YOUR DENOISING IMPLICIT MODEL IS A SUB-OPTIMAL ENSEMBLE OF DENOISING PREDICTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Denoising diffusion models construct a Markov denoising process to learn the transport from Gaussian noise distribution to the data distribution, however require thousands of denoising steps to achieve the SOTA generative performance. Denoising diffusion implicit models (DDIMs) introduce non-Markovian process to largely reduce the required steps, but its performance degenerates as the sampling steps further reducing. In this work, we show that DDIMs belong to our *ensemble denoising implicit models* which heavily rely on the convex ensemble of obtained denoising predictions. We propose improved DDIM (iDDIM) to demonstrate DDIMs adopt sub-optimal ensemble coefficients. The iDDIM can largely improve on DDIMs, but still deteriorates in the case of a few sampling steps. Thus we further propose *generalized denoising implicit model* (GDIM) that replace the ensemble prediction with a probabilistic inference conditioned on the obtained states. Then a specific instance t-GDIM that only depends on the latest state is parameterized by the conditional energy-based model (EBM) and variational sampler. The models are jointly trained with variational maximum likelihood. Extensive experiments show t-GDIM can reduces the sampling steps to only 4 and remains comparable generative quality to other generative models.

1 INTRODUCTION

Modern deep generative modelling focuses on learning transport from a tractable reference distribution (e.g. Gaussian) to the target distribution, and the learned transport is applied on reference samples to generate new data on the sampling stage. Among them, implicit generative model (IGM, Mohamed & Lakshminarayanan 2016) is the simplest one that directly mapping the reference samples to data through neural network. Sampling from IGMs requires only once forward evaluation of network. However, commonly used training algorithms for IGMs like generative adversarial networks (GANs, Goodfellow et al. 2014) meet the challenges of poor mode coverage and unstable optimization. The reason may be that, training the direct mapping to characterize the complex transport is difficult, since they lack of intermediate structural assumptions.

Recently, researchers focus on diffusion probabilistic model (DPM, Sohl-Dickstein et al. 2015; Ho et al. 2020), a well-specified probabilistic transport that constructs a generative Markov chain with its marginal distribution evolving from the Gaussian noise distribution into the data distribution. To accomplish it, DPM first gradually imposes Gaussian noise into the data samples with fixed noise scales, producing a Markov forward diffusion process. And the reversal of which, a Markov reverse process, is regarded as the learning target. DPM assumes the variance scale of each Gaussian forward kernel is small enough, leading to a Gaussian reverse process that is tractable for generative denoising process to learn. DPMs achieve impressive image generative quality even comparable with SOTA GANs (Dhariwal & Nichol, 2021). Nevertheless, the small noise scale assumption incurs quite long diffusion chains, resulting in far less efficient sampling process than IGMs.

To circumvent the small noise scale assumption, Song et al. (2021a) generalize the forward process in DPM to a non-Markovian one. The new forward process is represented by an inference process that, first infers the terminal state given data sample and then gradually infers the rest states along the reverse direction conditioned on data sample. A corresponding generative process is then constructed by replacing the conditional data sample with denoising predictions. The general process is proved to be an alternative sampling scheme for DPM. Song et al. (2021a) thus propose denoising diffusion implicit model (DDIM), an implicit variant of the general process, that can speed up 20× over DPM with similar generative quality. However, it remains inferior with fewer sampling steps.

In this work, we introduce a novel perspective on DDIM that the generative process relies heavily on the convex combination of obtained denoising predictions. Thus DDIM belongs to a general class of *ensemble denoising implicit models* whose convex coefficients can be adjusted flexibly (Sec. 3.1). It reveals the nature of each denoising step in ensemble models is predicting the denoising target with ensemble denoising prediction. Further we introduce iDDIM, an intuition guided ensemble model that allocates more trust on the latest denoising prediction based on DDIM (Sec. 3.2). Experiments on CIFAR10 indicate iDDIM largely improves on baseline DDIM especially in the case of fewer generative iterations, and convince that DDIM adopts sub-optimal convex coefficients.

However iDDIM still fails to generate realistic samples when further reducing the sampling steps. We find the reason is that the parameterization in iDDIM is unable to obtain good denoising targets with just a few denoising steps. To obtain better denoising targets, we instead propose *generalized denoising implicit model* (GDIM), a general probabilistic extension to the ensemble model that replaces the ensemble denoising prediction with a probabilistic inference conditioned on obtained states (Sec. 4). Finally we provide a specific choice that only relies on the current state radically, termed t-GDIM (Sec. 4.1). Conditional energy-based models (EBM, LeCun et al. 2006) and IGMs are used to construct t-GDIM, and are jointly trained with *variational maximum likelihood* (Grathwohl et al., 2021) (Sec. 4.2). Moreover, the iDDIM can be regarded as an ensemble augmentation trick which leverages predictions at previous steps. Experiments on various resolution image datasets show our t-GDIM+iDDIM can largely reduce the number of sampling steps to only 4, and still achieves high generative quality comparable to diffusion models or other generative models.

2 Background

DPM (Sohl-Dickstein et al., 2015) typically specifies a Markov forward diffusion process converting the data distribution $q(\mathbf{x}_0)$ into a terminal state $q(\mathbf{x}_T)$ that is closed to tractable prior $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$. It is achieved by repeated application of a Gaussian diffusion kernel $q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$, i.e., gradually imposing noise into data samples with fixed variance scales $\beta_i, i = 1, \ldots, T$. Then DPM defines a generative denoising process to simulate the reverse of the forward process with Gaussian denoising kernel $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \sigma_t^2\mathbf{I})$.

However, the feasibility of approximation comes up against commonly non-Gaussian reverse kernel $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$, unless the noise scale β_t is small enough. To keep noise scale small, DPM requires pretty long diffusion chain (∼1K steps), largely degenerating the training and sampling efficiency. To reduce the length of sampling chain, Song et al. (2021a) introduce a class of non-Markovian forward processes indexed by $\sigma \in \mathbb{R}^T_{\geq 0}$, characterized by the following inference process :

$$q_\sigma(\mathbf{x}_{1:T}|\mathbf{x}_0) = q_\sigma(\mathbf{x}_T|\mathbf{x}_0)\prod_{t=2}^{T} q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0), \quad q_\sigma(\mathbf{x}_T|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_T; \sqrt{\alpha_T}\mathbf{x}_0, (1-\alpha_T)\mathbf{I}),$$
$$q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \sqrt{\alpha_{t-1}}\mathbf{x}_0 + \sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \frac{\mathbf{x}_t - \sqrt{\alpha_t}\mathbf{x}_0}{\sqrt{1-\alpha_t}}, \sigma_t^2\mathbf{I}), \tag{1}$$

where the Gaussian form of $q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ free the forward process from Gaussian assumption. It's proved the marginal posteriors are the same as that in DPMs: $q_\sigma(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\sqrt{\alpha_t}\mathbf{x}_0, (1-\alpha_t)\mathbf{I})$, where $\alpha_t = \prod_{i=1}^t 1 - \beta_i$. Then a corresponding generative process is defined as[1]:

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \qquad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0^t), \tag{2}$$

where $\mathbf{x}_0^t = \boldsymbol{f}_\theta(\mathbf{x}_t, t)$ denotes the denoising prediction of $\mathbf{x}_0$ from $\mathbf{x}_t$ to meet with Eq. (1). Song et al. (2021a) find that training Eq. (2) with *variational inference* (Kingma & Welling, 2014) objective is equivalent to optimizing that of DPMs, from the perspective of global optimal solution. So Eq. (2) becomes a class of alternative sampling scheme to DPMs.

[1] Different from Song et al. (2021a), we set $\sigma_1 = 0$ to obtain Dirac distribution $p_\theta(\mathbf{x}_0|\mathbf{x}_1) = \delta(\mathbf{x}_0 - \mathbf{x}_0^t)$.

Specifically, Song et al. (2021a) focus on DDIM, an implicit generative process composed of deterministic transformations $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ in the case of $\sigma_t = 0$. Then DDIM is trained to fit a deterministic path from $\mathbf{x}_0$ to $\mathbf{x}_T$ characterized by the Dirac distributions (set $\sigma_t = 0$ in Eq. (1)):

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \delta(\mathbf{x}_{t-1} - \left[\sqrt{\alpha_{t-1}}\mathbf{x}_0 + \sqrt{1-\alpha_{t-1}} \cdot \frac{\mathbf{x}_t - \sqrt{\alpha_t}\mathbf{x}_0}{\sqrt{1-\alpha_t}}\right]). \tag{3}$$

Please see Appendix A for more detailed review and discussion.

3 Ensemble of Denoising Predictions

Thanks to the non-Markovian inference process, DDIM can speed up 20× over denoising DPM (DDPM, Ho et al. 2020) with similar high performance, it nevertheless degenerates when the number of sampling steps is no more than 20. In order to mitigate it, in this section, we explore how the denoising predictions $\mathbf{x}_0^{1:T}$ are leveraged to accomplish the generative process, as $\mathbf{x}_0^t = \boldsymbol{f}_\theta(\mathbf{x}_t, t)$ is the key for implementing $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$. Our key observation is that each $\mathbf{x}_{t-1}$ along the generative process in DDIM depends on a specific convex combination of $\mathbf{x}_0^{t:T}$. This leads to a general denoising implicit model which is an ensemble of the denoising predictions $\mathbf{x}_0^{1:T}$. Then the experiments on CIFAR10 indicate the coefficients used in DDIM are not optimal especially when T is small.

3.1 Ensemble Denoising Implicit Models

To show our novel perspective, let us define the *ensemble denoising implicit models* indexed by $\omega_t = [\omega_t^t, \ldots, \omega_t^T] \in \mathbb{R}_{\geq 0}^{T-t+1}$, characterized by the deterministic transformation $q_\omega(\mathbf{x}_{t-1}|\mathbf{x}_0^{t:T}, \mathbf{x}_T)$:

$$\mathbf{x}_{t-1} = B_{t-1} \cdot \sum_{k=t}^{T} \frac{\omega_t^k}{\sum_{k=t}^T \omega_t^k} \mathbf{x}_0^k + C_{t-1}\mathbf{x}_T = B_{t-1}\bar{\mathbf{x}}_0^t + C_{t-1}\mathbf{x}_T, \tag{4}$$

where $\bar{\mathbf{x}}_0^t$ denotes the convex ensemble of denoising predictions $\mathbf{x}_0^{t:T}$, and B_{t-1}, C_{t-1} are set to:

$$B_{t-1} = \sqrt{\alpha_{t-1}} - \sqrt{1-\alpha_{t-1}}\frac{\sqrt{\alpha_T}}{\sqrt{1-\alpha_T}}, \qquad C_{t-1} = \frac{\sqrt{1-\alpha_{t-1}}}{\sqrt{1-\alpha_T}} \tag{5}$$

to match up with the inference process (3) as shown later. We find the DDIM denoising kernel represented by the deterministic transformation $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0^t)$ (in the case of $\sigma_t = 0$ in Eq. (2)):

$$\mathbf{x}_{t-1} = \sqrt{\alpha_{t-1}}\mathbf{x}_0^t + \sqrt{1-\alpha_{t-1}} \cdot \frac{\mathbf{x}_t - \sqrt{\alpha_t}\mathbf{x}_0^t}{\sqrt{1-\alpha_t}}, \tag{6}$$

is a linear combination of $\mathbf{x}_t$ and $\mathbf{x}_0^t$. Since $\mathbf{x}_t$ is also a combination of $\mathbf{x}_{t+1}$ and $\mathbf{x}_0^{t+1}$, we can recursively expand the particles $\mathbf{x}_k$ along $t \to T$ and obtain the following result:

Proposition 1. *Denoising diffusion implicit model (6) can be reformulated as $q(\mathbf{x}_{t-1}|\mathbf{x}_0^{t:T}, \mathbf{x}_T)$:*

$$\mathbf{x}_{t-1} = \sqrt{1-\alpha_{t-1}} \cdot \sum_{k=t}^{T} (A_{k-1} - A_k)\, \mathbf{x}_0^k + \frac{\sqrt{1-\alpha_{t-1}}}{\sqrt{1-\alpha_T}}\mathbf{x}_T, \quad A_k = \frac{\sqrt{\alpha_k}}{\sqrt{1-\alpha_k}}, \tag{7}$$

and is a specific instance of ensemble denoising implicit models (4) with $\omega_t^k = A_{k-1} - A_k$.

We include a general proof in Appendix B.1. Proposition 1 demonstrates that the ensemble denoising implicit models are generalized DDIMs, and they are all first computing $\bar{\mathbf{x}}_0^t$ with a convex combination of $\mathbf{x}_0^{t:T}$ and then using linear combination with $\mathbf{x}_T$ to obtain $\mathbf{x}_{t-1}$. But which $\bar{\mathbf{x}}_0^t$ is the best for the general ensemble models? To answer this, we notice in DDIM, the denoising predictions $\mathbf{x}_0^{t:T}$ are trained to approximate the same real sample $\mathbf{x}_0 \sim q(\mathbf{x}_0|\mathbf{x}_T)$ given $\mathbf{x}_{t:T}$. So that if we let $\mathbf{x}_0^{t:T} = \mathbf{x}_0$ in the ensemble denoising model (4), it turns into $q(\mathbf{x}_{t-1}|\mathbf{x}_0, \mathbf{x}_T)$:

$$\mathbf{x}_{t-1} = B_{t-1}\mathbf{x}_0 + C_{t-1}\mathbf{x}_T = \sqrt{\alpha_{t-1}}\mathbf{x}_0 + \sqrt{1-\alpha_{t-1}} \cdot \frac{\mathbf{x}_T - \sqrt{\alpha_T}\mathbf{x}_0}{\sqrt{1-\alpha_T}}. \tag{8}$$

Equation (8) forms a deterministic path between $\mathbf{x}_0$ and $\mathbf{x}_T$, which is exactly the inference process in DDIM (3), but is rewritten into a more proper equivalent form:

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = q(\mathbf{x}_T|\mathbf{x}_0) \prod_{t=2}^{T} q(\mathbf{x}_{t-1}|\mathbf{x}_0, \mathbf{x}_T). \tag{9}$$

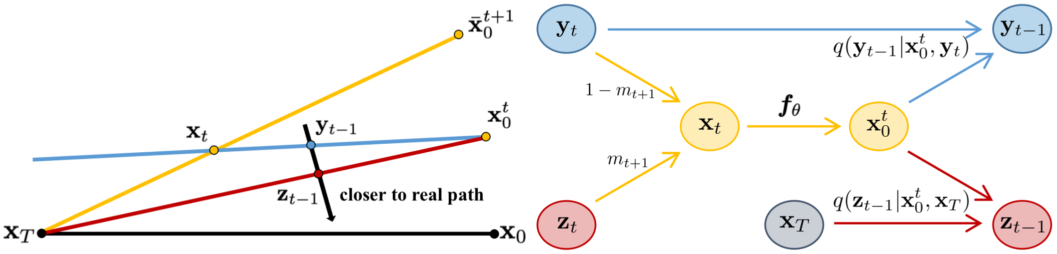

Figure 1: Design intuition (left) and graphical description (right) for improved DDIM. The ensemble model (yellow line) uses linear combination of $\bar{\mathbf{x}}_0^{t+1}$ and $\mathbf{x}_T$ to obtain $\mathbf{x}_t$. Then, DDIM (blue line) takes a denoising step along the deterministic path from $\mathbf{x}_t$ to $\mathbf{x}_0^t$, while radical ensemble model (red line) along the path from $\mathbf{x}_T$ to $\mathbf{x}_0^t$. If $\mathbf{x}_0^t$ is closer to some real $\mathbf{x}_0$ than $\bar{\mathbf{x}}_0^{t+1}$, the current denoising step of radical ensemble model (red line) becomes closer to the real path.

