The Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI-23)

CoopInit: Initializing Generative Adversarial Networks via Cooperative Learning

Yang Zhao, Jianwen Xie, Ping Li

Cognitive Computing Lab
Baidu Research
10900 NE 8th St. Bellevue, WA 98004, USA
{yangzhao.eric, jianwen.kenny, pingli98}@gmail.com

Abstract

Numerous research efforts have been made to stabilize the training of the Generative Adversarial Networks (GANs), such as through regularization and architecture design. However, we identify the instability can also arise from the fragile balance at the early stage of adversarial learning. This paper proposes the CoopInit, a simple yet effective cooperative learning-based initialization strategy that can quickly learn a good starting point for GANs, with a very small computation overhead during training. The proposed algorithm consists of two learning stages: (i) Cooperative initialization stage: The discriminator of GAN is treated as an energy-based model (EBM) and is optimized via maximum likelihood estimation (MLE), with the help of the GAN's generator to provide synthetic data to approximate the learning gradients. The EBM also guides the MLE learning of the generator via MCMC teaching; (ii) Adversarial finalization stage: After a few iterations of initialization, the algorithm seamlessly transits to the regular mini-max adversarial training until convergence. The motivation is that the MLE-based initialization stage drives the model towards mode coverage, which is helpful in alleviating the issue of mode dropping during the adversarial learning stage. We demonstrate the effectiveness of the proposed approach on image generation and one-sided unpaired image-to-image translation tasks through extensive experiments.

1 Introduction

Generative modeling has proven to be an effective approach in many scenarios, e.g., image synthesis (Xie et al. 2016, 2018; Brock, Donahue, and Simonyan 2019; Karras, Laine, and Aila 2019; Zhao, Xie, and Li 2021) and sequence generation (Tulyakov et al. 2018; Yu et al. 2017). One of the most popular and powerful generative frameworks to date is the Generative Adversarial Network (GAN) (Goodfellow et al. 2014), which defines a mini-max game seeking a Nash equilibrium between a discriminator and a generator. Despite the recent successes of GANs in modeling complex high-dimensional distributions and generating realistic images (Brock, Donahue, and Simonyan 2019; Karras et al. 2020b), their training suffers from instability issues due to alternating parameter update (Heusel et al. 2017), the sensitivity to the hyper-parameter choices (Salimans et al. 2016) and mode collapse issues (Arora, Risteski, and Zhang 2018). To alleviate these issues, several techniques have been proposed, including gradient penalty (Arjovsky, Chintala, and Bottou 2017; Mescheder, Geiger, and Nowozin 2018), spectral normalization (Miyato et al. 2018), discriminator bottleneck (Zhao et al. 2020c) and data augmentation (Karras et al. 2020a). In contrast, Generative Cooperative Networks (CoopNets) (Xie et al. 2018) are another class of generative framework that jointly trains a descriptor and a generator, which has been successfully applied to image synthesis (Xie, Zheng, and Li 2021; Xie et al. 2022b), 3D generation (Xie et al. 2020b), supervised conditional learning (Xie et al. 2022a), salient object prediction (Zhang et al. 2022), unpaired image-to-image translation (Xie et al. 2021), and image hashing (Doan et al. 2022). Unlike GANs, CoopNets are optimized through cooperative maximum likelihood estimation (MLE). The descriptor, essentially a generative energy-based model (EBM) (Xie et al. 2016; Nijkamp et al. 2019; Du and Mordatch 2019), incorporates the Stochastic Gradient Markov Chain Monte Carlo (SG-MCMC) to approximate the data distribution. The generator is an amortized sampler that simultaneously chases the descriptor towards the data distribution. The MLE-based learning scheme is often more stable and does not suffer from mode collapse issues. However, the training of CoopNets relies on an expensive MCMC sampler. It has also been suggested (Xie et al. 2020a) that likelihood-based generative models tend to generate blurry images because they are obliged to fit all the major modes of the empirical data distribution. If they cannot fit the modes closely, they interpolate the major modes.

In this work, we aim to combine adversarial learning and cooperative learning to create stable, efficient, and powerful generative models. We propose a novel approach that leverages the strengths of both learning schemes. We first demonstrate that CoopNets and GAN can share network structures, so that we can treat them as one framework conveniently. Specifically, the discriminator in GAN can be transformed into the descriptor in CoopNets, and vice versa. In other words, a bottom-up ConvNet, which plays the role of energy function of a descriptor in cooperative learning, can take a new role of discriminator in adversarial learning.

Moreover, we propose a hybrid and effective strategy to train generative models. Specifically, the proposed framework consists of two networks, a bottom-up network D_θ parameterized by θ and a top-down network G_ϕ parameterized

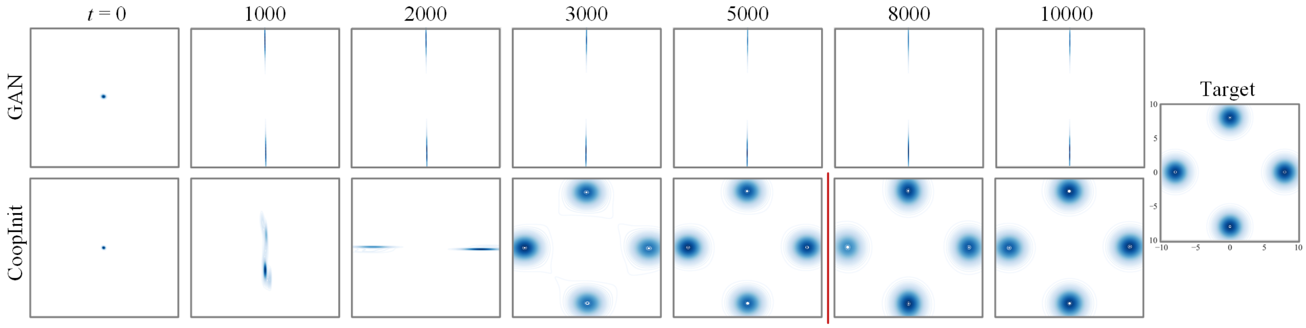

Figure 1: Comparison between standard GAN training (top row) and the GAN training with proposed CoopInit strategy (bottom row) trained on a 2D synthetic data distribution shown in the rightmost plot. Each column displays the generated distributions by the two methods at a different time step during training. The red vertical line displayed in the bottom row separates the cooperative initialization stage (left part) and the adversarial finalization stage (right part). The standard GAN training fails to converge to the target distribution because it encounters a severe mode collapse issue. In contrast, the GAN training using the CoopInit can benefit from the initial cooperative learning which helps overcome the mode collapse issue.

