top), DDLS (middle), and CMC-GAN (bottom). MH-GAN is often trapped in the same place because of the low acceptance probability of the independent proposals. DDLS and CMC-GAN instead take gradient-based proposals and gradually refine the samples at each step. In addition, CMC-GAN can reject the proposal with the MH test, which helps avoid some bad artifacts in the images (middle right).

We also plot the Inception Scores and the average acceptance ratio over epochs in Figure 3. It can be seen that in most cases, our MCMC is superior with much better sample efficiency. In Figure 4, we demonstrate two randomly sampled Markov chains with the same initial samples for each sampling method. Our CMC-GAN can explore the energy landscape more effectively with gradient-based proposals, while reject bad artifacts with the MH correction steps.

6 Conclusion

In this paper, we have proposed a novel method, CMC-GAN, to improve the sampling of GAN. It proposes a structured proposal that reparameterizes the sample-level transition of GAN into latent-level transition. More importantly, we first prove that this general proposal admits a tractable MH acceptance criterion. Based on this, we propose a tractable Latent Langevin Monte Carlo proposal to explore the latent landscape effectively with gradient. Experiments show our method does not only improve sample efficiency over MH-GAN but also demonstrate state-of-the-art Inception scores compared to previous methods.

REFERENCES

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

Samaneh Azadi, Catherine Olsson, Trevor Darrell, Ian Goodfellow, and Augustus Odena. Discriminator rejection sampling. *ICLR*, 2019.

Adi Ben-Israel. The change-of-variables formula using matrix volume. *SIAM Journal on Matrix Analysis and Applications*, 21(1):300–312, 1999.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Tong Che, Ruixiang Zhang, Jascha Sohl-Dickstein, Hugo Larochelle, Liam Paull, Yuan Cao, and Yoshua Bengio. Your gan is secretly an energy-based model and you should use discriminator driven latent sampling. *arXiv preprint arXiv:2003.06060*, 2020.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

Mevlana C Gemici, Danilo Rezende, and Shakir Mohamed. Normalizing flows on riemannian manifolds. *arXiv preprint arXiv:1611.02304*, 2016.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pp. 5767–5777, 2017.

Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

Matthew Hoffman, Pavel Sountsov, Joshua V Dillon, Ian Langmore, Dustin Tran, and Srinivas Vasudevan. Neutra-lizing bad geometry in hamiltonian monte carlo using neural transport. *arXiv preprint arXiv:1903.03704*, 2019.

Matthew D Hoffman. Learning deep latent gaussian models with markov chain monte carlo. In *International conference on machine learning*, pp. 1510–1519, 2017.

Zhiting Hu, Zichao Yang, Ruslan Salakhutdinov, and Eric P Xing. On unifying deep generative models. *arXiv preprint arXiv:1706.00550*, 2017.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in neural information processing systems*, pp. 10215–10224, 2018.

Richard A Levine and George Casella. Implementations of the monte carlo em algorithm. *Journal of Computational and Graphical Statistics*, 10(3):422–439, 2001.

Daniel Levy, Matthew D Hoffman, and Jascha Sohl-Dickstein. Generalizing hamiltonian monte carlo with neural networks. *arXiv preprint arXiv:1711.09268*, 2017.

Zinan Lin, Ashish Khetan, Giulia Fanti, and Sewoong Oh. Pacgan: The power of two samples in generative adversarial networks. In *Advances in neural information processing systems*, pp. 1498–1507, 2018.

Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are gans created equal? a large-scale study. In *Advances in neural information processing systems*, pp. 700–709, 2018.

Youssef Marzouk, Tarek Moselhy, Matthew Parno, and Alessio Spantini. An introduction to sampling via measure transport. *arXiv preprint arXiv:1602.05023*, 2016.

Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*, 2016.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

Radford M Neal et al. Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 54:113–162, 2010.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Tim Salimans, Diederik Kingma, and Max Welling. Markov chain monte carlo and variational inference: Bridging the gap. In *International Conference on Machine Learning*, pp. 1218–1226, 2015.

Jiaming Song, Shengjia Zhao, and Stefano Ermon. A-nice-mc: Adversarial training for mcmc. In *Advances in Neural Information Processing Systems*, pp. 5140–5150, 2017.

Yuxuan Song, Qiwei Ye, Minkai Xu, and Tie-Yan Liu. Discriminator contrastive divergence: Semi-amortized generative modeling by exploring energy of the discriminator. *arXiv preprint arXiv:2004.01704*, 2020.

Akinori Tanaka. Discriminator optimal transport. In *Advances in Neural Information Processing Systems*, pp. 6816–6826, 2019.

Michalis K Titsias. Learning model reparametrizations: Implicit variational inference by fitting mcmc distributions. *arXiv preprint arXiv:1708.01529*, 2017.

Ryan Turner, Jane Hung, Yunus Saatci, and Jason Yosinski. Metropolis-hastings generative adversarial networks. *ICML*, 2019.