So in other words, the ensemble denoising implicit models are trained to fit the same target process as in DDIM, providing a class of alternative sampling schemes. And what's more, ensemble models (include DDIM) are essentially predicting the real sample $\mathbf{x}_0$ with $\bar{\mathbf{x}}_0^t$ at each denoising step, by means of leveraging the convex ensemble of the denoising predictions $\mathbf{x}_0^{t:T}$. However in practice, $\mathbf{x}_0^{t:T}$ are commonly different, let alone be equal to $\mathbf{x}_0 \sim q(\mathbf{x}_0|\mathbf{x}_T)$. This leads to the ensemble denoising prediction $\bar{\mathbf{x}}_0^t$ not always be like some $\mathbf{x}_0$. In this work, the proposed ensemble denoising implicit model provides a flexible way to combine $\mathbf{x}_0^{t:T}$ for better ensemble prediction $\bar{\mathbf{x}}_0^t$, potentially results in a generative process closer to the target inference process.

3.2 Sub-optimal Coefficients in DDIM

However, finding out the optimal coefficients in ensemble models is intractable as we know nothing about how the performance of $\mathbf{x}_0^{t:T}$ contributes to the additive ensemble prediction $\bar{\mathbf{x}}_0^t$. Since denoising $\mathbf{x}_k$ becomes more difficult along $k \to T$, the latest denoising prediction $\mathbf{x}_0^t$ is intuitively more precise than $\mathbf{x}_0^{t+1:T}$ and thus more precise than $\bar{\mathbf{x}}_0^{t+1}$. If we let $\omega_t = [1, 0, \ldots, 0]$, the ensemble model (4) will only trust the latest $\mathbf{x}_0^t$ radically, and the resulting radical ensemble model is presented as $q(\mathbf{x}_{t-1}|\mathbf{x}_0^t, \mathbf{x}_T)$:

$$\mathbf{x}_{t-1} = B_{t-1}\mathbf{x}_0^t + C_{t-1}\mathbf{x}_T = \sqrt{\alpha_{t-1}}\mathbf{x}_0^t + \sqrt{1-\alpha_{t-1}} \cdot \frac{\mathbf{x}_T - \sqrt{\alpha_T}\mathbf{x}_0^t}{\sqrt{1-\alpha_T}}. \tag{10}$$

Inspired by the intuition (Fig. 1, left) that trusting more on $\mathbf{x}_0^t$ may bring about a generative process closer to a real one, we introduce an improved DDIM (iDDIM) where $\mathbf{y}_{t-1}$ and $\mathbf{z}_{t-1}$ are computed with Eqs. (6) and (10) respectively (Fig. 1, right):

$$\mathbf{x}_{t-1} = (1 - m_t)\mathbf{y}_{t-1} + m_t\mathbf{z}_{t-1}. \tag{11}$$

Equation (11) actually comes from replacing $m_t \in [0, 1]$ proportion of $\mathbf{x}_0^{t+1:T}$ with $\mathbf{x}_0^t$ in DDIM, so it is still an ensemble denoising implicit model. We include derivations in Appendix B.2. The iDDIM behaves as an interpolation between DDIM (6) and radical ensemble model (10). And as $m_t \to 1$, it allocates more trust on $\mathbf{x}_0^t$ as expected.

In Sec. 6.1, we conduct experiments on CIFAR10 to explore how the performance of iDDIM influenced by varying m_t. The results demonstrates that, DDIM adopts sub-optimal coefficients and allocating higher proportion (ω_t^t) on $\mathbf{x}_0^t$ achieves prominent improvement especially as T decreasing.

4 Generalized Denoising Implicit Models

As we emphasized in the previous section, in order to fit the deterministic inference process characterized by Eq. (9), the ensemble denoising implicit models (4) are actually predicting $\mathbf{x}_0$ with the ensemble prediction $\bar{\mathbf{x}}_0^t$ at each step. Thus the performance of generative process largely rests with the alignment between $\bar{\mathbf{x}}_0^t$ and $\mathbf{x}_0$. We have verified that carefully selecting the coefficients ω_t does produce better $\bar{\mathbf{x}}_0^t$, however, the misalignment between $\bar{\mathbf{x}}_0^t$ and $\mathbf{x}_0$ still remains and is exacerbated when T further reducing (see Fig. 4). It is because $\mathbf{x}_0^t = \boldsymbol{f}_\theta(\mathbf{x}_t, t)$ is a Dirac approximation of

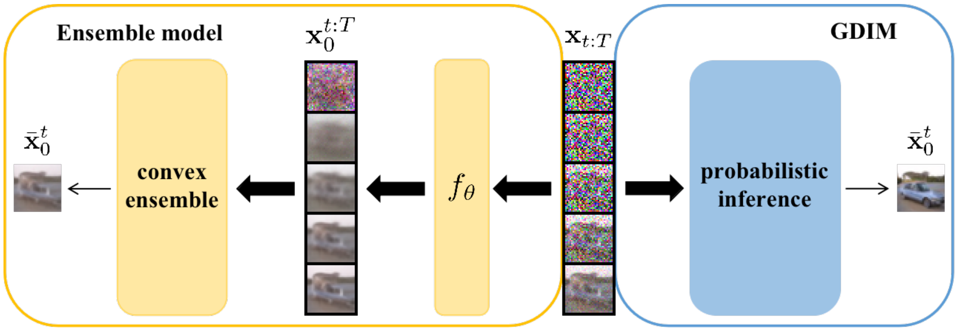

Figure 2: Denoising target $\bar{\mathbf{x}}_0^t$ in the ensemble denoising implicit model and the generalized denoising implicit model. Ensemble model uses convex combination of potentially inferior $\mathbf{x}_0^{t:T}$ to obtain blurry $\bar{\mathbf{x}}_0^t$, while GDIM leverages probabilistic inference $p_\theta(\bar{\mathbf{x}}_0^t|\mathbf{x}_{t:T})$ to get better $\bar{\mathbf{x}}_0^t$ directly.

$q(\mathbf{x}_0^t|\mathbf{x}_t)$, i.e., $p_\theta(\mathbf{x}_0^t|\mathbf{x}_t) = \delta(\mathbf{x}_0^t - \boldsymbol{f}_\theta(\mathbf{x}_t, t))$. As shown in Xiao et al. (2022), this deterministic parameterization struggles with the commonly multimodal $q(\mathbf{x}_0^t|\mathbf{x}_t)$ as $t \to T$, brings about potentially inferior $\mathbf{x}_0^t$ (see Fig. 2 for instance). As a result, ensemble models require more steps and carefully coefficients seeking to get gradually better $\bar{\mathbf{x}}_0^t$ along the generative process. In order to obtain more realistic $\bar{\mathbf{x}}_0^t$ at each denoising step, we propose to replace the convex ensemble of $\mathbf{x}_0^{t:T}$ with probabilistic inference conditioned on $\mathbf{x}_{t:T}$, i.e., $p_\theta(\bar{\mathbf{x}}_0^t|\mathbf{x}_{t:T})$. This leads to the generalized denoising implicit model (GDIM):

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_{t:T}),\ p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_{t:T}) = \int p_\theta(\bar{\mathbf{x}}_0^t|\mathbf{x}_{t:T})q(\mathbf{x}_{t-1}|\bar{\mathbf{x}}_0^t, \mathbf{x}_T)\mathrm{d}\bar{\mathbf{x}}_0^t. \quad (12)$$

The GDIM is a general extension to the ensemble models as they share the same spirit that, first predicting a current denoising target $\bar{\mathbf{x}}_0^t$ given obtained states $\mathbf{x}_{t:T}$ and then taking one denoising step to $\mathbf{x}_{t-1}$ via deterministic transform (8). See Fig. 2 for comparison. While the benefit is that GDIM can directly predicts $\bar{\mathbf{x}}_0^t$ via $p_\theta(\bar{\mathbf{x}}_0^t|\mathbf{x}_{t:T})$ represented by some expressive probabilistic models, and thus the dependence on $\mathbf{x}_{t:T}$ (corresponding to the coefficients in ensemble models) is learned adaptively. More importantly, if $p_\theta(\bar{\mathbf{x}}_0^t|\mathbf{x}_{t:T})$ is properly trained to generate good $\bar{\mathbf{x}}_0^t$, the denoising process will no longer need many steps as that in ensemble models.

4.1 Radical GDIM

Notice the GDIM is autoregressive and enables us to flexible choose which $\mathbf{x}_{t:T}$ does $p_\theta(\bar{\mathbf{x}}_0^t|\mathbf{x}_{t:T})$ conditioned on. We next discuss a specific choice that only relies on $\mathbf{x}_t$ radically, i.e., $p_\theta(\mathbf{x}_0^t|\mathbf{x}_t)$. It seems to be a probabilistic counterpart to the radical ensemble model (10) and is termed t-GDIM:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_T) = \int p_\theta(\mathbf{x}_0^t|\mathbf{x}_t)q(\mathbf{x}_{t-1}|\mathbf{x}_0^t, \mathbf{x}_T)\mathrm{d}\mathbf{x}_0^t, \quad (13)$$

We adopt expressive conditional energy-based model (EBM) to represent the denoising distribution:

$$p_\theta(\mathbf{x}_0^t|\mathbf{x}_t) = \frac{\exp(-E_\theta(\mathbf{x}_0^t, \mathbf{x}_t))}{\int \exp(-E_\theta(\mathbf{x}_0^t, \mathbf{x}_t))\mathrm{d}\mathbf{x}_0^t} = \frac{\exp(-E_\theta(\mathbf{x}_0^t, \mathbf{x}_t))}{Z(\theta, \mathbf{x}_t)}, \quad (14)$$

where $E_\theta : \mathcal{X} \times \mathcal{X} \times \mathbb{R} \to \mathbb{R}$ denotes the joint energy over $\mathbf{x}_0^t$ and $\mathbf{x}_t$, and the dependence on t is not displayed for brevity. The same inference process as that in ensemble models (9) is regarded as the learning target for t-GDIM. Then we again optimize θ with the *variational inference* (Kingma & Welling, 2014) objective:

$$\begin{aligned} &-\mathbb{E}_{q(\mathbf{x}_0)}[\log p_\theta(\mathbf{x}_0)] \le \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\right] \\ &\doteq \sum_{t=1}^{T} \mathbb{E}_{q(\mathbf{x}_0, \mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{x}_T)}\left[E_\theta(T^{-1}(\mathbf{x}_{t-1}; \mathbf{x}_T), \mathbf{x}_t) + \log Z(\theta, \mathbf{x}_t)\right], \end{aligned} \quad (15)$$

where the *diffeomorphism* $T(\mathbf{x}_0; \mathbf{x}_T) = \mathbf{x}_{t-1}$ stands for the deterministic transform $q(\mathbf{x}_{t-1}|\mathbf{x}_0, \mathbf{x}_T)$. For convenience, we use $\mathcal{J}(\theta, t)$ to represent the energy objective at each time step t. However, computing $Z(\theta, \mathbf{x}_t)$ requires intractable integral over the whole space, and fortunately, a more efficient alternate is to estimate the optimizing gradients:

$$\nabla_\theta \mathcal{J}(\theta, t) = \mathbb{E}_{q(\mathbf{x}_0)q(\mathbf{x}_t|\mathbf{x}_0)p_\theta(\mathbf{x}_0^t|\mathbf{x}_t)} \left[\nabla_\theta E_\theta(\mathbf{x}_0, \mathbf{x}_t) - \nabla_\theta E_\theta(\mathbf{x}_0^t, \mathbf{x}_t)\right]. \tag{16}$$

Notice it resembles the gradient of *maximum likelihood* objective for learning EBMs (LeCun et al., 2006), and an unbiased Monte Carlo gradient estimator can be accomplished by sampling a batch of $(\mathbf{x}_0, \mathbf{x}_t, \mathbf{x}_0^t)$ at each training iteration. See Appendix C.1 for derivations.

4.2 Sampling from Energy-based Denoising Distribution

The gradient estimator commonly suffers from sampling from unnormalized distributions, e.g. Eq. (14). Recent attempts (Du & Mordatch, 2019; Nijkamp et al., 2020b) resort to dynamic-based Markov chain Monte Carlo (MCMC), but fall into the trouble of mixing and again requires lots of sampling steps. Here we amortize the MCMC sampling into training conditional IGMs constructed by $\mathbf{x}_0^t = G_\phi(\mathbf{u}; \mathbf{x}_t, t), \mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. In other word, the expressive conditional IGMs are used as approximate samplers $p_\phi(\mathbf{x}_0^t|\mathbf{x}_t) = \int \delta(\mathbf{x}_0^t - G_\phi(\mathbf{u}; \mathbf{x}_t, t))\mathcal{N}(\mathbf{u}; \mathbf{0}, \mathbf{I})\mathrm{d}\mathbf{u}$, and are trained by minimizing the following KL divergence with respect to ϕ (see Appendix C.2 for derivations):

$$D_{\mathrm{KL}}(p_\phi(\mathbf{x}_0^t|\mathbf{x}_t)||p_\theta(\mathbf{x}_0^t|\mathbf{x}_t)) \doteq \mathbb{E}_{\mathcal{N}(\mathbf{u};\mathbf{0},\mathbf{I})}\left[E_\theta(G_\phi(\mathbf{u}; \mathbf{x}_t), \mathbf{x}_t)\right] - \mathcal{H}(p_\phi(\mathbf{x}_0^t|\mathbf{x}_t)). \tag{17}$$

This variational approximation can be incorporated in $\mathcal{J}(\theta, t)$ as an additional inner optimization which is similar to the *variational maximum likelihood* (Grathwohl et al., 2021). And the resulting nested objective is commonly handled with alternating optimization. Notice that $\mathcal{H}(p_\phi(\mathbf{x}_0^t|\mathbf{x}_t))$ is the entropy of sampler and is typically difficult to optimize. If we ignore this entropy term, the nested optimization becomes similar to WGAN (Arjovsky et al., 2017):

$$\min_\theta \max_\phi \left\{\mathbb{E}_{q(\mathbf{x}_0,\mathbf{x}_t)\mathcal{N}(\mathbf{u};\mathbf{0},\mathbf{I})}\Big[E_\theta\big(\mathbf{x}_0, \mathbf{x}_t\big) - E_\theta\big(G_\phi(\mathbf{u}; \mathbf{x}_t), \mathbf{x}_t\big)\Big]\right\}. \tag{18}$$

Therefore, we can borrow the proven optimizing technique from GANs for jointly training the conditional EBM $p_\theta(\mathbf{x}_0^t|\mathbf{x}_t)$ and its variational sampler $p_\phi(\mathbf{x}_0^t|\mathbf{x}_t)$.

5 Related Work and Discussion

Score-based generative model (SGM). As shown to have interesting connection with denoising score matching (Vincent, 2011), DPMs as well as noise conditional score networks (NCSN, Song & Ermon 2019; 2020) are usually referred to together as SGMs. Song et al. (2021c) further propose a unified forward-reverse stochastic differential equation (SDE) framework that treats them as discretizations of specific SDEs. After that, lots of works explore the intrinsic properties or numerical approximations of different SDEs to improve the generative quality and the sampling efficiency (Dockhorn et al., 2022; Jolicoeur-Martineau et al., 2021a; Lu et al., 2022; Liu et al., 2022). However, they still generate inferior samples when further reduce the number of sampling steps. It is possibly because the numerical simulation for SDEs always assumes the discretization steps are small, and the case of only few sampling iterations violates the assumption.