by ϕ. Our hybrid learning algorithm includes two stages, each of which corresponds to a different learning scheme: At the first stage (*cooperative initialization*): we train D_θ and G_ϕ in the cooperative learning scheme, where D_θ serves as an expressive EBM, to encourage mode coverage; at the second stage (*adversarial finalization*): we continue to train D_θ and G_ϕ in the adversarial learning scheme, with parameters $\{\theta, \phi\}$ initialized from the first stage. The cooperative initialization stage only takes a small amount of time at the very beginning of the whole learning process. Intuitively, we first allow the stable cooperative learning to capture the majority of the mode structure of the data distribution to avoid mode collapse or dropping, and then the subsequent adversarial learning focus on refining the synthesis details through mode chasing. We call the proposed method the CoopInit, which can be considered a learning-based initialization approach for GAN training. We demonstrate the effectiveness of CoopInit through a synthetic experiment in Figure 1. We highlight our main contributions below:

- We are the first to study how to combine the adversarial learning (i.e., GAN) and the cooperative learning (i.e., CoopNets) for generative modeling. It stabilizes and improves the adversarial training by firstly performing likelihood-based cooperative learning for initialization.
- We conduct extensive experiments for model analysis and ablation study in order to understand the behavior of the proposed learning algorithm.
- We demonstrate that the proposed training strategy can outperform previous CoopNets and GANs, and obtain state-of-the-art performance in image generation benchmarks and one-sided image translation benchmarks.

The rest of the paper is organized as follows: In Section 2, we present preliminaries of adversarial learning and cooperative learning. Section 3 describes the proposed learning framework and its theoretical understanding in detail. In Section 4, we present prior arts that are related to our model. In Section 5, we validate the proposed method via extensive experiments. Finally, in Section 6, we conclude our work.

2 Preliminaries

The generator, denoted by G_ϕ, seeks to transform a prior distribution of latent space $z \sim p(z)$, via a top-down network, into a distribution that can approximate the ground truth data distribution $p_{\text{data}}(x)$. The generator G_ϕ can pair up with either a discriminator for adversarial training or a descriptor for cooperative training, both of which can be parameterized by a bottom-up network D_θ. θ and ϕ are parameters.

2.1 Adversarial Learning

GANs (Goodfellow et al. 2014) define a minimax game between the discriminator D_θ and the generator G_ϕ. The generator G_ϕ tries to generate realistic examples to fool the discriminator D_θ whereas the discriminator D_θ aims to distinguish between the generated examples $G_\phi(z)$ where $z \sim p(z)$ and the real data examples $x \sim p_{\text{data}}(x)$. Goodfellow et al. (2014) proposed an adversarial loss, given by

$$\mathcal{L}^{\text{adv}} = \mathbb{E}_{p_{\text{data}}(x)}[\log D_\theta(x)] - \mathbb{E}_{p(z)}[\log(1 - D_\theta(G_\phi(z)))].$$

The generator tries to minimize $\mathcal{L}^{\text{adv}}$ while the discriminator tries to maximize $\mathcal{L}^{\text{adv}}$. In practice, to circumvent the vanishing gradient issues caused by a saturated discriminator, the generator is instead trained to maximize $\mathbb{E}_{p(z)}[\log D_\theta(G_\phi(z))]$. This non-saturating (NS) loss is used in a series of StyleGAN models (Karras, Laine, and Aila 2019; Karras et al. 2020b,a) and related works (Choi et al. 2020; Pidhorskyi, Adjeroh, and Doretto 2020). Wasserstein distance (Arjovsky, Chintala, and Bottou 2017) (WAS) is also a standard divergence used to train GANs. However, the introduced Lipschitz constraint in WGAN usually relies on weight clipping and is sensitive to parameters. A follow-up WGAN-GP (Gulrajani et al. 2017) instead proposes to add a gradient penalty (GP) to the WAS for enforcing the Lipschitz continuity. A notable example of WAS-GP is the ProgressiveGAN (Karras et al. 2018). The hinge loss (Lim and Ye 2017; Tran, Ranganath, and Blei 2017) (Hinge) is another common objective used to train GANs, for example in BigGAN (Brock, Donahue, and Simonyan 2019) and SN-

GAN (Miyato et al. 2018). We will evaluate the three aforementioned variants of adversarial loss in the experiments.

2.2 Cooperative Learning

In contrast to GANs, CoopNets apply a cooperative learning strategy to train the generator G and the descriptor D simultaneously via MCMC teaching. The descriptor D is essentially an EBM (Xie et al. 2016), which is defined as:

$$p_\theta(x) = \frac{1}{Z(\theta)} \exp[D_\theta(x)], \tag{1}$$

where $D_\theta(x)$ is the negative energy function defined on data domain and Z_θ is the intractable normalizing constant. To learn the descriptor, we seek to maximize the log-likelihood:

$$\mathcal{L} = \mathbb{E}_{p_{\text{data}}(x)}[\log p_\theta(x)], \tag{2}$$

which is equivalent to minimizing the Kullback-Leibler divergence $\mathsf{KL}(p_{\text{data}}(x)||p_\theta)$. Its derivative is given by

$$\begin{aligned} \nabla_\theta \mathcal{L} &= \mathbb{E}_{p_{\text{data}}(x)}[\nabla_\theta D_\theta(x)] - \mathbb{E}_{p_\theta(x)}[\nabla_\theta D_\theta(x)] \\ &\approx \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta D_\theta(x_i) - \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta D_\theta(\tilde{x}_i), \end{aligned} \tag{3}$$