Accelerate sampling. Besides, there are lots of other studies focusing on accelerating the sampling process for DPMs (Kong & Ping, 2021; Watson et al., 2021; Lyu et al., 2022; Nachmani et al., 2021; Zheng et al., 2022). They are all basically orthogonal to us since they do not involve the inherent understanding of how DPMs (or DDIMs) leverage the denoising predictions $\mathbf{x}_0^{t:T}$. Watson et al. (2022) propose a similar method to our ensemble model, which use a combination of obtained state $\mathbf{x}_{t:T}$ to represent DPMs, and learn the coefficients by differentiating through the sample quality. But we point out that the denoising predictions rather than obtained states are the keys in nature.

Discussion. Notice the objective for t-GDIM (18) does not depend on $\mathbf{x}_T$, and $p_\theta(\mathbf{x}_0^t|\mathbf{x}_t)$ at different step t are trained to approximate corresponding $q(\mathbf{x}_0|\mathbf{x}_t)$ rather than the same $q(\mathbf{x}_0|\mathbf{x}_T)$. It implies that t-GDIM may be not always going to fit the deterministic inference process (9) as GDIM supposed to. Notice the same case can also be found in DDIM, so iDDIM can be useful in t-GDIM as an ensemble augmentation for denoising targets. A potential solution to the missing dependence on $\mathbf{x}_T$ is to incorporate explicit condition, i.e., $p_\theta(\mathbf{x}_0^t|\mathbf{x}_t, \mathbf{x}_T)$, but may incur additional input for networks. Or we can force the training of $p_\theta(\mathbf{x}_0^t|\mathbf{x}_t)$ to depend on $\mathbf{x}_T$ implicitly. Furthermore, we find the models used by Gao et al. (2021); Xiao et al. (2022) have similar spirit to t-GDIM, but ours follows a distinct theoretical route. Inspired by them, we explore feasible methods to introduce the dependence explicitly or implicitly. Please see Appendices D and F for more discussions.

	T	4	10	20	50	100	1000
m	0.0	37.82	13.74	7.55	4.78	4.14	3.88
	0.1	37.50	12.03	6.12	3.82	**3.55**	**3.59**
	0.2	37.02	10.74	5.29	**3.82**	4.05	4.68
	0.3	36.50	9.71	**5.05**	4.97	6.04	-
	0.4	36.56	**8.84**	5.51	8.16	10.74	-
	1.0	**35.72**	11.99	56.60	112.23	-	-
DDPM		-	-	137.77	35.29	10.61	**3.19**

Table 1: CIFAR10 image generation measured in FID↓. m denotes the replacing ratio in iDDIM. $m = 0.0$ and $m = 1.0$ represent DDIM and radical ensemble model respectively. Sampling process uses the self-trained DDPM score predictor.

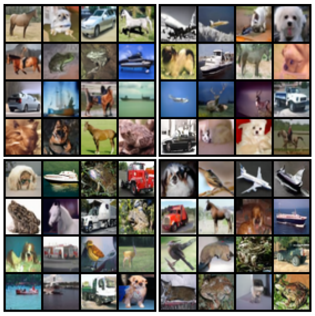

Figure 3: Samples of iDDIM with $T = 10, 20, 50, 100$ and best m^*.

6 Experiments

In this section, we conduct experiments to verify our two claims: **1).** The DDIMs are ensemble denoising implicit models with sub-optimal convex coefficients. With the intuition that the latest denoising prediction $\mathbf{x}_0^t$ is more precise, our iDDIM provides a simple but effective way to seek better coefficients. **2).** Our t-GDIM can largely reduce the number of denoising steps to just a few, but still achieve comparable generative quality to more expensive diffusion-based models.

Datasets and metrics. For our iDDIMs, we conduct extensive experrments on CIFAR10 (Krizhevsky, 2009) for comparison. For our t-GDIMs which are trained similarly to conditional GAN, we additionally consider CelebA (Liu et al., 2015) and CelebAHQ (Karras et al., 2018) with higher resolutions. We resize the images in CelebA to 64 × 64 and 128 × 128, termed CelebA-64 and CelebA-128 respectively. Besides, we resize the CelebAHQ to 256 × 256. For all datasets, only the random horizontal flipping is used for pre-processing. We use the image generation quality to characterize the performance of different methods. The image generation quality on CIFAR10 is evaluated by Frechet inception distance (FID, Heusel et al. 2017) and Inception Score (IS, Salimans et al. 2016). For higher resolutions, only FID is reported since IS is not proper.

Generative process. For all experiments, we use 1000-step linear noise schedule in DDPM to construct the complete diffusion process. Following Song et al. (2021a), quadratic timesteps selection is used to construct the generative sub-sequence. We consider various T for iDDIM experiments, while only adopt $T = 4$ for the t-GDIM part to evaluate its performance on few sampling steps. Please see Appendix E.1 for the architecture of models or complementary experimental details.

6.1 Seeking Better Coefficients

For simplicity, we choose $m_t = m$ for all t in iDDIM. In Tab. 1, we show the generation quality of our iDDIM trained on CIFAR10. We find that DDIM ($m = 0.0$) performs worse than iDDIM consistently for each T, if the ratio m is properly increased. More interestingly, the sample quality further becomes better when we choose higher m as T decreasing, but overly trusting the latest $\mathbf{x}_0^t$ ($m \to 1.0$) leads to worse quality and the radical ensemble model ($m = 1.0$) performs bad.

In Fig. 4, we display the ensemble denoising prediction $\bar{\mathbf{x}}_0^t$ at each sampling step of the 20-step and 4-step iDDIM generative processes for CIFAR10 image. It shows that, though $\bar{\mathbf{x}}_0^t$ are becoming more realistic as t decreasing, the final prediction of DDIM ($m = 0.0$) is still blurry. With increasing m, the ensemble prediction at each step becomes much clearer, but with too high m, especially when $m = 1.0$ (the radical ensemble model), the predictions becomes distorted. Besides, in 4-step sampling process, all the ensemble predictions $\bar{\mathbf{x}}_0^{1:T}$ are blurry regardless of m. These results suggest the coefficients used by DDIM is sub-optimal, and our iDDIM with proper m leads to better generative quality, nevertheless still fails when T is too small.

In Tab. 2, we report the generation quality of iDDIM with best tuned m^*, and compare iDDIM with recent proposed impressive methods for accelerating sampling. For a fair comparison, we report the results of other methods that have similar settings to ours. It demonstrates that our iDDIM achieves the best result among baseline methods in the case of $T = 10$ and $T = 20$, though iDDIM is much simpler than others. When T is larger, iDDIM is slightly worse than FastDPM. It indicates that we

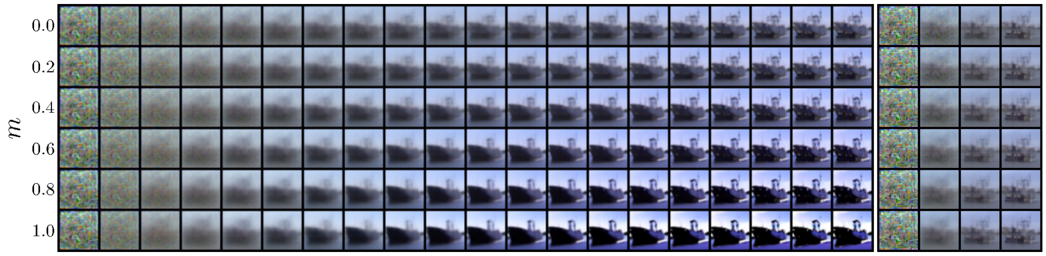

Figure 4: Ensemble denoising predictions $\bar{\mathbf{x}}_0^{1:T}$ in 20-step (left) and 4-step (right) iDDIM sampling process with varying m.

T	10		20		50		100	
m^*	0.6		0.3		0.15		0.1	
IS↑ FID↓	**8.85**	**8.24**	9.09	**5.05**	**9.27**	3.61	9.21	3.55
DDIM (Song et al., 2021a)	8.28	13.74	8.81	7.55	8.98	4.78	9.11	4.14
DDPM (Ho et al., 2020)	-		3.98	137.77	8.53	35.29	**9.45**	10.61
GGDM (Watson et al., 2022)	**8.84**	**8.23**	**9.18**	5.57	-		-	
FastDPM (Kong & Ping, 2021)	-	9.90	-	5.22	8.98	**3.41**	-	**3.01**
Analytic-DDIM (Bao et al., 2022)	-	14.00	-	5.81	-	4.04	-	3.55

Table 2: The best m^* for iDDIM on CIFAR10, searched by traversing $[0, 1]$ with 0.05 intervals.

require more complexly coefficients seeking since more denoising predictions are involved. Figure 3 presents some randomly generated CIFAR10 samples by our iDDIM with the best m^*.

6.2 Sample Quality in GDIM

For an overall evaluation of the proposed t-GDIM, Figs. 5 and 6 presents the qualitative samples which are generated with only 4 sampling steps. These images are of high fidelity consistently.

In Tab. 3, we present our quantitative results on CIFAR10. Here we report the FID of our t-GDIM, with iDDIM as an ensemble augmentation technique for the prediction of denoising target $\bar{\mathbf{x}}_0^t$. As discussed in Sec. 5, iDDIM can improve the quality of t-GDIM marginally. We provide related studies in Appendix E.2. When comparing with score-based models, our t-GDIM can largely reduce the number of function evaluations (NFE) to only 4, while achieve comparable quality. When comparing with GANs, we find our models surpass most of SOTA GANs, though we do not use any data augmentation technique. Notice DDGAN is superior to ours. We suggest the reason is that: the optimization method of t-GDIM is similar to GAN which ignores the entropy term of variational sampler in *variational maximum likelihood* (17) in theory, leading to poor mode coverage in practice. Please see Appendix E.2 for qualitative results. While DDGAN is based on the DPM framework, and as a result, additional noises are introduced during training, which is an important data augmentation method for GAN-based optimization (see Appendix F.2). These imply the proven augmentation techniques for GAN-based optimization are useful for our t-GDIM and potentially improve the generation quality. But here we report the pure version of the proposed t-GDIM, and

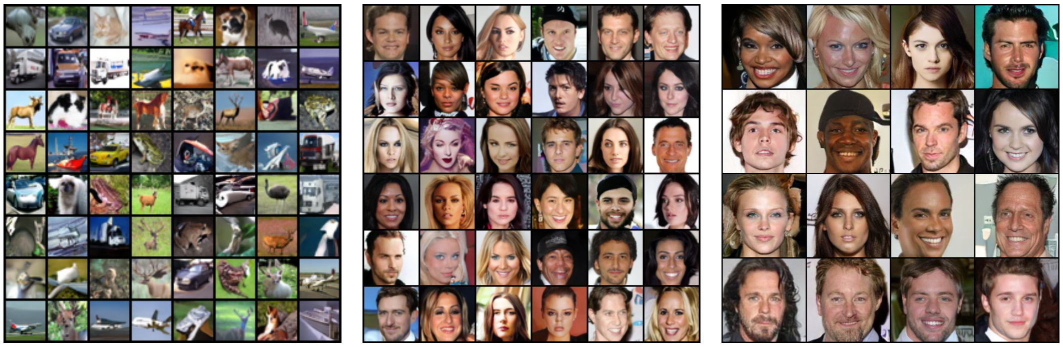

Figure 5: Qualitative samples of t-GDIM. Left: CIFAR10. Middle: CelebA-64. Right: CelebA-128.

Method	IS↑	FID↓	NFE↓
Improved DDPM (Nichol & Dhariwal, 2021)	-	2.90	4000
UDM (Kim et al., 2021)	10.1	2.33	2000
Likelihood SDE (Song et al., 2021b)	-	2.87	2000
Score SDE (VE) (Song et al., 2021c)	9.89	2.20	2000
DDPM (Ho et al., 2020)	9.47	3.19	1000
NCSN (Song & Ermon, 2019)	8.87	25.3	1000
Adversarial DSM (Jolicoeur-Martineau et al., 2021b)	-	6.10	1000
VDM (Kingma et al., 2021)	-	4.00	1000
Recovery EBM, $T6$ (Gao et al., 2021)	8.30	9.58	180
Gotta Go Fast (Jolicoeur-Martineau et al., 2021a)	-	2.44	180
LSGM (Vahdat et al., 2021)	9.87	2.10	147
CLD-SGM (Prob. Flow) (Dockhorn et al., 2022)	-	2.71	147
Probability Flow (VP) (Song et al., 2021c)	9.83	3.08	140
DiffuseVAE, $T = 100$ (Pandey et al., 2022)	8.27	11.71	100
FastDPM, $T = 50$ (Kong & Ping, 2021)	8.98	3.41	50
F-PNDM, $T = 50$ (Liu et al., 2022)	-	3.68	50
gDDIM, $T = 50$ (Zhang et al., 2022)	-	2.28	50
SNGAN+DGflow (Ansari et al., 2021)	9.35	9.62	25
DDGAN, $T = 4$ (Xiao et al., 2022)	9.63	3.75	4
DDPM Distillation (Luhman & Luhman, 2021)	8.36	9.36	1
AutoGAN (Gong et al., 2019)	8.60	12.4	1
TransGAN (fan Jiang et al., 2021)	9.02	9.26	1
StyleGAN2 w/o ADA (Karras et al., 2020a)	9.18	8.32	1
StyleGAN2 w/ ADA (Karras et al., 2020a)	9.83	2.92	1
StyleGAN2 w/ Diffaug (Zhao et al., 2020)	9.40	5.79	1
t-GDIM (ours)	9.55	5.51	4
t-GDIM+iDDIM, $m = 0.7$ (ours)	9.50	5.24	4

Table 3: CIFAR10 image generation measured in IS↑ and FID↓.

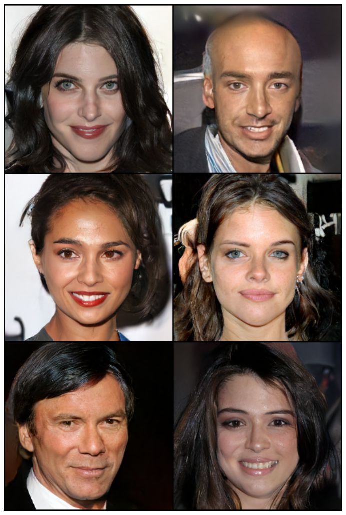

Figure 6: Qualitative samples of t-GDIM on CelebAHQ-256.

Method	CelebA-64	CelebA-128
DDPM (Ho et al., 2020)	3.26	5.65
Recovery EBM, $T6$ (Gao et al., 2021)	5.98	-
F-PNDM, $T = 10$ (Liu et al., 2022)	7.71	-
StyleGAN2+ES-DDPM (Lyu et al., 2022)	9.15	6.15
TDPM-GAN, $T = 4$ (Zheng et al., 2022)	3.96	-
COCO-GAN (Lin et al., 2019)	4.00	5.74
t-GDIM+iDDIM, $m = 0.6$ (ours)	**2.93**	**4.04**

Table 4: CelebA-64 and CelebA-128 image generation measured in FID↓.