where $\{x_i\} \sim p_{\text{data}}(x)$ are observed examples and $\{\tilde{x}_i\} \sim p_\theta(x)$ are synthesized examples generated via MCMC, such as Langevin dynamics (Zhu and Mumford 1998) that iterates the following step

$$x_{t+1} = x_t + \eta \nabla_x D_\theta(x_t) + \epsilon_t, \epsilon_t \sim \mathcal{N}(0, \sqrt{2\eta} I), \tag{4}$$

with t indexing the Langevin time step and $x_{t=0}$ being initialized by random noise. η is a hyperparameter for Langevin step size. In high dimensional modeling cases, the MCMC can be expensive and difficult to converge. However, CoopNets can improve the sampling by using a generator G_ϕ to generate initial synthesized examples to initialize a finite-step MCMC that samples and trains the descriptor D_θ. The generator serves as an amortized sampler for the descriptor. The generator updates its parameters by directly learning from the synthesized examples produced by the MCMC, which is called MCMC teaching. The descriptor learns from the difference between MCMC outputs and training examples, while the generator learns from how the descriptor revises the initial outputs. Algorithm 1 presents one iteration of the cooperative learning.

Algorithm 1: Cooperative Learning

Require: descriptor D_θ, generator G_ϕ, Langevin dynamics step size η, number of Langevin steps T.

```
# Step G1 : Generate initial examples x̂
```
$\hat{z}_i \sim p(z), \hat{x}_i = G_\phi(\hat{z}_i)$
```
# Step D1 : Revise x̃ for T steps via LD
```
Initialize $\tilde{x}_i = \hat{x}_i$

for 1 to T **do**

  $\tilde{x}_i \leftarrow \tilde{x}_i + \eta \nabla_x D_\theta(\tilde{x}_i) + \epsilon, \epsilon \sim \mathcal{N}(0, \sqrt{2\eta} I)$

end for
```
# Step D2 : Train descriptor D_θ
```
Train D_θ with gradient descent via Eq. (3)
```
# Step G2 : Train generator G_φ
```
Train G_ϕ with Adam on $\frac{1}{n}\sum_{i=1}^{n} ||\tilde{x}_i - G_\phi(\hat{z}_i)||^2$

Algorithm 2: Training a GAN with CoopInit

Require: descriptor D_θ , generator G_ϕ, number of examples consumed by cooperative initialization N_{coop}, and number of examples consumed by adversarial finalization N_{adv}, batch size n.

$N \leftarrow 0$
```
# Train D and G as CoopNets
```
while $N \leq N_{\text{coop}}$ **do**

  Run Algorithm 1 to update D_θ and G_ϕ

  $N \leftarrow N + n$

end while
```
# Train D and G as GAN
```
while $N \leq (N_{\text{adv}} + N_{\text{coop}})$ **do**

  Update D_θ and G_ϕ according to $\mathcal{L}^{\text{adv}}$

  $N \leftarrow N + n$

end while

---

3 CoopInit: A Strategy to Initialize GAN Training via Cooperative Learning

3.1 Proposed Framework

Our generative learning framework, shown in Figure 2, integrates CoopNets and GAN, enabling us to smoothly switch between cooperative learning and adversarial learning. The proposed method begins with limited iterations of cooperative learning and then switches to adversarial learning until completion. We monitor the training progress using the number of training examples processed by the model. Specifically, We use N_{coop} and N_{adv} to represent the numbers of training examples consumed during cooperative learning and adversarial learning, respectively. The full description of training a GAN with CoopInit is shown in Algorithm 2. In this paper, we always ensure that $N_{\text{coop}}/N_{\text{adv}} < 3$ to keep the computational overhead from MCMC negligible.

One might question why we don't simply use a combined objective of cooperative and adversarial learning. However, in practice, we have found that their compatibility is poor, resulting in an FID (Heusel et al. 2017) of approximately 35 for image generation on CIFAR-10 (Krizhevsky 2009) dataset using both cooperative and adversarial learning simultaneously. The cooperative learning leads to an MLE solution, which corresponds to a forward Kullback–Leibler (KL)-divergence, while the adversarial learning corresponds to Jensen–Shannon divergence, which involves a reverse KL-divergence. Thus, learning the models using these two objectives at the same time might lead to undesirable outcome due to incompatibility. Although both CoopNets and GAN use an alternating optimization procedure between D_θ and G_ϕ, the key difference between cooperative and adversarial learning lies in that CoopNets uses MLE but GAN uses an adversarial loss. Further analysis of their optimization procedures reveals the following:

(i) The role of D_θ differs in the optimization of GAN and CoopNets. In GAN, D_θ functions as a classifier that distinguishes between real data and generated data. In CoopNets, D_θ is a score (negative energy) function that assigns lower scores to generated data and higher scores to real data.

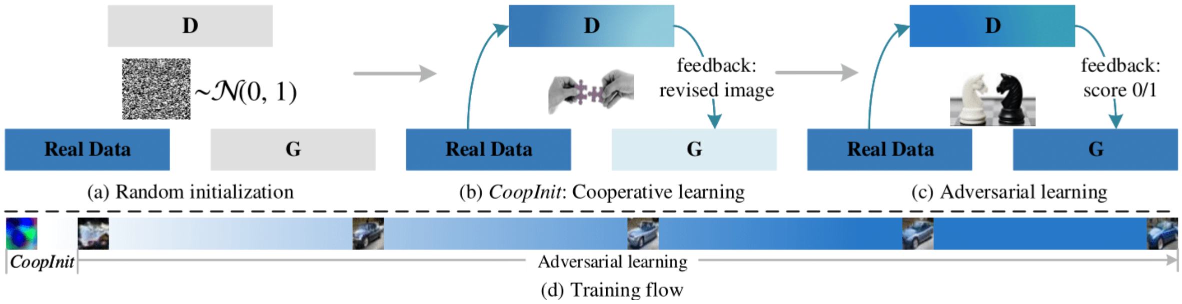

Figure 2: An illustration of the CoopInit technique for improving GAN training. D: discriminator or descriptor. G: generator.