Method	FID↓
Score SDE (Song et al., 2021c)	7.23
LSGM (Vahdat et al., 2021)	7.22
UDM (Kim et al., 2021)	**7.16**
VAEBM (Xiao et al., 2021)	20.4
PGGAN (Karras et al., 2018)	8.03
VQ-GAN (Esser et al., 2021)	10.2
DDGAN (Xiao et al., 2022)	7.64
t-GDIM+iDDIM, $m = 0.4$ (ours)	7.26

Table 5: CelebAHQ-256 image generation measured in FID↓.

leave them for future work. Table 4 presents the quantitative results on CelebA-64 and CelebA-128. When comparing our model with recent few-step diffusion-based models, we find t-GDIM achieve the best quality among baseline methods with similar NFE. Surprisingly, it even surpass 1000-step DDPM especially on 128×128 resolution. Besides, we report FID on CelebAHQ-256 in Tab. 5. We find our model can still achieve competitive generative performance among SOTA models, and performs marginally better than DDGAN. The results show the potential to apply t-GDIM on more complex and higher resolutions.

7 Conclusion

We have provided an insightful perspective that DDIM is a specific instance of our *ensemble denoising implicit model* with sub-optimal convex coefficients. This explains why DDIM fails to achieve good generation quality with fewer sampling steps. Our iDDIM is an intuition guided modification on DDIM which simply allocates more trust on the latest denoising prediction, but can improve on DDIM largely. To further decrease the sampling steps, we propose GDIM, a general extension to ensemble model, that replaces the additive ensemble of denoising predictions to a principled probabilistic inference. Then the variational maximum likelihood is used to train t-GDIM, a specific GDIM only conditioned on the latest state at each step, and derivates more favorable GAN-based optimization methods. Extensive experiments demonstrate t-GDIM can reduce the number of sampling steps to only 4 while achieve comparable performance to other generative models. It also shows the potential to apply t-GDIM on higher resolutions, where we leave it for future work.

REFERENCES

Abdul Fatir Ansari, Ming Liang Ang, and Harold Soh. Refining deep generative models via discriminator gradient flow. In *International Conference on Learning Representations*, 2021. 9

Martín Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017. 6

Fan Bao, Chongxuan Li, Jun Zhu, and Bo Zhang. Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. *International Conference on Learning Representations*, 2022. 8

Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 2021. 1

Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Score-based generative modeling with critically-damped langevin diffusion. *International Conference on Learning Representations*, 2022. 6, 9

Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. In *Advances in Neural Information Processing Systems*, 2019. 6, 19

Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12868–12878, 2021. 9

Yi fan Jiang, Shiyu Chang, and Zhangyang Wang. Transgan: Two transformers can make one strong gan. *ArXiv*, abs/2102.07074, 2021. 9

Ruiqi Gao, Yang Song, Ben Poole, Ying Nian Wu, and Diederik P. Kingma. Learning energy-based models by diffusion recovery likelihood. *International Conference on Learning Representations*, 2021. 6, 9, 21

Xinyu Gong, Shiyu Chang, Yi fan Jiang, and Zhangyang Wang. Autogan: Neural architecture search for generative adversarial networks. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3223–3233, 2019. 9

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014. 1

Will Grathwohl, Jacob Kelly, Milad Hashemi, Mohammad Norouzi, Kevin Swersky, and David Kristjanson Duvenaud. No mcmc for me: Amortized sampling for fast and stable training of energy-based models. In *International Conference on Learning Representations*, 2021. 2, 6, 20

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, 2017. 7

Jonathan Ho, Ajay Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 2020. 1, 3, 8, 9, 14, 22

Alexia Jolicoeur-Martineau, Ke Li, Remi Piche-Taillefer, Tal Kachman, and Ioannis Mitliagkas. Gotta go fast when generating data with score-based models. *ArXiv*, abs/2105.14080, 2021a. 6, 9

Alexia Jolicoeur-Martineau, Remi Piche-Taillefer, Rémi Tachet des Combes, and Ioannis Mitliagkas. Adversarial score matching and improved sampling for image generation. *International Conference on Learning Representations*, 2021b. 9

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *International Conference on Learning Representations*, 2018. 7, 9

Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in Neural Information Processing Systems*, 2020a. 9

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8107–8116, 2020b. 22

Dongjun Kim, Seungjae Shin, Kyungwoo Song, Wanmo Kang, and Il-Chul Moon. Score matching model for unbounded data score. *ArXiv*, abs/2106.05527, 2021. 9

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2014. 2, 5, 14

Diederik P. Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *ArXiv*, abs/2107.00630, 2021. 9

Zhifeng Kong and Wei Ping. On fast sampling of diffusion probabilistic models. *ArXiv*, abs/2106.00132, 2021. 6, 8, 9

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. 7

Yann LeCun, Sumit Chopra, Raia Hadsell, Aurelio Ranzato, and Fu Jie Huang. A tutorial on energy-based learning. 2006. 2, 6

Chieh Hubert Lin, Chia-Che Chang, Yu-Sheng Chen, Da-Cheng Juan, Wei Wei, and Hwann-Tzong Chen. Coco-gan: Generation by parts via conditional coordinating. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4511–4520, 2019. 9

Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. *International Conference on Learning Representations*, 2022. 6, 9

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3730–3738, 2015. 7

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *ArXiv*, abs/2206.00927, 2022. 6

Eric Luhman and Troy Luhman. Knowledge distillation in iterative generative models for improved sampling speed. *ArXiv*, abs/2101.02388, 2021. 9

Zhaoyang Lyu, Xu Xudong, Ceyuan Yang, Dahua Lin, and Bo Dai. Accelerating diffusion models via early stop of the diffusion process. *ArXiv*, abs/2205.12524, 2022. 6, 9

Shakir Mohamed and Balaji Lakshminarayanan. Learning in implicit generative models. *arXiv preprint arXiv:1610.03483*, 2016. 1

Eliya Nachmani, Robin San-Roman, and Lior Wolf. Non gaussian denoising diffusion models. *ArXiv*, abs/2106.07582, 2021. 6

Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. *International conference on machine learning*, 2021. 9

Erik Nijkamp, Ruiqi Gao, Pavel Sountsov, Srinivas Vasudevan, Bo Pang, Song-Chun Zhu, and Ying Nian Wu. Learning energy-based model with flow-based backbone by neural transport mcmc. *arXiv preprint arXiv:2006.06897*, 2020a. 19

Erik Nijkamp, Mitch Hill, Tian Han, Song-Chun Zhu, and Ying Nian Wu. On the anatomy of mcmc-based maximum likelihood learning of energy-based models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020b. 6, 19

Kushagra Pandey, Avideep Mukherjee, Piyush Rai, and Abhishek Kumar. Diffusevae: Efficient, controllable and high-fidelity generation from low-dimensional latents. *ArXiv*, abs/2201.00308, 2022. 9

George Papamakarios, Eric T. Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22:57:1–57:64, 2021. 18

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical image computing and computer-assisted intervention*, 2015. 22

Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in Neural Information Processing Systems*, 2016. 7

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *International conference on machine learning*, 2015. 1, 2, 14

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *International Conference on Learning Representations*, 2021a. 1, 2, 3, 7, 8, 15, 22

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 2019. 6, 9

Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in Neural Information Processing Systems*, 2020. 6

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. In *Advances in Neural Information Processing Systems*, 2021b. 9

Yang Song, Jascha Narain Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *International Conference on Learning Representations*, 2021c. 6, 9, 22

Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. In *Advances in Neural Information Processing Systems*, 2021. 9

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23:1661–1674, 2011. 6

Daniel Watson, Jonathan Ho, Mohammad Norouzi, and William Chan. Learning to efficiently sample from diffusion probabilistic models. *ArXiv*, abs/2106.03802, 2021. 6

Daniel Watson, William Chan, Jonathan Ho, and Mohammad Norouzi. Learning fast samplers for diffusion models by differentiating through sample quality. *International Conference on Learning Representations*, 2022. 6, 8

Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. In *International conference on machine learning*, 2011. 19

Zhisheng Xiao, Karsten Kreis, Jan Kautz, and Arash Vahdat. Vaebm: A symbiosis between variational autoencoders and energy-based models. In *International Conference on Learning Representations*, 2021. 9

Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion gans. *International Conference on Learning Representations*, 2022. 5, 6, 9, 21, 22, 23, 25

Qinsheng Zhang, Molei Tao, and Yongxin Chen. gddim: Generalized denoising diffusion implicit models. *ArXiv*, abs/2206.05564, 2022. 9

Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training. *Advances in Neural Information Processing Systems*, 2020. 9

Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. Truncated diffusion probabilistic models. *ArXiv*, abs/2202.09671, 2022. 6, 9

A Review of Diffusion Probabilistic Models

In this section, we present the formulations of diffusion probabilistic model (DPM) and denoising diffusion implicit model (DDIM) for completeness.

A.1 Diffusion Probabilistic Models

DPM defines a Markov forward diffusion process represented by an inference process ($\mathbf{x}_{1:T}$ are latent variables):

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \qquad q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}), \tag{19}$$

that converting the data distribution $q(\mathbf{x}_0)$ into the terminal state $q(\mathbf{x}_T)$ closed to tractable prior $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$. The noise variance scale β_t at each time step is fixed and the resulting posteriors only conditioned on $\mathbf{x}_0$ are of Gaussian form

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_0, (1-\alpha_t)\mathbf{I}), \qquad \alpha_t = \prod_{i=1}^{t} 1-\beta_i, \tag{20}$$

allowing us to directly obtain $\mathbf{x}_t$ by ancestral sampling trick $\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1-\alpha_t}\epsilon, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

Although the forward diffusion process has analytically tractable inference distributions and is pretty convenient to obtain latent variables at each noise step, its reversal is typically hard to handle due to the unknown reverse kernel $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t) = \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1})q(\mathbf{x}_{t-1})}{q(\mathbf{x}_t)} = \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1})\int q(\mathbf{x}_{t-1}|\mathbf{x}_0)q(\mathbf{x}_0)\mathrm{d}\mathbf{x}_0}{\int q(\mathbf{x}_t|\mathbf{x}_0)q(\mathbf{x}_0)\mathrm{d}\mathbf{x}_0}. \tag{21}$$

Computing the exact $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ requires intractable integrals over the whole data space, and furthermore, we even know nothing about its form. In fact, $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is commonly multimodal, except the case that the noise scale β_t is small, on which the denoising distribution over $\mathbf{x}_{t-1}$ is approximately unimodal Gaussian near $\mathbf{x}_t$. Corresponding to the small noise scale, however, the diffusion chain is required to be long enough, which is exactly the assumption made in DPMs.

Under the Gaussian assumption, DPM adopts parametric Gaussian denosing kernel $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ to approximate $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$, leading to a Gaussian denoising process started from the tractable prior $p(\mathbf{x}_T)$:

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \qquad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \sigma_t^2\mathbf{I}). \tag{22}$$

The variance σ_t^2 is commonly fixed.

From the perspective of *variational inference* (Kingma & Welling, 2014), the forward diffusion process acts as the inference distribution (law) of the denoising process, so the processes in opposite directions actually form a process-based VAE. But unlike traditional VAE, DPM has unlearnable inference side. Therefore, the widely adopted evidence lower bound (ELBO) of negative log likelihood is used for training such models

$$\begin{aligned} &-\mathbb{E}_{q(\mathbf{x}_0)}[\log p_\theta(\mathbf{x}_0)] \le \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[\log\frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\right] \\ &= \mathbb{E}_q\left[-\log q(\mathbf{x}_0) + \sum_{t=1}^{T} D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) + D_{\mathrm{KL}}(q(\mathbf{x}_T)||p(\mathbf{x}_T))\right]. \end{aligned} \tag{23}$$

But the middle KL terms are still intractable. Fortunately, the variational bound can be rewritten into a more favorable form (see Ho et al. (2020); Sohl-Dickstein et al. (2015) for details)

$$= \mathbb{E}_q\left[-\log p_\theta(\mathbf{x}_0|\mathbf{x}_1) + \sum_{t=2}^{T} D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) + D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0)||p(\mathbf{x}_T))\right]. \tag{24}$$

Thanks to the observation that the reverse kernel conditioned on $\mathbf{x}_0$ has a special form

$$\begin{aligned} q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) &= \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1})q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)} = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t\mathbf{I}) \\ \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) &= \frac{\sqrt{1-\beta_t}(1-\alpha_{t-1})}{1-\alpha_t}\mathbf{x}_t + \frac{\sqrt{\alpha_{t-1}}\beta_t}{1-\alpha_t}\mathbf{x}_0, \qquad \tilde{\beta}_t = \frac{1-\alpha_{t-1}}{1-\alpha_t}\beta_t, \end{aligned} \tag{25}$$

DPM can matches up the parameterization of $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ with $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \frac{\sqrt{1-\beta_t}(1-\alpha_{t-1})}{1-\alpha_t}\mathbf{x}_t + \frac{\sqrt{\alpha_{t-1}}\beta_t}{1-\alpha_t}\boldsymbol{f}_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{1-\beta_t}}(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\alpha_t}}\epsilon_\theta(\mathbf{x}_t, t)) \tag{26}$$

where $\boldsymbol{f}_\theta(\mathbf{x}_t, t)$ predicts $\mathbf{x}_0$ in nature. The RHS is derived from $\mathbf{x}_0 = \frac{\mathbf{x}_t - \sqrt{1-\alpha_t}\epsilon}{\sqrt{\alpha_t}}$, which means that $\epsilon_\theta(\mathbf{x}_t, t)$ actually predicts the noise ϵ imposed to data sample $\mathbf{x}_0$ at step t.

A.2 Denoising Diffusion Implicit Models

To avoid the Gaussian assumption (small noise scales) made in DPMs, Song et al. (2021a) explore a class of non-Markovian forward process with Gaussian reverse kernel conditioned on $\mathbf{x}_0$ but whose marginal distributions still match up with the DPM forward process. Song et al. (2021a) present it as an inference distribution (1)

$$\begin{gathered} q_\sigma(\mathbf{x}_{0:T}) = q(\mathbf{x}_0)q_\sigma(\mathbf{x}_T|\mathbf{x}_0)\prod_{t=2}^{T} q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0), \quad q_\sigma(\mathbf{x}_T|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_T; \sqrt{\alpha_T}\mathbf{x}_0, (1-\alpha_T)\mathbf{I}), \\ q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \sqrt{\alpha_{t-1}}\mathbf{x}_0 + \sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \frac{\mathbf{x}_t - \sqrt{\alpha_t}\mathbf{x}_0}{\sqrt{1-\alpha_t}}, \sigma_t^2\mathbf{I}). \end{gathered}$$

Then a learnable generative model is denoted as Eq. (2)

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \qquad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{f}_\theta(\mathbf{x}_t, t))$$

where $\boldsymbol{f}_\theta(\mathbf{x}_t, t)$ are also used to predict $\mathbf{x}_0$. Song et al. (2021a) show that the optimal solution of the variational inference objective for training generative process is the same as that of DPM. So that the class of generative models are alternative sampling schemes for DPM. But this time, we do not need to assume that the noise scales are small, since the learning targets $q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ are already Gaussian.

Specifically, when $\sigma_t = \sqrt{\frac{1-\alpha_{t-1}}{1-\alpha_t}}\sqrt{1 - \frac{\alpha_t}{\alpha_{t-1}}}$ the generative process recovers to that in DDPM, when $\sigma_t = 0$ the generative process is termed DDIM. DDIM removes all the randomness in the generative process, except that in the generative starting point $\mathbf{x}_T$. Therefore, DDIM is an implicit generative model characterized by the deterministic path from $\mathbf{x}_T$ to $\mathbf{x}_0$. Correspondingly, the inference process of DDIM is a deterministic path from $\mathbf{x}_0$ to $\mathbf{x}_T$, whose randomness totally comes from $q(\mathbf{x}_T|\mathbf{x}_0)$:

$$\begin{gathered} q(\mathbf{x}_{1:T}|\mathbf{x}_0) = q(\mathbf{x}_T|\mathbf{x}_0)\prod_{t=2}^{T} q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0), \quad q(\mathbf{x}_T|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_T; \sqrt{\alpha_T}\mathbf{x}_0, (1-\alpha_T)\mathbf{I}), \\ q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \delta(\mathbf{x}_{t-1} - \left[\sqrt{\alpha_{t-1}}\mathbf{x}_0 + \sqrt{1-\alpha_{t-1}} \cdot \frac{\mathbf{x}_t - \sqrt{\alpha_t}\mathbf{x}_0}{\sqrt{1-\alpha_t}}\right]). \end{gathered} \tag{27}$$

B Details for Ensemble Denoising Implicit Models

B.1 Derivation for Ensemble Denoising Implicit Models

Proof of Proposition 1. Given the general $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ ($\sigma_t \geq 0$) presented as the sampling procedure of $q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0^t)$:

$$\mathbf{x}_{t-1} = \sqrt{\alpha_{t-1}}\mathbf{x}_0^t + \sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \frac{\mathbf{x}_t - \sqrt{\alpha_t}\mathbf{x}_0^t}{\sqrt{1-\alpha_t}} + \sigma_t \epsilon_t, \tag{28}$$

we provide a derivation for the general ensemble denoising implicit models $q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_0^{t:T}, \mathbf{x}_T)$ ($\sigma \geq 0$) by recursively expanding the particles $\mathbf{x}_k$ in Eq. (28) along $t \to T$:

$$\begin{aligned}
\mathbf{x}_{t-1} &= \sqrt{\alpha_{t-1}}\mathbf{x}_0^t + \sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \frac{\mathbf{x}_t - \sqrt{\alpha_t}\mathbf{x}_0^t}{\sqrt{1-\alpha_t}} + \sigma_t \epsilon_t \\
&= \left(\sqrt{\alpha_{t-1}} - \sqrt{1-\alpha_{t-1}-\sigma_t^2}\frac{\sqrt{\alpha_t}}{\sqrt{1-\alpha_t}}\right)\mathbf{x}_0^t + \frac{\sqrt{1-\alpha_{t-1}-\sigma_t^2}}{\sqrt{1-\alpha_t}}\mathbf{x}_t + \sigma_t\epsilon_t \\
&= \frac{\sqrt{\alpha_{t-1}}\sqrt{1-\alpha_t} - \sqrt{1-\alpha_{t-1}-\sigma_t^2}\sqrt{\alpha_t}}{\sqrt{1-\alpha_t}}\mathbf{x}_0^t \\
&+ \frac{\sqrt{1-\alpha_{t-1}-\sigma_t^2}}{\sqrt{1-\alpha_t}} \frac{\sqrt{\alpha_t}\sqrt{1-\alpha_{t+1}} - \sqrt{1-\alpha_t-\sigma_{t+1}^2}\sqrt{\alpha_{t+1}}}{\sqrt{1-\alpha_{t+1}}}\mathbf{x}_0^{t+1} \\
&+ \frac{\sqrt{1-\alpha_{t-1}-\sigma_t^2}}{\sqrt{1-\alpha_t}} \frac{\sqrt{1-\alpha_t-\sigma_{t+1}^2}}{\sqrt{1-\alpha_{t+1}}}\mathbf{x}_{t+1} + \frac{\sqrt{1-\alpha_{t-1}-\sigma_t^2}}{\sqrt{1-\alpha_t}}\sigma_{t+1}\epsilon_{t+1} + \sigma_t\epsilon_t \qquad (29) \\
&\cdots \\
&\cdots \\
&= \underbrace{\sum_{k=t}^{T}\left[\left(\prod_{m=t}^{k-1}\frac{\sqrt{1-\alpha_{m-1}-\sigma_m^2}}{\sqrt{1-\alpha_m}}\right)\frac{\sqrt{\alpha_{k-1}}\sqrt{1-\alpha_k} - \sqrt{1-\alpha_{k-1}-\sigma_k^2}\sqrt{\alpha_k}}{\sqrt{1-\alpha_k}}\mathbf{x}_0^k\right]}_{\textbf{linear combination of } \mathbf{x}_0^{t:T}} \\
&\quad + \underbrace{\prod_{m=t}^{T}\frac{\sqrt{1-\alpha_{m-1}-\sigma_m^2}}{\sqrt{1-\alpha_m}}\mathbf{x}_T}_{\textbf{dependence on } \mathbf{x}_T} + \underbrace{\sum_{k=t}^{T}\left[\left(\prod_{m=t}^{k-1}\frac{\sqrt{1-\alpha_{m-1}-\sigma_m^2}}{\sqrt{1-\alpha_m}}\right)\sigma_k\epsilon_k\right]}_{\textbf{combination of Gaussian noise } \epsilon_{t:T}}.
\end{aligned}$$

Note that $\epsilon_{t:T}$ are sampled from $\mathcal{N}(\mathbf{0}, \mathbf{I})$, we may replace the combination term of $\epsilon_{t:T}$ with

$$\epsilon^t \sim \mathcal{N}\left(\mathbf{0}, \sum_{k=t}^{T}\left[\left(\prod_{m=t}^{k-1}\frac{\sqrt{1-\alpha_{m-1}-\sigma_m^2}}{\sqrt{1-\alpha_m}}\right)^2\sigma_k^2\right]\mathbf{I}\right), \tag{30}$$

and as a result, $q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_0^{t:T}, \mathbf{x}_T)$ is still Gaussian.

In the case of DDIM ($\sigma = 0$) which this paper mainly concern, $q(\mathbf{x}_{t-1}|\mathbf{x}_0^{t:T}, \mathbf{x}_T)$:

$$\begin{aligned}
\mathbf{x}_{t-1} &= \sum_{k=t}^{T}\left[\left(\prod_{m=t}^{k-1}\frac{\sqrt{1-\alpha_{m-1}}}{\sqrt{1-\alpha_m}}\right)\frac{\sqrt{\alpha_{k-1}}\sqrt{1-\alpha_k} - \sqrt{1-\alpha_{k-1}}\sqrt{\alpha_k}}{\sqrt{1-\alpha_k}}\mathbf{x}_0^k\right] + \prod_{m=t}^{T}\frac{\sqrt{1-\alpha_{m-1}}}{\sqrt{1-\alpha_m}}\mathbf{x}_T \\
&= \underbrace{\sqrt{1-\alpha_{t-1}} \cdot \sum_{k=t}^{T}\left(\frac{\sqrt{\alpha_{k-1}}}{\sqrt{1-\alpha_{k-1}}} - \frac{\sqrt{\alpha_k}}{\sqrt{1-\alpha_k}}\right)\mathbf{x}_0^k}_{\textbf{linear combination of } \mathbf{x}_0^{t:T}} + \underbrace{\frac{\sqrt{1-\alpha_{t-1}}}{\sqrt{1-\alpha_T}}\mathbf{x}_T}_{\textbf{dependence on } \mathbf{x}_T} \\
&= \sqrt{1-\alpha_{t-1}} \cdot \sum_{k=t}^{T}(A_{k-1} - A_k)\,\mathbf{x}_0^k + \frac{\sqrt{1-\alpha_{t-1}}}{\sqrt{1-\alpha_T}}\mathbf{x}_T, \quad A_k = \frac{\sqrt{\alpha_k}}{\sqrt{1-\alpha_k}},
\end{aligned}$$

becomes a Dirac distribution. We can easily find the unrolled DDIM denoising kernel $q(\mathbf{x}_{t-1}|\mathbf{x}_0^{t:T}, \mathbf{x}_T)$ is a specific instance of the ensemble denoising implicit model (4) with $\omega_t^k = A_{k-1} - A_k$. □

B.2 Derivation for Improved DDIM

Given the unrolled DDIM denoising kernel (7):

$$\mathbf{y}_{t-1} = \sqrt{1-\alpha_{t-1}} \cdot \sum_{k=t}^{T} \left(A_{k-1} - A_k\right) \mathbf{x}_0^k + \frac{\sqrt{1-\alpha_{t-1}}}{\sqrt{1-\alpha_T}} \mathbf{x}_T, \quad A_k = \frac{\sqrt{\alpha_k}}{\sqrt{1-\alpha_k}}, \tag{31}$$

where $\mathbf{y}_{t-1}$ is introduced to distinguish from $\mathbf{x}_k$ used in $\mathbf{x}_0^k = \boldsymbol{f}_\theta(\mathbf{x}_k, k)$, we have

$$\sum_{k=t+1}^{T} \left(A_{k-1} - A_k\right) \mathbf{x}_0^k = \frac{\mathbf{y}_t}{\sqrt{1-\alpha_t}} - \frac{\mathbf{x}_T}{\sqrt{1-\alpha_T}}. \tag{32}$$

Then we introduce m_t to control the replacement proportion of $\mathbf{x}_0^{t+1:T} \to \mathbf{x}_0^t$, resulting in

$$\begin{aligned}
\mathbf{x}_{t-1} = {} & \sqrt{1-\alpha_{t-1}}(A_{t-1} - A_t)\mathbf{x}_0^t + \frac{\sqrt{1-\alpha_{t-1}}}{\sqrt{1-\alpha_T}} \mathbf{x}_T \\
& + \sqrt{1-\alpha_{t-1}} \sum_{k=t+1}^{T} \left(A_{k-1} - A_k\right) \left[m_t \mathbf{x}_0^t + (1-m_t)\mathbf{x}_0^k\right] \\
= {} & \sqrt{\alpha_{t-1}} \mathbf{x}_0^t - \sqrt{1-\alpha_{t-1}} A_t \mathbf{x}_0^t + \frac{\sqrt{1-\alpha_{t-1}}}{\sqrt{1-\alpha_T}} \mathbf{x}_T \\
& + \sqrt{1-\alpha_{t-1}} \sum_{k=t+1}^{T} \left(A_{k-1} - A_k\right) m_t \mathbf{x}_0^t + \sqrt{1-\alpha_{t-1}} \sum_{k=t+1}^{T} \left(A_{k-1} - A_k\right)(1-m_t)\mathbf{x}_0^k \\
= {} & \sqrt{\alpha_{t-1}} \mathbf{x}_0^t - \sqrt{1-\alpha_{t-1}} A_t \mathbf{x}_0^t + \frac{\sqrt{1-\alpha_{t-1}}}{\sqrt{1-\alpha_T}} \mathbf{x}_T \\
& + \sqrt{1-\alpha_{t-1}}(A_t - A_T) m_t \mathbf{x}_0^t + \underbrace{\sqrt{1-\alpha_{t-1}}(1-m_t)\left(\frac{\mathbf{y}_t}{\sqrt{1-\alpha_t}} - \frac{\mathbf{x}_T}{\sqrt{1-\alpha_T}}\right)}_{\textbf{by Eq. (32)}} \\
= {} & \sqrt{\alpha_{t-1}} \mathbf{x}_0^t - \sqrt{1-\alpha_{t-1}} A_t (1-m_t) \mathbf{x}_0^t + \sqrt{1-\alpha_{t-1}}(1-m_t)\frac{\mathbf{y}_t}{\sqrt{1-\alpha_t}} \\
& - \sqrt{1-\alpha_{t-1}} A_T m_t \mathbf{x}_0^t + \frac{\sqrt{1-\alpha_{t-1}}}{\sqrt{1-\alpha_T}} m_t \mathbf{x}_T \\
= {} & (1-m_t) \underbrace{\left[\sqrt{\alpha_{t-1}} \mathbf{x}_0^t + \sqrt{1-\alpha_{t-1}} \cdot \frac{\mathbf{y}_t - \sqrt{\alpha_t}\mathbf{x}_0^t}{\sqrt{1-\alpha_t}}\right]}_{\mathbf{y}_{t-1} \textbf{ computed by Eq. (6)}} \\
& \qquad + m_t \underbrace{\left[\sqrt{\alpha_{t-1}} \mathbf{x}_0^t + \sqrt{1-\alpha_{t-1}} \cdot \frac{\mathbf{x}_T - \sqrt{\alpha_T}\mathbf{x}_0^t}{\sqrt{1-\alpha_T}}\right]}_{\mathbf{z}_{t-1} \textbf{ computed by Eq. (10)}}.
\end{aligned} \tag{33}$$

Since we are replacing the former denoising predictions $\mathbf{x}_0^{t+1:T}$ with the latest denoising prediction $\mathbf{x}_0^t$, Eq. (33) is still a specific instance of the proposed ensemble denoising implicit models (4) with

$$\begin{aligned}
\omega_t^t &= A_{k-1} - A_k + m_t \cdot \sum_{k=t+1}^{T} \left(A_{k-1} - A_k\right) \\
\omega_t^k &= (1-m_t) \cdot (A_{k-1} - A_k), \qquad k = t+1, \ldots, T
\end{aligned} \tag{34}$$

for all t.

C DETAILS FOR GENERALIZED DENOISING IMPLICIT MODELS

C.1 LEARNING OBJECTIVES AND OPTIMIZING GRADIENTS

Preparations.

Lemma 1 (Change of variables, Papamakarios et al. (2021)). *Given a diffeomorphism T that transforms a real vector $\mathbf{z}$ sampled from the base distribution $p_{\boldsymbol{z}}(\mathbf{z})$ into $\mathbf{x} \sim p_{\boldsymbol{x}}(\mathbf{x})$, i.e., $\mathbf{x} = T(\mathbf{z})$, then we have:*

$$p_{\boldsymbol{z}}(\mathbf{z}) = p_{\boldsymbol{x}}(T(\mathbf{z})) \left| \det \frac{\partial T(\mathbf{z})}{\partial \mathbf{z}} \right|, \tag{35}$$

$$p_{\boldsymbol{x}}(\mathbf{x}) = p_{\boldsymbol{z}}(T^{-1}(\mathbf{x})) \left| \det \frac{\partial T^{-1}(\mathbf{x})}{\partial \mathbf{x}} \right|, \tag{36}$$

where $\frac{\partial T(\mathbf{z})}{\partial \mathbf{z}}$ and $\frac{\partial T^{-1}(\mathbf{x})}{\partial \mathbf{x}}$ denotes the Jacobian of T and T^{-1} respectively.