(ii) The objective of G_ϕ differs in the optimization of GAN and CoopNets. In GAN, G_ϕ is optimized by fooling D_θ into believing that generated examples are real. On the other hand, in CoopNets, G_ϕ is optimized by moving the generator's distribution towards the descriptor's distribution.

3.2 Theoretical Understanding

We use $\mathcal{M}_\theta$ to denote the T-step MCMC transition kernel of the descriptor p_θ. We also use $\mathcal{M}_\theta q_\phi$ to denote the marginal distribution obtained by running the Markov transition $\mathcal{M}_\theta$ starting from the generator q_ϕ. At each iteration t, the cooperative learning algorithm alternates the following two steps: (i) Update θ: it learns θ by minimizing

$$\mathsf{KL}(p_{\text{data}} \| p_\theta) - \mathsf{KL}(\mathcal{M}_{\theta^{(t)}} q_{\phi^{(t)}} \| p_\theta) \tag{5}$$

over θ, which is a modified contrastive divergence (Xie et al. 2018) for the energy-based model p_θ, and (ii) Update ϕ: it learns ϕ by minimizing

$$\mathsf{KL}(\mathcal{M}_{\theta^{(t)}} q_{\phi^{(t)}} \| q_\phi) \tag{6}$$

over ϕ. In an idealized situation where the generator q_ϕ has infinite capacity, the objective in Eq. (6) can be minimized to zero, which means that q_ϕ becomes the stationary distribution of $\mathcal{M}_\theta$, i.e., $\mathcal{M}_\theta q_\phi = q_\phi$, or equivalently $p_\theta = q_\phi$ (the generator has caught up with the descriptor and become an amortized sampler for the descriptor). Once this happens, the second KL-divergence in Eq. (5) vanishes, because $\mathsf{KL}(\mathcal{M}_\theta q_\phi \| p_\theta) = \mathsf{KL}(q_\phi \| p_\theta) = 0$. Then the learning of θ becomes maximum likelihood estimate that minimizes only the first KL-divergence $\mathsf{KL}(p_{\text{data}} \| p_\theta)$ in Eq. (5). Since q_ϕ chases p_θ toward p_{data}, the learning of ϕ is also a maximum likelihood estimate.

In the second stage of the proposed algorithm, known as adversarial finalization, we continue to train G_ϕ, which is initialized by the cooperative learning, to further refine its ability to capture major modes. Since G_ϕ already aims to cover all modes during the cooperative initialization stage, it is less likely to dropping major modes it already covers at the second stage. As to the discriminator or the descriptor D_θ, in the stage of cooperative initialization, the output of the descriptor D_θ is a score representing negative energy. Real data typically receives higher scores (i.e., lower energy) from descriptor D_θ. Similarly, in the adversarial finalization stage, the discriminator D_θ assigns larger probabilities to real data. Thus, both the descriptor and the discriminator can be viewed as classifiers with a shared objective. This allows us to initialize the discriminator with the descriptor.

4 Related Work

The following themes are closely related to our work, and we will briefly review each of them and explain their connection to our work.

Regularization Techniques for GANs: This line of research is based on both theoretical investigations and empirical studies on the convergence properties of GANs, in which regularization is used to ensure a good local equilibrium with new model assumptions. Various research efforts have been made in this direction, e.g., adding loss penalty (Gulrajani et al. 2017; Mescheder, Geiger, and Nowozin 2018), weight regularization (Miyato et al. 2018; Brock, Donahue, and Simonyan 2019) and implementing a discriminator bottleneck (Zhao et al. 2020c). We can interpret the CoopInit as a special regularization technique, which only takes effect at the early stage of the learning process, to enforce the model to cover most of the modes in the data distribution.

Link MLE to GAN: The most successful works in linking MLE to GAN exist in the applications of GAN-based text generation (Yu et al. 2017; Nie, Narodytska, and Patel 2019). To mitigate the gradient estimation difficulty and mode collapse issues on discrete data, they apply large amount of MLE pretraining and limited adversarial fine-tuning. However, our CoopInit simultaneously trains both networks as a whole in an MLE-based cooperative manner using very limited time, whereas MLE pretraining used in Yu et al. (2017) trains them separately for most of the time. Besides, another work Flow-GAN (Grover, Dhar, and Ermon 2018) uses a normalizing flow (Kingma and Dhariwal 2018) as the generator to build a GAN. But, the expressive power of a normalizing flow is limited due to its restrictive network design. Zhao et al. (2020a) explore unifying the advantages of MLE and adversarial learning via α-divergence but only trains the generator by MLE. Our method seamlessly bridges the MLE and GAN by the energy-based cooperative learning.

Link EBM to GAN: Several works have investigated the relationship between EBMs and GANs (Finn et al. 2016; Che et al. 2020). Among these, DDLS (Che et al. 2020) is the most relevant, as it considers the discriminator as an energy function and employs MCMC in the latent space to generate refined samples. But, our CoopInit differs from DDLS in that we jointly train an EBM and a generator before GAN training, whereas DDLS only refines samples via MCMC after GAN training, without explicitly training an EBM.

5 Experiments

In this section, we extensively evaluate the effectiveness of our proposed initialization strategy, CoopInit, for GANs. We begin by testing our method on image generation and unpaired image-to-image translation, comparing our framework to state-of-the-art models. Then we perform some analysis on our model. All experiments were conducted on 4 Nvidia Titan Xp (12GB) GPUs and Google Colab.