Note that Eq. (13) is exactly to transform the base distribution $p_\theta(\mathbf{x}_0^t|\mathbf{x}_t)$ to the target distribution $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_T)$ with $\mathbf{x}_{t-1} = T(\mathbf{x}_0^t; \mathbf{x}_T) = \sqrt{\alpha_{t-1}}\mathbf{x}_0^t + \sqrt{1-\alpha_{t-1}}\frac{\mathbf{x}_T - \sqrt{\alpha_T}\mathbf{x}_0^t}{\sqrt{1-\alpha_T}}$, since the transform kernel is a Dirac distribution $q(\mathbf{x}_{t-1}|\mathbf{x}_0^t, \mathbf{x}_T) = \delta(\mathbf{x}_{t-1} - T(\mathbf{x}_0^t; \mathbf{x}_T))$. Because $T(\mathbf{x}_0^t; \mathbf{x}_T)$ is invertible and differentiable with respect to $\mathbf{x}_0^t$, i.e., a *diffeomorphism*, we can apply Lemma 1 on them to obtain:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_T) = p_\theta(T^{-1}(\mathbf{x}_{t-1}; \mathbf{x}_T)|\mathbf{x}_t) \left| \det \frac{\partial T^{-1}(\mathbf{x}_{t-1}; \mathbf{x}_T)}{\partial \mathbf{x}_{t-1}} \right|. \tag{37}$$

Besides, we also provide a more inherent derivation for Eq. (37):

$$\begin{aligned} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_T) &= \int q(\mathbf{x}_{t-1}|\mathbf{x}_0^t, \mathbf{x}_T) p_\theta(\mathbf{x}_0^t|\mathbf{x}_t) \mathrm{d}\mathbf{x}_0^t \\ &= \int \delta(\mathbf{x}_{t-1} - T(\mathbf{x}_0^t; \mathbf{x}_T)) p_\theta(\mathbf{x}_0^t|\mathbf{x}_t) \mathrm{d}\mathbf{x}_0^t \\ &= \int \delta(\mathbf{x}_{t-1} - \hat{\mathbf{x}}_{t-1}) p_\theta(T^{-1}(\hat{\mathbf{x}}_{t-1}; \mathbf{x}_T)|\mathbf{x}_t) \left| \det \frac{\partial T^{-1}(\hat{\mathbf{x}}_{t-1}; \mathbf{x}_T)}{\partial \hat{\mathbf{x}}_{t-1}} \right| \mathrm{d}\hat{\mathbf{x}}_{t-1} \\ &= p_\theta(T^{-1}(\mathbf{x}_{t-1}; \mathbf{x}_T)|\mathbf{x}_t) \left| \det \frac{\partial T^{-1}(\mathbf{x}_{t-1}; \mathbf{x}_T)}{\partial \mathbf{x}_{t-1}} \right|. \end{aligned} \tag{38}$$

Then we present the forward (inference) process and the generative process of t-GDIM for completeness:

$$\text{Forward} \quad q(\mathbf{x}_{0:T}) = q(\mathbf{x}_0) q(\mathbf{x}_T|\mathbf{x}_0) \prod_{t=2}^{T} q(\mathbf{x}_{t-1}|\mathbf{x}_0, \mathbf{x}_T) \tag{39}$$

$$\text{Generative} \quad p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_T). \tag{40}$$

Derivation for Equation (15).

$$
\begin{aligned}
&-\mathbb{E}_{q(\mathbf{x}_0)}[\log p_\theta(\mathbf{x}_0)] = \int q(\mathbf{x}_{0:T}) \log \frac{p_\theta(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} \mathrm{d}\mathbf{x}_{0:T} \\
&= \int q(\mathbf{x}_{0:T}) \log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} \frac{p_\theta(\mathbf{x}_{1:T}|\mathbf{x}_0)}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \mathrm{d}\mathbf{x}_{0:T} \\
&= \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\right] - \underbrace{\mathbb{E}_{q(\mathbf{x}_0)}\left[D_{\mathrm{KL}}(q(\mathbf{x}_{1:T}|\mathbf{x}_0)||p_\theta(\mathbf{x}_{1:T}|\mathbf{x}_0))\right]}_{\geq 0} \\
&\leq \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\right] = \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_T)}\right] \\
&\doteq -\mathbb{E}_{q(\mathbf{x}_{0:T})}\left[\log \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_T)\right] = -\sum_{t=1}^{T} \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[\log p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_T)\right] \qquad (41) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \textbf{by Eq. (37)} \\
&= -\sum_{t=1}^{T} \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[\log \left[p_\theta(T^{-1}(\mathbf{x}_{t-1}; \mathbf{x}_T)|\mathbf{x}_t)\left|\det \frac{\partial T^{-1}(\mathbf{x}_{t-1}; \mathbf{x}_T)}{\partial \mathbf{x}_{t-1}}\right|\right]\right] \\
&\doteq -\sum_{t=1}^{T} \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[\log \frac{\exp(-E_\theta(T^{-1}(\mathbf{x}_{t-1}; \mathbf{x}_T), \mathbf{x}_t))}{Z(\theta, \mathbf{x}_t)}\right] \\
&= \sum_{t=1}^{T} \underbrace{\mathbb{E}_{q(\mathbf{x}_0, \mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{x}_T)}\left[E_\theta(T^{-1}(\mathbf{x}_{t-1}; \mathbf{x}_T), \mathbf{x}_t)) + \log Z(\theta, \mathbf{x}_t)\right]}_{\mathcal{J}(\theta, t)},
\end{aligned}
$$

where $\doteq$ denotes the equivalence with respect to optimizing θ.

Derivation for Equation (16).

$$
\begin{aligned}
\nabla_\theta \mathcal{J}(\theta, t) &= \mathbb{E}_{q(\mathbf{x}_0, \mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{x}_T)}\left[\nabla_\theta E_\theta(T^{-1}(\mathbf{x}_{t-1}; \mathbf{x}_T), \mathbf{x}_t)) + \nabla_\theta \log Z(\theta, \mathbf{x}_t)\right] \\
&= \mathbb{E}_{q(\mathbf{x}_0, \mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{x}_T)}\left[\nabla_\theta E_\theta(\mathbf{x}_0, \mathbf{x}_t) + \frac{\nabla_\theta \int \exp(-E_\theta(\mathbf{x}_0^t, \mathbf{x}_t))\mathrm{d}\mathbf{x}_0^t}{\int \exp(-E_\theta(\mathbf{x}_0^t, \mathbf{x}_t))\mathrm{d}\mathbf{x}_0^t}\right] \\
&= \mathbb{E}_{q(\mathbf{x}_0, \mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{x}_T)}\left[\nabla_\theta E_\theta(\mathbf{x}_0, \mathbf{x}_t) - \int \frac{\exp(-E_\theta(\mathbf{x}_0^t, \mathbf{x}_t))\nabla_\theta E_\theta(\mathbf{x}_0^t, \mathbf{x}_t)}{\int \exp(-E_\theta(\mathbf{x}_0^t, \mathbf{x}_t))\mathrm{d}\mathbf{x}_0^t} \mathrm{d}\mathbf{x}_0^t\right] \qquad (42) \\
&= \mathbb{E}_{q(\mathbf{x}_0, \mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{x}_T)}\left[\nabla_\theta E_\theta(\mathbf{x}_0, \mathbf{x}_t) - \mathbb{E}_{p_\theta(\mathbf{x}_0^t|\mathbf{x}_t)}\left[\nabla_\theta E_\theta(\mathbf{x}_0^t, \mathbf{x}_t)\right]\right] \\
&= \mathbb{E}_{q(\mathbf{x}_0)q(\mathbf{x}_t|\mathbf{x}_0)p_\theta(\mathbf{x}_0^t|\mathbf{x}_t)}\left[\nabla_\theta E_\theta(\mathbf{x}_0, \mathbf{x}_t) - \nabla_\theta E_\theta(\mathbf{x}_0^t, \mathbf{x}_t)\right].
\end{aligned}
$$

C.2 Approximate Sampling with IGMs

Dynamic-based sampling methods. In general, sampling from an unnormalized distribution, e.g. the proposed energy-based denoising distribution, is still an open problem. Recent works (Du & Mordatch, 2019; Nijkamp et al., 2020a) on studying generative EBMs typically adopt dynamic-based sampling methods for classic maximum likelihood training. Specifically, Langevin dynamic MCMC is accomplished by simulating K step Langevin stochastic dynamic which treats the Gibbs distribution induced by energy as the invariant distribution (Welling & Teh, 2011):

$$
\mathbf{y}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \qquad \mathbf{y}_{k+1} = \mathbf{y}_k - \frac{\eta}{2}\nabla_{\mathbf{y}_k} E_\theta(\mathbf{y}_k, \mathbf{x}_t) + \sqrt{\eta}\mathbf{w}_k, \qquad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{43}
$$

where η denotes the step size and the resulting $\mathbf{y}_K$ is regarded as the denoising prediction $\mathbf{x}_0^t$. However, running Langevin dynamic in data space is very slow and expensive, while also getting in trouble of mixing (Nijkamp et al., 2020b). So we introduce IGMs as the approximate sampler for faster inference.

Derivation for Equation (17). Recall $p_\phi(\mathbf{x}_0^t|\mathbf{x}_t)$ is represented by $\mathbf{x}_0^t = G_\phi(\mathbf{u}; \mathbf{x}_t, t), \mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, then we have (do not display the dependence on t):

$$
\begin{aligned}
&D_{\mathrm{KL}}(p_\phi(\mathbf{x}_0^t|\mathbf{x}_t)||p_\theta(\mathbf{x}_0^t|\mathbf{x}_t)) = \mathbb{E}_{p_\phi(\mathbf{x}_0^t|\mathbf{x}_t)}\left[\log\frac{p_\phi(\mathbf{x}_0^t|\mathbf{x}_t)}{p_\theta(\mathbf{x}_0^t|\mathbf{x}_t)}\right] \\
=\ & \mathbb{E}_{p_\phi(\mathbf{x}_0^t|\mathbf{x}_t)}\left[\log p_\phi(\mathbf{x}_0^t|\mathbf{x}_t)\right] - \mathbb{E}_{p_\phi(\mathbf{x}_0^t|\mathbf{x}_t)}\left[\log p_\theta(\mathbf{x}_0^t|\mathbf{x}_t)\right] \\
=\ & -\mathcal{H}(p_\phi(\mathbf{x}_0^t|\mathbf{x}_t)) - \mathbb{E}_{p_\phi(\mathbf{x}_0^t|\mathbf{x}_t)}\left[\log\frac{\exp(-E_\theta(\mathbf{x}_0^t, \mathbf{x}_t))}{Z(\theta, \mathbf{x}_t)}\right] \\
\doteq\ & -\mathcal{H}(p_\phi(\mathbf{x}_0^t|\mathbf{x}_t)) + \mathbb{E}_{p_\phi(\mathbf{x}_0^t|\mathbf{x}_t)}\left[E_\theta(\mathbf{x}_0^t, \mathbf{x}_t)\right] \\
& \qquad\qquad\qquad\qquad\qquad\qquad \textbf{by reparameterize trick} \\
\doteq\ & -\mathcal{H}(p_\phi(\mathbf{x}_0^t|\mathbf{x}_t)) + \mathbb{E}_{\mathcal{N}(\mathbf{u};\mathbf{0},\mathbf{I})}\left[E_\theta(G_\phi(\mathbf{u}; \mathbf{x}_t), \mathbf{x}_t)\right].
\end{aligned}
\tag{44}
$$

Reinterpretation of approximate sampler. In this part, we demonstrate that, training the sampler $p_\phi(\mathbf{x}_0^t|\mathbf{x}_t)$ to approximate the energy-based denoising distribution $p_\theta(\mathbf{x}_0^t|\mathbf{x}_t)$ with forward KL divergence is equivalent to the inner optimization for the variational formulation of $\mathcal{J}(\theta, t)$. The derivation is similar to *variational maximum likelihood* (Grathwohl et al., 2021).

First we express $\log Z(\theta, \mathbf{x}_t)$ with a variational dual formulation:

$$
\begin{aligned}
&\log Z(\theta, \mathbf{x}_t) = \max_\phi \left\{\log Z(\theta, \mathbf{x}_t) - D_{\mathrm{KL}}(p_\phi(\mathbf{x}_0^t|\mathbf{x}_t)||p_\theta(\mathbf{x}_0^t|\mathbf{x}_t))\right\} \\
=\ & \max_\phi \left\{\log Z(\theta, \mathbf{x}_t) + \int p_\phi(\mathbf{x}_0^t|\mathbf{x}_t)\log\frac{p_\theta(\mathbf{x}_0^t|\mathbf{x}_t)}{p_\phi(\mathbf{x}_0^t|\mathbf{x}_t)}\mathrm{d}\mathbf{x}_0^t\right\} \\
=\ & \max_\phi \left\{\log Z(\theta, \mathbf{x}_t) + \mathcal{H}(p_\phi(\mathbf{x}_0^t|\mathbf{x}_t)) + \int p_\phi(\mathbf{x}_0^t|\mathbf{x}_t)\log\frac{\exp(-E_\theta(\mathbf{x}_0^t, \mathbf{x}_t))}{Z(\theta, \mathbf{x}_t)}\mathrm{d}\mathbf{x}_0^t\right\} \\
=\ & \max_\phi \left\{\mathcal{H}(p_\phi(\mathbf{x}_0^t|\mathbf{x}_t)) - \mathbb{E}_{p_\phi(\mathbf{x}_0^t|\mathbf{x}_t)}\left[E_\theta(\mathbf{x}_0^t, \mathbf{x}_t)\right]\right\} \\
=\ & \max_\phi \left\{\mathcal{H}(p_\phi(\mathbf{x}_0^t|\mathbf{x}_t)) - \mathbb{E}_{\mathcal{N}(\mathbf{u};\mathbf{0},\mathbf{I})}\left[E_\theta(G_\phi(\mathbf{u}; \mathbf{x}_t), \mathbf{x}_t)\right]\right\}.
\end{aligned}
\tag{45}
$$

Then we plug it into the optimization for $\mathcal{J}(\theta, t)$:

$$
\begin{aligned}
&\min_\theta \left\{\mathcal{J}(\theta, t)\right\} = \min_\theta \left\{\mathbb{E}_{q(\mathbf{x}_0, \mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{x}_T)}\left[E_\theta(T^{-1}(\mathbf{x}_{t-1}; \mathbf{x}_T), \mathbf{x}_t) + \log Z(\theta, \mathbf{x}_t)\right]\right\} \\
=\ & \min_\theta \left\{\mathbb{E}_q\left[E_\theta(\mathbf{x}_0, \mathbf{x}_t) + \max_\phi \left\{\mathcal{H}(p_\phi(\mathbf{x}_0^t|\mathbf{x}_t)) - \mathbb{E}_{\mathcal{N}(\mathbf{u};\mathbf{0},\mathbf{I})}\left[E_\theta(G_\phi(\mathbf{u}; \mathbf{x}_t), \mathbf{x}_t)\right]\right\}\right]\right\} \\
=\ & \min_\theta \max_\phi \left\{\mathbb{E}_q\left[E_\theta(\mathbf{x}_0, \mathbf{x}_t) + \mathcal{H}(p_\phi(\mathbf{x}_0^t|\mathbf{x}_t)) - \mathbb{E}_{\mathcal{N}(\mathbf{u};\mathbf{0},\mathbf{I})}\left[E_\theta(G_\phi(\mathbf{u}; \mathbf{x}_t), \mathbf{x}_t)\right]\right]\right\}.
\end{aligned}
\tag{46}
$$

In other word, the minimization of original $\mathcal{J}(\theta, t)$ is transformed to a nested optimization problem. This nested optimization is actually a bi-level optimization problem, however, alternating optimization scheme that commonly adopted in GANs or Actor-Critic is used in this work.

D More Detailed Relations

Connection to recovery EBM. Inspired by diffusion models, Gao et al. (2021) propose *diffusion recovery likelihood* to learn a sequence of conditional EBMs:

$$p_\theta(\mathbf{y}_{t-1}|\mathbf{x}_t) = \frac{1}{\tilde{Z}(\theta, \mathbf{x}_t)} \exp\left(-E_\theta(\mathbf{y}_{t-1}) - \frac{\|\mathbf{x}_t - \mathbf{y}_{t-1}\|^2}{2\beta_t^2}\right), \tag{47}$$

where $\mathbf{y}_{t-1} = \sqrt{1-\beta_{t-1}^2}\mathbf{x}_{t-1}$, and we do not display the dependence on t for brevity. The corresponding joint energy is denoted by $E_\theta(\mathbf{y}_{t-1}) + \frac{\|\mathbf{x}_t - \mathbf{y}_{t-1}\|^2}{2\beta_t^2}$. It implies that the distribution represented by the marginal EBM $p_\theta(\mathbf{y}_{t-1}) \propto \exp(-E_\theta(\mathbf{y}_{t-1}))$ is constrained to close to the condition $\mathbf{x}_t$ in the Euclidean distance, which stands for *recovery*.

In the case of t-GDIM, we parameterize the denoising prediction with conditional EBM (14):

$$p_\theta(\mathbf{x}_0^t|\mathbf{x}_t) = \frac{\exp(-E_\theta(\mathbf{x}_0^t, \mathbf{x}_t))}{Z(\theta, \mathbf{x}_t)},$$

which can represent the conditional distribution $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_T)$ via deterministic transform (37).

Although both recovery EBM and t-GDIM are using conditional EBMs to learn $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ or $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_T)$, our parameterization, modelling the joint energy rather than the marginal energy, is more flexible than that used in recovery EBMs if the diffusion scale is large (corresponding to few diffusion steps). This is because, the quadratic Euclidean distance is not a good measure if $\mathbf{y}_{t-1}$ is not close to $\mathbf{x}_t$ in high-dimensional space. But recovery EBM provides us an insightful way to bring t-GDIM explicit dependence on $\mathbf{x}_T$:

$$p_\theta(\mathbf{x}_0^t|\mathbf{x}_t, \mathbf{x}_T) \propto \exp\left(-E_\theta(\mathbf{x}_0^t, \mathbf{x}_t) - \gamma\|\mathbf{x}_0^t - \mathbf{x}_0^T\|^2\right), \tag{48}$$

where γ denotes the strength of Euclidean distance constraints. The additional Euclidean distance term constrains the joint energy to be close to the denoising target at step T, $\mathbf{x}_0^T \sim p_\theta(\mathbf{x}_0^T|\mathbf{x}_T)$. It forces the current denoising prediction $\mathbf{x}_0^t$ to lie in the same mode as that of $\mathbf{x}_0^T$, thus the generative process is able to fit the deterministic inference path (9). However, this conditional EBM requires dynamic-based MCMC for training and sampling, just like that in Gao et al. (2021). We leave this for future work.

Connections to DDGAN. Xiao et al. (2022) propose a GAN-based diffusion model:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \int p_\theta(\mathbf{x}_0|\mathbf{x}_t) q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \mathrm{d}\mathbf{x}_0 = \int \mathcal{N}(\mathbf{u}; \mathbf{0}, \mathbf{I}) q(\mathbf{x}_{t-1}|\mathbf{x}_t, G_\theta(\mathbf{u}; \mathbf{x}_t)) \mathrm{d}\mathbf{u}, \tag{49}$$

named denoising diffusion GAN (DDGAN). DDGAN shares a similar spirit to our t-GDIM. However, the GDIM is based on denoising implicit model which has deterministic $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$, while DDGAN is based on DPM. Since the DPM is a special instance of Eq. (2), and by Eqs. (29) and (30) we know the DPM generative process is also an ensemble of denoising predictions, so DDGAN is an ensemble denoising *diffusion* model. Our GDIM is a more general framework that replace the ensemble denoising prediction with probabilistic inference. And the t-GDIM specifics the probabilistic ensemble denoising prediction to a radical one that only conditioned on $\mathbf{x}_t$.

Notice DDGAN use a conditional discriminator $D_\psi(\mathbf{x}_{t-1}|\mathbf{x}_t)$ to guide the training of generator, we may derive a variant of t-GDIM similar to DDGAN (termed T-GDIM):

$$\min_\theta \max_\phi \left\{ \mathbb{E}_{q(\mathbf{x}_0, \mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{x}_T)\mathcal{N}(\mathbf{u};\mathbf{0},\mathbf{I})} \left[E_\theta(\mathbf{x}_{t-1}, \mathbf{x}_t) - E_\theta\left(T(G_\phi(\mathbf{u}; \mathbf{x}_t); \mathbf{x}_T), \mathbf{x}_t\right) \right] \right\}, \tag{50}$$

where $E_\theta(\mathbf{x}_{t-1}, \mathbf{x}_t)$ denotes the joint energy over $\mathbf{x}_{t-1}$ and $\mathbf{x}_t$. Now the new conditional EBM is used to directly represent the denoising kernel:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \frac{\exp(-E_\theta(\mathbf{x}_{t-1}, \mathbf{x}_t))}{\tilde{Z}(\theta, \mathbf{x}_t)}. \tag{51}$$

T-GDIM can be regarded as a denoising implicit model counterpart of Xiao et al. (2022). Since the complete denoising kernel ought to be $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_T)$, T-GDIM is still missing the dependence on $\mathbf{x}_T$. However, the optimizing objective (50) indicates that the dependence on $\mathbf{x}_T$ is incorporated implicitly in the training procedure. Besides, the T-GDIM denoising kernel $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_T)$ is typically more unimodal than $p_\theta(\mathbf{x}_0^t|\mathbf{x}_t)$ in t-GDIM, so is easier to be trained. We provide some additional studies on T-GDIM in Appendix E.

E Experimental Details

E.1 Experimental Settings

Model architecture. Since iDDIMs are modifications on DDIM, and thus are alternative sampling methods for DDPM, we use our self-trained DDPM score predictor ϵ_θ for computing the denoising prediction $\mathbf{x}_0^t = \frac{\mathbf{x}_t - \sqrt{1-\alpha_t}\epsilon_\theta(\mathbf{x}_t,t)}{\sqrt{\alpha_t}}$ in iDDIM experiments. The architecture of ϵ_θ follows the U-Net (Ronneberger et al., 2015) designed in Ho et al. (2020). Please see Ho et al. (2020) for details.

Since the entropy ignored variational maximum likelihood framework used for the conditional EBM and variational sampler in t-GDIM and T-GDIM is much similar to GAN-based optimization, we largely follow the architecture design in Xiao et al. (2022) which use GAN to accomplish DDPM. Our conditional EBM is adapted from the discriminator used in Karras et al. (2020b), and the variational sampler is adapted from the NCSN++ proposed in Song et al. (2021c). For CIFAR10 and CelebAHQ-256, we partially borrow the structure and hyper-parameters from Xiao et al. (2022), while for CelebA-64 and CelebA-128 that they do not consider, we provide a similar design scheme in Tab. 6 and Tab. 7.

	CIFAR10	CelebA-64	CelebA-128	CelebAHQ-256
# of ResNet blocks per scale	2	2	2	2
Initial # of channels	128	64	64	64
Channel multiplier for each scale	(1,2,2,2)	(1,2,3,4)	(1,1,2,3,4)	(1,1,2,2,4,4)
Scale of attention block	16	16	16	16
Latent dimension	100	100	100	100
# of latent mapping layers	4	3	3	3
Latent embedding dimension	256	256	256	256

Table 6: Hyper-parameters for the variational sampler.

CIFAR10
1×1 conv2d, 128
ResBlock, 128
ResBlock down, 256
ResBlock down, 512
ResBlock down, 512
minibatch std layer
Global Sum Pooling
FC layer → scalar

CelebA-64
1×1 conv2d, 128
ResBlock, 128
ResBlock down, 256
ResBlock down, 512
ResBlock down, 512
ResBlock down, 512
minibatch std layer
Global Sum Pooling
FC layer → scalar

CelebA-128
1×1 conv2d, 128
ResBlock down, 128
ResBlock down, 256
ResBlock down, 512
ResBlock down, 512
ResBlock down, 512
minibatch std layer
Global Sum Pooling
FC layer → scalar

CelebAHQ-256
1×1 conv2d, 128
ResBlock down, 128
ResBlock down, 256
ResBlock down, 512
ResBlock down, 512
ResBlock down, 512
ResBlock down, 512
minibatch std layer
Global Sum Pooling
FC layer → scalar

Table 7: Structure for the conditional EBM.

Training. As described in the main paper, we adopt the 1000 step linear noise schedule from $\beta_1 = 10^{-4}$ to $\beta_{1000} = 0.02$ to build the complete diffusion process as in Ho et al. (2020). To build the generative sub-sequence, we follow the quadratic timesteps selection in Song et al. (2021a), i.e., $\tau_i = \text{Ceil}(ci^2)$ such that $cT^2 = 1000$. Different from Song et al. (2021a), the quadratic selection is used for all datasets in this work.

For iDDIM, we use the self-trained DDPM score predictor. The score predictor is trained with the same objective in DDIM or DDPM. Our pre-trained DDPM model gets 3.19 FID and is slightly worse than 3.17 reported in Ho et al. (2020). For the t-GDIM, we use the following GAN-based optimization:

$$\max_{\theta} \mathbb{E}_{U(t)q(\mathbf{x}_0)q(\mathbf{x}_t|\mathbf{x}_0)\mathcal{N}(\mathbf{u};\mathbf{0},\mathbf{I})}\Big[\log\big(-E_\theta(\mathbf{x}_0,\mathbf{x}_t)\big) + \log\big(1 + E_\theta(G_\phi(\mathbf{u};\mathbf{x}_t),\mathbf{x}_t)\big)\Big] \tag{52}$$

$$\max_{\phi} \mathbb{E}_{U(t)q(\mathbf{x}_0)q(\mathbf{x}_t|\mathbf{x}_0)\mathcal{N}(\mathbf{u};\mathbf{0},\mathbf{I})}\Big[\log\big(-E_\theta(G_\phi(\mathbf{u};\mathbf{x}_t),\mathbf{x}_t)\big)\Big], \tag{53}$$

where $t \sim U(t)$ denotes randomly selecting a t from $1, \dots, T$. For the T-GDIM (50), we use:

$$\max_\theta \mathbb{E}_{U(t)q(\mathbf{x}_0)q(\mathbf{x}_{t-1},\mathbf{x}_t,\mathbf{x}_T|\mathbf{x}_0)\mathcal{N}(\mathbf{u};\mathbf{0},\mathbf{I})}\Big[\log\big(-E_\theta(\mathbf{x}_{t-1},\mathbf{x}_t)\big) \tag{54}$$

$$+\log\big(1+E_\theta(T(G_\phi(\mathbf{u};\mathbf{x}_t);\mathbf{x}_T),\mathbf{x}_t)\big)\Big] \tag{55}$$

$$\max_\phi \mathbb{E}_{U(t)q(\mathbf{x}_0)q(\mathbf{x}_{t-1},\mathbf{x}_t,\mathbf{x}_T|\mathbf{x}_0)\mathcal{N}(\mathbf{u};\mathbf{0},\mathbf{I})}\Big[\log\big(-E_\theta(T(G_\phi(\mathbf{u};\mathbf{x}_t);\mathbf{x}_T),\mathbf{x}_t)\big)\Big], \tag{56}$$

where $q(\mathbf{x}_{t-1},\mathbf{x}_t,\mathbf{x}_T|\mathbf{x}_0) = q(\mathbf{x}_T|\mathbf{x}_0)q(\mathbf{x}_{t-1},\mathbf{x}_t|\mathbf{x}_0,\mathbf{x}_T)$ denotes the deterministic path between $\mathbf{x}_0$ and $\mathbf{x}_T \sim q(\mathbf{x}_T|\mathbf{x}_0)$, computed by $\mathbf{x}_{t-1} = T(\mathbf{x}_0;\mathbf{x}_T,t-1)$. Correspondingly, the fake $\hat{\mathbf{x}}_{t-1}$ is computed by $\hat{\mathbf{x}}_{t-1} = T(G_\phi(\mathbf{u};\mathbf{x}_t);\mathbf{x}_T,t-1)$.

The optimization hyper-parameters is partially following that in Xiao et al. (2022), we summarize them in Tab. 8. We trained our models on CIFAR10 and CelebA-64 using 4 RTX2080Ti GPUs. On CelebA-128 and CelebAHQ-256, we use 4 A100 GPUs.

	CIFAR10	CelebA-64	CelebA-128	CelebAHQ-256
Initial learning rate for sampler	1.6×10^{-4}	1.6×10^{-4}	1.8×10^{-4}	2.0×10^{-4}
Initial learning rate for EBM	1.25×10^{-4}	10^{-4}	10^{-4}	10^{-4}
Adam optimizer β_1	0.5	0.5	0.5	0.5
Adam optimizer β_2	0.9	0.9	0.9	0.9
EMA	0.9999	0.9999	0.9999	0.9999
Batch size	256	128	128	32
# of training iterations	400k	400k	500k	700k
# of GPUs	4	4	4	4

Table 8: Optimization hyper-parameters.

E.2 Additional Studies

Using iDDIM to improve GDIMs. In Tab. 9, we using iDDIM sampler on 4-step t-GDIM and T-GDIM on CIFAR10 image generation measured by FID. Thanks to the implicit dependence on $\mathbf{x}_T$, T-GDIM performs slightly better than t-GDIM. Besides, iDDIM sampler can marginally improve the generation performance on t-GDIM, but weaken that of T-GDIM. In Tab. 10, we show the results on CelebA-64 and CelebA-128, where only the influence of iDDIM on t-GDIM is presented. Figure 7 shows the CelebA-128 generation samples of t-GDIM+iDDIM with varying m. We find the performance with different m is hard to tell with human eyes, it confirms that the denoising targets $\mathbf{x}_0^t \sim p_\theta(\mathbf{x}_0^t|\mathbf{x}_t)$ in t-GDIM are almost the same, which is compatible with the target deterministic inference process. So that in the case of 4 sampling steps, the t-GDIM is able to fit the deterministic inference process even though without the dependence on $\mathbf{x}_T$.

m	1.0	0.9	0.8	0.7	0.6	0.0
t-GDIM	5.51	6.36	5.34	**5.24**	5.31	5.57
T-GDIM	**4.90**	5.04	5.15	5.20	-	6.15

Table 9: CIFAR10 image generation of t-GDIM+iDDIM and T-GDIM+iDDIM, measured by FID↓.

m	1.0	0.9	0.8	0.7	0.6	0.5
CelebA-64	3.28	3.14	3.14	3.04	**2.93**	2.95
CelebA-128	4.45	4.29	4.17	4.11	**4.04**	4.07

Table 10: CelebA-64 and CelebA-128 image generation of t-GDIM+iDDIM, measured by FID↓.

Mode coverage of t-GDIM sampler. As we discussed in the main paper, the GAN-based optimization for t-GDIM is similar to the entropy ignored variational maximum likelihood in theory.