5.1 Experimental Setup

Base Model In terms of performance, StyleGAN2 is currently the most attractive GAN model that can achieve state-of-the-art results on a variety of image synthesis tasks, such as image generation (Karras et al. 2020a; Zhao et al. 2020b), image translation (Richardson et al. 2020; Zhao and Chen 2020) and image manipulation (Abdal, Qin, and Wonka 2019). StyleGAN2-ADA (Karras et al. 2020a) is a specifically tuned GAN with techniques such as shallow mapping, disable style mixing regularization (Karras, Laine, and Aila 2019), path length regularization, and residual connections in the discriminator. This model currently achieves state-of-the-art results on CIFAR-10 (Krizhevsky 2009) image generation among all GANs. BigGAN (Brock, Donahue, and Simonyan 2019) that was designed for generating high-resolution and high-fidelity images is also considered.

Datasets We evaluate the performance of image generation on four widely used datasets listed below:

(i) CIFAR-10 (Krizhevsky 2009): This dataset consists of 60K 32×32 images in 10 evenly distributed classes, including 50K training images and 10K testing images.

(ii) ImageNet (Russakovsky et al. 2015): To balance the computational budget, we use a down-sampled version of ImageNet that consists of 32×32 images. ImageNet contains over 10 million natural images of 1,000 classes.

(iii) FFHQ (Karras, Laine, and Aila 2019): This dataset consists of 70K high-quality and diverse human facial images. We choose to use a down-sampled version of the data with a resolution of 256×256.

Metric Frèchet Inception Distance (FID) (Heusel et al. 2017) is a widely used metric for evaluating the quality of generated images. It computes the distance between the Inception feature vectors for real and generated images. It is also consistent with increasing disturbances and human judgment. A low FID indicates that the model can create high-quality images. We adopt the commonly used 50K-FID, which generates 50K examples to evaluate image generation quality, as in most contemporary GAN works.

5.2 Image Generation

Evaluation on CIFAR-10 Dataset We compare the proposed approach with state-of-the-art models on CIFAR-10 generation, and the results are shown in Table 1. It is worth noting that when we disable the adaptive discriminator augmentation (ADA) in the base model StyleGAN2-ADA, the CoopInit can greatly reduce the FID from 6.40 to 4.34, even without R_1 regularization (Mescheder, Geiger, and Nowozin 2018) (i.e., we set the hyperparameter of R_1 regularization $\gamma = 0$). This is currently the best FID achieved by GANs on CIFAR-10 without using ADA. We further find that increasing the network depth hurts performance. When we double the width, the performance of tuned StyleGAN2-CoopInit is on par with that of NCSN++cont. and achieves a new state-of-the-art result of GAN on CIFAR-10. We report the best FID of the generated images and evaluate the Inception Score (IS). Figure 3 shows uncurated generated examples.

Models	FID↓	IS↑
Conditional		
BigGAN (Brock, Donahue, and Simonyan 2019)	14.73	9.22
MultiHinge (Kavalerov and Czaja 2019)	6.40	9.58
FQ-GAN (Zhao et al. 2020c)	5.59	8.48
BigGAN + CoopInit (ours)	6.95	9.35
StyleGAN2 w/ ADA (Karras et al. 2020a)	2.42	10.14
+ CoopInit + tuning (ours)	**2.20**	**10.20**
Unconditional		
CoopNets (Xie et al. 2018)	33.61	-
CoopVAEBM (Xie, Zheng, and Li 2021)	36.20	-
CoopFlow (Xie et al. 2022b)	15.80	-
CF-EBM (Zhao, Xie, and Li 2021)	16.71	-
ProGAN (Karras et al. 2018)	15.52	8.56
NCSNv2 (Song and Ermon 2020)	10.87	8.40
CAS (Jolicoeur-Martineau et al. 2021)	3.65	-
DDPM (Ho, Jain, and Abbeel 2020)	3.17	9.46
StyleGAN2-ADA (Karras et al. 2020a)	2.92	9.83
NCSN++cont. (Song et al. 2021)	**2.20**	9.89
StyleGAN2 w/o ADA ($\gamma = 0.01$)	6.40	9.55
+ CoopInit (ours) ($\gamma = 0.00$)	**4.34**	9.69
StyleGAN2 w/ ADA (Karras et al. 2020a)	2.92	9.83
+ CoopInit (ours)	2.82	9.88
+ tuning (ours)	2.55	**9.94**

Table 1: FID and Inception score (IS) comparison on conditional and unconditional CIFAR-10 image generation.

Models	FID ↓
BigGAN (Brock, Donahue, and Simonyan 2019)	11.48
U-Net GAN (Schonfeld, Schiele, and Khoreva 2020)	7.48
StyleGAN2	3.84
+ CoopInit (Ours)	**3.61**

Table 2: FID comparison on FFHQ 256×256.

Evaluation on FFHQ Dataset Compared to CIFAR-10 and CIFAR-100 datasets, the image distribution of FFHQ dataset is more concentrated but less diverse. The CoopInit method can consistently outperform the baseline, as shown in Table 2. Qualitative results are presented in Figure 4.

Evaluation on ImageNet Dataset In our previous study, we show that CoopInit can significantly improve the performance of GANs in various scenarios. To further evalu-

Figure 3: Generated examples by the StyleGAN2-CoopInit-ADA models trained on the CIFAR10 dataset. (Left: Unconditional generation. Right: Conditional generation.)

Figure 4: Qualitative results of FFHQ 256×256 image generation.

ate its effectiveness, we conduct a study on a more complex dataset, ImageNet. The results in Table 3 indicate that although CoopInit performs better on unconditional generation, its performance on conditional generation is only comparable to the baseline. We suspect that this is because the label information can alleviate the mode collapse issue to some extent, which aligns with the objective of CoopInit.

Models	FID↓
ImageNet (32×32)	
PixelCNN (Van den Oord et al. 2016)	33.27
PixelIQN (Ostrovski, Dabney, and Munos 2018)	22.99
IGEBM (Du and Mordatch 2019)	14.31
StyleGAN2 w/o labels	6.87
+ CoopInit (ours)	**5.84**
StyleGAN2 w/ labels	3.87
+ CoopInit (ours)	3.84
ImageNet (64×64)	
BigGAN w/ labels (Brock, Donahue, and Simonyan 2019)	10.55
+CoopInit	10.63

Table 3: FID comparison on ImageNet dataset.

5.3 Unpaired One-sided Image Translation

The proposed CoopInit is also tested in the context of adversarial image-to-image translation. We evaluate our approach on the recently proposed approach CUT (Park et al. 2020), which enables one-sided image-to-image translation using patch-wise contrastive learning and adversarial learning for content preservation and style transfer. The results, both quantitative and qualitative, shown in Table 4 and Figure 5, outperform the baselines. We observe an improvement in the performance of CUT when CoopInit is employed. The baseline method CF-EBM (Zhao, Xie, and Li 2021) is an energy-based model that uses short-run Langevin dynamics as a flow-like generator to transform images from the source domain to the target domain. We encountered difficulties when applying CF-EBM to the Horse⇒Zebra task, and we suspect that this may be due to misalignment between the source and target datasets. Additionally, it is worth noting that in the cooperative initialization stage, our generator performs a direct transformation of the source domain images to the target domain. The output is then fed into the Langevin dynamics of the descriptor for a few steps of revision. Compared to CF-EBM, CoopInit employs a top-down generator to amortize the computationally expensive MCMC process.

Models	FID↓ C⇒D	FID↓ H⇒Z
Distance (Benaim and Wolf 2017)	155.3	72.0
SelfDistance (Benaim and Wolf 2017)	144.4	80.8
GCGAN (Fu et al. 2019)	96.6	86.7
CF-EBM (Zhao, Xie, and Li 2021)	**55.1**	-
CUT (Park et al. 2020)	76.2	45.5
+ CoopInit (ours)	61.3	**38.7**

Table 4: Comparison on one-sided unpaired image-to-image translation. (C⇒D: Cat⇒Dog. H⇒Z: Horse⇒Zebra)

5.4 Model Analysis

To investigate the impact of adversarial loss functions and hyperparameters, we test the CoopInit on CIFAR-10 dataset for image generation. Following Zhao et al. (2020b), we halve the number of channels of feature maps at higher resolution layers (i.e., 16×16 and above) to enable faster computation. We further apply the non-saturating loss, set the learning rate to 0.0025, and use the original connection unless specified otherwise, following the approach of Karras et al. (2020a). To ensure fair comparisons, we temporarily disable the lazy mode of R_1 regularization. This is because the lazy mode leads to a different optimization process, which requires a decrease in the learning rate and hyperparameters in the Adam optimizer (Karras et al. 2020b).

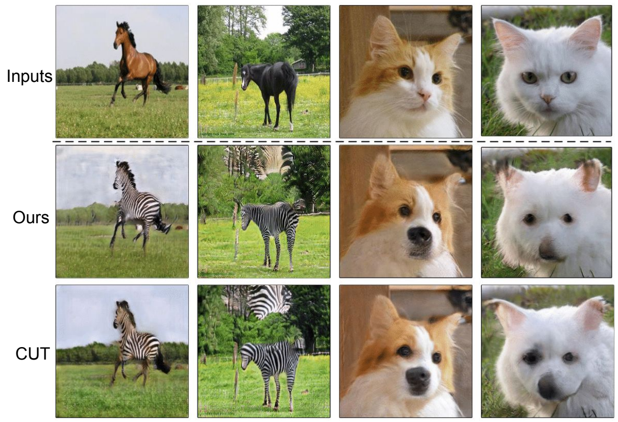

Figure 5: Comparison of qualitative results for one-sided unpaired image-to-image translation using the baseline CUT and our method CUT+CoopInit.

We use 100M real images for each run with data augmentation and 25M without.

Impact of Loss Functions We conduct an investigation into the impact of different adversarial loss variants on training GANs, including Hinge loss (Hinge), non-saturating loss (NS), and Wasserstein distance with gradient penalty (WAS-GP). After extensive hyper-parameter tuning, we select the best learning rate and report the FIDs in Table 5. To ensure a fair comparison, all tests share the same architecture, and each column uses the same optimizers. As shown in Table 5, CoopInit consistently yields lower FIDs on the three loss variants, with the most significant improvements observed on Hinge and NS losses. We also find that the default NS loss with R_1 regularization is the most appropriate loss function to train StyleGAN2, but this is no longer the case when CoopInit is applied. Interestingly, CoopInit with StyleGAN2-Hinge is found to yield a much better FID compared with StyleGAN2-NS with R_1 regularization.

Methods	NS	Hinge	WAS-GP
StyleGAN2	13.95 (8.71*)	11.64	13.21
+ CoopInit (ours)	**5.85**	**5.09**	**11.83**

Table 5: CoopInit improves StyleGAN2 with different variants of adversarial loss. The sign * indicates a performance obtained using R_1 regularization with $\gamma = 0.01$.

Impact of Hyperparameters We evaluate the impact of hyperparameters, including learning rate lr and R_1 regularization strength γ, on the proposed learning algorithm.

Learning Rate. We conduct two sets of experiments with non-saturating (NS) loss and Hinge loss, respectively, to study how our CoopInit technique behaves when the learning rate varies. The results are shown in Figure 6 (left) and Table 6. As shown in Figure 6 (Left), StyleGAN2-Hinge benefits greatly from the CoopInit technique across all different learning rates. In particular, when we increase the learning rate lr to 0.003, CoopInit can eliminate the acute oscillation of the original StyleGAN2-Hinge and drive the model to reach the fastest convergence rate among all learning rate settings. This verifies the effectiveness of CoopInit and the importance of a good initial point for GAN training. Table 6 also confirms the results under different R_1 regularization hyperparameters γ with NS loss.

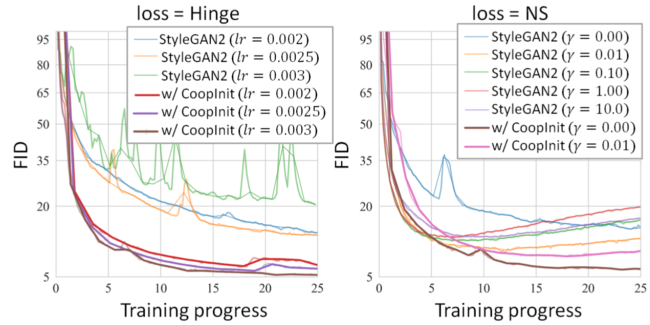

Figure 6: Learning curves with different GAN variants.