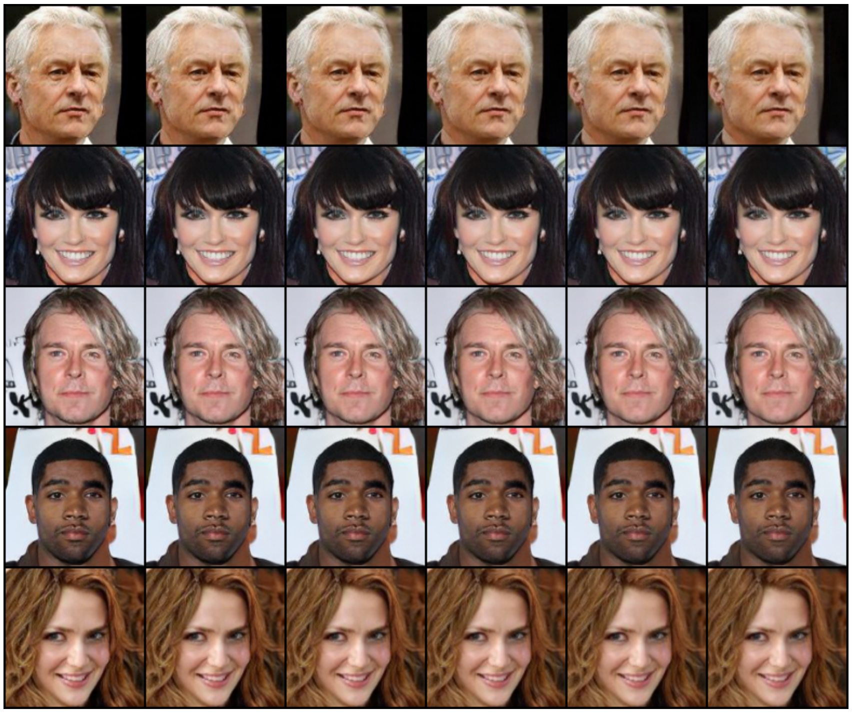

Figure 7: CelebA-128 samples of t-GDIM+iDDIM with varying m. From left to right: $m = 1.0, 0.8, 0.6, 0.4, 0.2, 0.0$.

Since the entropy term compels the variational sampler to be smoother, the absence of entropy leads to poor mode coverage of sampler. In Fig. 8, we show groups of $(\mathbf{x}_0, \mathbf{x}_0^t, \mathbf{x}_t)$ in the training procedure of t-GDIM. We find given the largest noise scale $\mathbf{x}_T$, the sampler tend to give predictions $\mathbf{x}_0^T$ with poor mode coverage. In fact, since $\mathbf{x}_T$ are almost pure noises, the case at step T is similar to the pure GANs that directly mapping the reference distribution to the data distribution (as we discussed in Sec. 1). Pure GANs have suffered from the poor mode coverage a lot, while data augmentation or other methods to compel the samplers to cover more modes is critical for GAN training. For example, DDGANs use DDPM framework and thus introduce additional noises as data augmentation, leading to better generation performance than us. Nevertheless, our t-GDIM can still generate samples with high diversity, that is because in the multi-step generative process, intermediate structural assumptions are introduced, and the $\mathbf{u}_t \sim \mathcal{N}(\mathbf{u}_t; \mathbf{0}, \mathbf{I})$ in sampler bring about more randomness to mitigate the influence of poor mode coverage at step T. Therefore, we may leverage proven optimization techniques in GANs to further improve our t-GDIM, but we only report the pure version of our model in this work, as it already achieves comparable generation quality to other generative models.

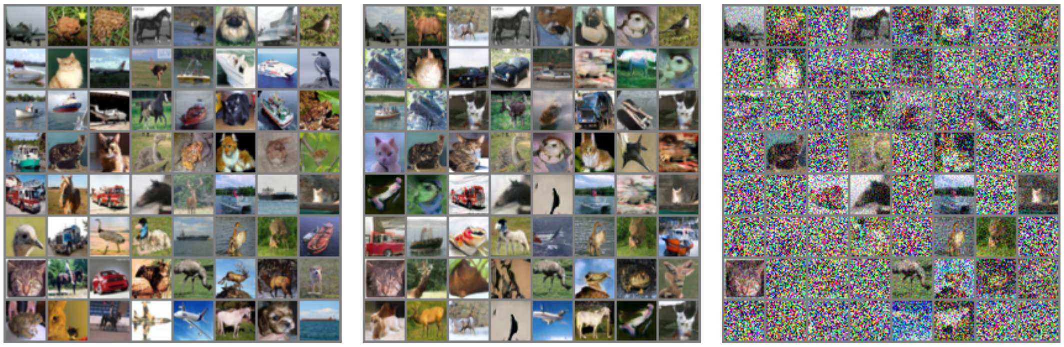

Figure 8: Denoising predictions of t-GDIM. Left: $\mathbf{x}_0$. Middle: $\mathbf{x}_0^t$. Right: $\mathbf{x}_t$.

F Connection and Comparison with DDGAN

In this section, we provide a deeper comparison with DDGAN (Xiao et al., 2022).

F.1 Motivation

DDGAN follows the basic framework of DDPM that constructs Markovian generative process to approximate the fixed reverse process:

$$q(\mathbf{x}_{0:T}) = q(\mathbf{x}_T)\prod_{t=1}^{T} q(\mathbf{x}_{t-1}|\mathbf{x}_t), \tag{57}$$

where $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is multi-modal if T is small. The central idea of DDGAN is predicting $\mathbf{x}_0$ by the generator with latent variable $\mathbf{u}$, $\mathbf{x}_0^t = G_\theta(\mathbf{u};\mathbf{x}_t,t)$, and then approximates Eq. (57) with the following generative process:

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \quad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \int p_\theta(\mathbf{x}_0^t|\mathbf{x}_t)q(\mathbf{x}_{t-1}|\mathbf{x}_0^t,\mathbf{x}_t)\mathrm{d}\mathbf{x}_0^t, \tag{58}$$

where $p_\theta(\mathbf{x}_0^t|\mathbf{x}_t)$ is the implicit distribution imposed by the generator. Thanks to the flexibility of $p_\theta(\mathbf{x}_0^t|\mathbf{x}_t)$, $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is able to approximate the multi-modal $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$.

Since the idea that predicting $\mathbf{x}_0$ with stronger stochastic mapping (instead of deterministic mapping $\mathbf{x}_0^t = \boldsymbol{f}_\theta(\mathbf{x}_t,t)$) is quit simple and natural, and the well-trained generator can provide good denoising predictions as expected. We use the same generator $G_\theta(\mathbf{z};\mathbf{x}_t,t)$ to accomplish the probabilistic inference $p_\theta(\mathbf{x}_0^t|\mathbf{x}_t)$ in our t-GDIM. However there are some basic differences and we will explain them along the theoretical route of t-GDIM.

Most obviously, DDGAN is based on DDPM, while our GDIM is based on DDIM. The frameworks of DDPM and DDIM are presented in Appendix A. In addition to the difference about the stochastic path (based on DDPM) versus the deterministic path (based on DDIM), the most important difference is the learning target (or target inference process). Different from Eq. (57) in DDGAN, the learning target of GDIM is exactly the same one of the ensemble models, (9) or (39):

$$q(\mathbf{x}_{0:T}) = q(\mathbf{x}_0)q(\mathbf{x}_T|\mathbf{x}_0)\prod_{t=2}^{T} q(\mathbf{x}_{t-1}|\mathbf{x}_0,\mathbf{x}_T). \tag{59}$$

The special explicit dependence on $\mathbf{x}_T$ and $\mathbf{x}_0$ implies that, at each denoising step, we should first predict $\mathbf{x}_0$ and then combine it with $\mathbf{x}_T$ to obtain the next state $\mathbf{x}_{t-1}$. For the ensemble denoising implicit models (4) (include DDIM (6), radical ensemble model (10) and iDDIM (11)), we use the convex ensemble of denoising predictions $\mathbf{x}_0^{t:T}$ to predict $\mathbf{x}_0$. For the general GDIMs (12):

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_{t:T}),\ p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_{t:T}) = \int p_\theta(\bar{\mathbf{x}}_0^t|\mathbf{x}_{t:T})q(\mathbf{x}_{t-1}|\bar{\mathbf{x}}_0^t,\mathbf{x}_T)\mathrm{d}\bar{\mathbf{x}}_0^t, \tag{60}$$

we use probabilistic inference $p_\theta(\bar{\mathbf{x}}_0^t|\mathbf{x}_{t:T})$ to sample a prediction $\bar{\mathbf{x}}_0^t$. By comparing the general GDIM generative process (60) with that of DDGAN (58), one can find two differences: **1)** the probabilistic inference in Eq. (60) depends on additional states $\mathbf{x}_{t+1:T}$; **2)** the denoising kernel in Eq. (60), $q(\mathbf{x}_{t-1}|\bar{\mathbf{x}}_0^t,\mathbf{x}_T)$, is a deterministic transform depending on $\mathbf{x}_T$, while in Eq. (58), $q(\mathbf{x}_{t-1}|\mathbf{x}_0^t,\mathbf{x}_t)$ is stochastic and depends on $\mathbf{x}_t$.

In **1)**, the additional dependence allow the general GDIM to **consider more information in previous states** $\mathbf{x}_{t+1:T}$ to perform inference of $\bar{\mathbf{x}}_0^t$. To explain the difference **2)**, we first consider the stochastic version of ensemble denoising models discussed in Eqs. (29) and (30). Since $q(\mathbf{x}_{t-1}|\mathbf{x}_0^t,\mathbf{x}_t)$ (also used in DDPM) is a special instance of Eq. (28), the generative process in DDGAN is actually an ensemble denoising diffusion model $q(\mathbf{x}_{t-1}|\mathbf{x}_0^{t:T},\mathbf{x}_T)$ (see Eq. (29)), which consists of the ensemble of obtained denoising predictions $\mathbf{x}_0^{t:T}$, the $\mathbf{x}_T$ term and the ensemble of noises $\epsilon_{t:T}$. Then we back to the GDIM, the deterministic denoising kernel $q(\mathbf{x}_{t-1}|\bar{\mathbf{x}}_0^t,\mathbf{x}_T)$ **gets rid of the dependence on all the noises** $\epsilon_{t:T}$, and moreover, **gets rid of the dependence on the previously obtained denoising predictions** $\mathbf{x}_0^{t+1:T}$. Now the general GDIM only trusts the most recent denoising prediction $\bar{\mathbf{x}}_0^t$ in theory. And the $\bar{\mathbf{x}}_0^t$, sampled from the probabilistic model in **1)**, seems like a probabilistic extension to the ensemble of denoising predictions (see Fig. 2). So we think GDIM as a more general framework than that in DDGAN.

However, it is difficult to implement the probabilistic model $p_\theta(\bar{\mathbf{x}}_0^t|\mathbf{x}_{t:T})$ with recent strong but efficient generative models, because of the explicit dependence on more than one state. To derive a tractable implementation, we introduce t-GDIM, the simplest GDIM whose probabilistic inference only depends on the most recent state $\mathbf{x}_t$:

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_T),\ p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_T) = \int p_\theta(\mathbf{x}_0^t|\mathbf{x}_t) q(\mathbf{x}_{t-1}|\mathbf{x}_0^t, \mathbf{x}_T) \mathrm{d}\mathbf{x}_0^t. \tag{61}$$

This eliminates the difference **1)** but will not damage the performance as long as $p_\theta(\mathbf{x}_0^t|\mathbf{x}_t)$ can provide good denoising predictions. Indeed, experiments in Sec. 6.2 have shown t-GDIM can get good denoising predictions as expected. On the other hand, **2)** still remains, which implies the basic difference between t-GDIM and DDGAN in theory.

F.2 Implementation and Limitations

The discussion above shows both DDGAN and t-GDIM use generator $G_\theta(\mathbf{u}; \mathbf{x}_t, t)$ to obtain $\mathbf{x}_0^t$, but they leverage $\mathbf{x}_0^t$ in distinct ways to construct the generative denoising process. In this section, we show the differences about training methods and limitations.

First we discuss the training method from the perspective of GAN-based optimization. Note in DDGAN, the Markovian denoising kernels $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ (58) are trained to approximate $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ (57). So DDGAN advocates to leverage a discriminator $D(\mathbf{x}_{t-1}, \mathbf{x}_t, t)$ to guide the training of $p_\theta(\mathbf{x}_0^t|\mathbf{x}_t)$. In practice, they use generator $G_\theta(\mathbf{u}; \mathbf{x}_t, t)$ to get a denoising prediction $\mathbf{x}_0^t$ and then use Gaussian kernel $q(\mathbf{x}_{t-1}|\mathbf{x}_0^t, \mathbf{x}_t)$ to sample a fake $\hat{\mathbf{x}}_{t-1}$. On the other hand, the real $\mathbf{x}_{t-1}$ is sampled from the forward process $q(\mathbf{x}_0)q(\mathbf{x}_{t-1}|\mathbf{x}_0)q(\mathbf{x}_t|\mathbf{x}_{t-1})$. Then the discriminator $D(\mathbf{x}_{t-1}, \mathbf{x}_t, t)$ is trained to distinguish the real $\mathbf{x}_{t-1}$ and the fake $\hat{\mathbf{x}}_{t-1}$ depending on $\mathbf{x}_t$.

But in our t-GDIM, the GAN-based optimization leverages a distinct discriminator that is trained to distinguish the real $\mathbf{x}_0$ and the denoising prediction $\mathbf{x}_0^t$ depending on $\mathbf{x}_t$. We denote it as the joint energy $E(\mathbf{x}_0, \mathbf{x}_t, t)$ in the main paper. Of course, the tuple used in t-GDIM training is sampled by $(\mathbf{x}_0, \mathbf{x}_t, \mathbf{x}_0^t) \sim q(\mathbf{x}_0)q(\mathbf{x}_t|\mathbf{x}_0)p_\theta(\mathbf{x}_0^t|\mathbf{x}_t)$, and is also different from that in DDGAN. In fact, the training differences comes from the distinct learning targets. Given the learning target Eq. (59), our GAN-based optimization is derived totally from the perspective of *variational maximum likelihood* (see Sec. 4.2 and related appendix). In addition to t-GDIM, we introduce T-GDIM in Appendix D whose discriminator is similar to that in DDGAN. However, unfortunately, both T-GDIM and DDGAN can not be easily interpreted from the perspective of *variational maximum likelihood*. So in principle we only consider the t-GDIM in our main paper.

As shown in Sec. 4.2 and Appendix E.2, the GAN-based optimization used in t-GDIM is an entropy-ignored variational maximum likelihood. So t-GDIM behaves like a pure unconditional GAN at step T since $\mathbf{x}_T$ are almost pure noises, and thus the sampler performs poor mode coverage (see the middle image of Fig. 8, there are very similar denoising predictions though denoised from different $\mathbf{x}_T$). We believe the mode coverage will damage the performance of t-GDIM, but we still report the pure version of t-GDIM as it is good enough. Generally speaking, DDGAN also does not involve the entropy of sampler. But notice the discriminator of DDGAN $D(\mathbf{x}_{t-1}, \mathbf{x}_t, t)$ is defined over the space of $\mathbf{x}_{t-1}$. Since $\mathbf{x}_{t-1}$ is a noisy version of $\mathbf{x}_0$ which is closer to $\mathbf{x}_t$, the $D(\mathbf{x}_{t-1}, \mathbf{x}_t, t)$ is smoother than the discriminator of t-GDIM $E(\mathbf{x}_0, \mathbf{x}_t, t)$. In GAN training, adding noises to the samples $\mathbf{x}_0$ is an efficient data augmentation used to compel the sampler to be smoother. As a result, DDGAN performs good mode coverage and thus gets better FID than our t-GDIM on CIFAR10 experiments (see Tab. 3). However, our t-GDIM achieves better performance than DDGAN on CelebAHQ-256 (see Tab. 5).