Methods	γ	$lr \times 10^{-3}$		
		2.0	2.5	3.0
StyleGAN2	0.00	13.21	13.95	14.94
+ CoopInit		6.16	6.07	**5.85**
StyleGAN2	0.01	9.28	8.95	8.71
+ CoopInit		8.87	8.29	**7.58**

Table 6: CoopInit improves StyleGAN2-NS across different learning rates (lr) and two R_1 regularization settings.

R_1 Regularization Strength. R_1 regularization is a critical technique to stabilize StyleGAN2-NS training and helps to reach a local equilibrium faster (Mescheder, Geiger, and Nowozin 2018). In the right panel of Figure 6, we plot learning curves for models using various values of γ, which is a hyperparameter in R_1 regularization and represents the strength. We observe that StyleGAN2-NS is very sensitive to the regularization strength, and the performance deteriorates after some iterations. We find that $\gamma = 0.01$ works best, which is consistent with Karras et al. (2020a). In contrast, the minimum FID and the most stable learning curve can be obtained when we replace R_1 regularization by CoopInit in training StyleGAN2-NS. Results in Table 6 demonstrates that CoopInit works best without using R_1 regularization.

6 Conclusion

To summarize, this paper establishes a new connection between cooperative learning and adversarial learning by proposing to adopt cooperative learning (i.e., CoopNets algorithm) to initialize GAN training. Our hybrid learning scheme, CoopInit, allows us to seamlessly integrate the strengths of both CoopNets and GAN, and it is compatible with various techniques for stabilizing and enhancing GANs. We demonstrate significantly improved performance across extensive experimental settings and datasets. We also achieve a new state-of-the-art result for image generation on CIFAR-10 dataset. Future works can explore broader applications of CoopInit, e.g., generative representation learning.

References

Abdal, R.; Qin, Y.; and Wonka, P. 2019. Image2stylegan: How to embed images into the stylegan latent space? In *IEEE International Conference on Computer Vision (ICCV)*, 4432–4441.

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein generative adversarial networks. In *International Conference on Machine Learning (ICML)*, 214–223.

Arora, S.; Risteski, A.; and Zhang, Y. 2018. Do GANs learn the distribution? Some Theory and Empirics. In *International Conference on Learning Representations (ICLR)*.

Benaim, S.; and Wolf, L. 2017. One-sided unsupervised domain mapping. *arXiv preprint arXiv:1706.00826*.

Brock, A.; Donahue, J.; and Simonyan, K. 2019. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations (ICLR)*.

Che, T.; Zhang, R.; Sohl-Dickstein, J.; Larochelle, H.; Paull, L.; Cao, Y.; and Bengio, Y. 2020. Your GAN is Secretly an Energy-based Model and You Should Use Discriminator Driven Latent Sampling. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Choi, Y.; Uh, Y.; Yoo, J.; and Ha, J.-W. 2020. Stargan v2: Diverse image synthesis for multiple domains. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 8188–8197.

Doan, K. D.; Xie, J.; Zhu, Y.; Zhao, Y.; and Li, P. 2022. CoopHash: Cooperative Learning of Multipurpose Descriptor and Contrastive Pair Generator via Variational MCMC Teaching for Supervised Image Hashing. *arXiv preprint arXiv:2210.04288*.

Du, Y.; and Mordatch, I. 2019. Implicit Generation and Modeling with Energy Based Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 3603–3613.

Finn, C.; Christiano, P.; Abbeel, P.; and Levine, S. 2016. A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models. *arXiv preprint arXiv:1611.03852*.

Fu, H.; Gong, M.; Wang, C.; Batmanghelich, K.; Zhang, K.; and Tao, D. 2019. Geometry-consistent generative adversarial networks for one-sided unsupervised domain mapping. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2427–2436.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, 2672–2680.

Grover, A.; Dhar, M.; and Ermon, S. 2018. Flow-gan: Combining maximum likelihood and adversarial learning in generative models. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 32.

Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. C. 2017. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems (NIPS)*, 5767–5777.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems (NIPS)*, 6626–6637.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems (NeurIPS)*.

Jolicoeur-Martineau, A.; Piché-Taillefer, R.; Mitliagkas, I.; and des Combes, R. T. 2021. Adversarial score matching and improved sampling for image generation. In *International Conference on Learning Representations (ICLR)*.

Karras, T.; Aila, T.; Laine, S.; and Lehtinen, J. 2018. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In *International Conference on Learning Representations (ICLR)*.

Karras, T.; Aittala, M.; Hellsten, J.; Laine, S.; Lehtinen, J.; and Aila, T. 2020a. Training generative adversarial networks with limited data. *Advances in Neural Information Processing Systems (NeurIPS)*.

Karras, T.; Laine, S.; and Aila, T. 2019. A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4401–4410.

Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; and Aila, T. 2020b. Analyzing and improving the image quality of stylegan. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 8110–8119.

Kavalerov, I.; and Czaja, W. 2019. cGANs with Multi-Hinge Loss. *arXiv preprint arXiv:1912.04216*.

Kingma, D. P.; and Dhariwal, P. 2018. Glow: Generative Flow with Invertible 1x1 Convolutions. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 10236–10245.

Krizhevsky, A. 2009. Learning multiple layers of features from tiny images. Technical report, University of Toronto.

Lim, J. H.; and Ye, J. C. 2017. Geometric GAN. *arXiv preprint arXiv:1705.02894*.

Mescheder, L.; Geiger, A.; and Nowozin, S. 2018. Which training methods for GANs do actually converge? In *International Conference on Machine Learning (ICML)*, 3481–3490.

Miyato, T.; Kataoka, T.; Koyama, M.; and Yoshida, Y. 2018. Spectral Normalization for Generative Adversarial Networks. In *International Conference on Learning Representations (ICLR)*.

Nie, W.; Narodytska, N.; and Patel, A. 2019. RelGAN: Relational Generative Adversarial Networks for Text Generation. In *International Conference on Learning Representations (ICLR)*.

Nijkamp, E.; Hill, M.; Zhu, S.; and Wu, Y. N. 2019. Learning Non-Convergent Non-Persistent Short-Run MCMC Toward Energy-Based Model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 5233–5243.

Ostrovski, G.; Dabney, W.; and Munos, R. 2018. Autoregressive quantile networks for generative modeling. In *International Conference on Machine Learning (ICML)*, 3936–3945.

Park, T.; Efros, A. A.; Zhang, R.; and Zhu, J.-Y. 2020. Contrastive learning for unpaired image-to-image translation. In *European Conference on Computer Vision (ECCV)*, 319–345. Springer.

Pidhorskyi, S.; Adjeroh, D. A.; and Doretto, G. 2020. Adversarial Latent Autoencoders. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 14104–14113.

Richardson, E.; Alaluf, Y.; Patashnik, O.; Nitzan, Y.; Azar, Y.; Shapiro, S.; and Cohen-Or, D. 2020. Encoding in style: a stylegan encoder for image-to-image translation. *arXiv preprint arXiv:2008.00951*.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M. S.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3): 211–252.

Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; and Chen, X. 2016. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems (NIPS)*, 2226–2234.

Schonfeld, E.; Schiele, B.; and Khoreva, A. 2020. A u-net based discriminator for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 8207–8216.

Song, Y.; and Ermon, S. 2020. Improved techniques for training score-based generative models. *Advances in Neural Information Processing Systems (NeurIPS)*.

Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B. 2021. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations (ICLR)*.

Tran, D.; Ranganath, R.; and Blei, D. M. 2017. Hierarchical implicit models and likelihood-free variational inference. *Advances in Neural Information Processing Systems (NIPS)*, 5523–5533.

Tulyakov, S.; Liu, M.-Y.; Yang, X.; and Kautz, J. 2018. Mocogan: Decomposing motion and content for video generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1526–1535.

Van den Oord, A.; Kalchbrenner, N.; Espeholt, L.; Vinyals, O.; Graves, A.; et al. 2016. Conditional image generation with pixelcnn decoders. *Advances in Neural Information Processing Systems (NIPS)*, 29: 4790–4798.

Xie, J.; Gao, R.; Nijkamp, E.; Zhu, S.-C.; and Wu, Y. N. 2020a. Representation learning: A statistical perspective. *Annual Review of Statistics and Its Application*, 7: 303–335.

Xie, J.; Lu, Y.; Gao, R.; Zhu, S.-C.; and Wu, Y. N. 2018. Cooperative training of descriptor and generator networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 42(1): 27–45.

Xie, J.; Lu, Y.; Zhu, S.-C.; and Wu, Y. 2016. A theory of generative convnet. In *International Conference on Machine Learning (ICML)*, 2635–2644.

Xie, J.; Zheng, Z.; Fang, X.; Zhu, S.-C.; and Wu, Y. N. 2021. Learning Cycle-Consistent Cooperative Networks via Alternating MCMC Teaching for Unsupervised Cross-Domain Translation. In *AAAI Conference on Artificial Intelligence (AAAI)*, 10430–10440.

Xie, J.; Zheng, Z.; Fang, X.; Zhu, S.-C.; and Wu, Y. N. 2022a. Cooperative Training of Fast Thinking Initializer and Slow Thinking Solver for Multi-Modal Conditional Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 44(8): 3957–3973.

Xie, J.; Zheng, Z.; Gao, R.; Wang, W.; Zhu, S.-C.; and Wu, Y. N. 2020b. Generative VoxelNet: Learning Energy-Based Models for 3D Shape Synthesis and Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Xie, J.; Zheng, Z.; and Li, P. 2021. Learning Energy-Based Model with Variational Auto-Encoder as Amortized Sampler. In *AAAI Conference on Artificial Intelligence (AAAI)*, 10441–10451.

Xie, J.; Zhu, Y.; Li, J.; and Li, P. 2022b. A Tale of Two Flows: Cooperative Learning of Langevin Flow and Normalizing Flow Toward Energy-Based Model. In *The Tenth International Conference on Learning Representations (ICLR)*.

Yu, L.; Zhang, W.; Wang, J.; and Yu, Y. 2017. SeqGAN: Sequence generative adversarial nets with policy gradient. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2852–2858.

Zhang, J.; Xie, J.; Zheng, Z.; and Barnes, N. 2022. Energy-Based Generative Cooperative Saliency Prediction. In *AAAI Conference on Artificial Intelligence (AAAI)*, 3280–3290.

Zhao, M.; Cong, Y.; Dai, S.; and Carin, L. 2020a. Bridging Maximum Likelihood and Adversarial Learning via α-Divergence. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, 6901–6908.

Zhao, S.; Liu, Z.; Lin, J.; Zhu, J.-Y.; and Han, S. 2020b. Differentiable augmentation for data-efficient GAN training. *Advances in Neural Information Processing Systems (NeurIPS)*, 33.

Zhao, Y.; and Chen, C. 2020. Unpaired Image-to-Image Translation via Latent Energy Transport. *arXiv preprint arXiv:2012.00649*.

Zhao, Y.; Li, C.; Yu, P.; Gao, J.; and Chen, C. 2020c. Feature Quantization Improves GAN Training. In *International Conference on Machine Learning (ICML)*, 11376–11386.

Zhao, Y.; Xie, J.; and Li, P. 2021. Learning Energy-Based Generative Models via Coarse-to-Fine Expanding and Sampling. In *International Conference on Learning Representations (ICLR)*.

Zhu, S. C.; and Mumford, D. 1998. Grade: Gibbs reaction and diffusion equations. In *International Conference on Computer Vision (ICCV)*, 847–854.