ShiftDDPMs: Exploring Conditional Diffusion Models by Shifting Diffusion Trajectories

Zijian Zhang[1], **Zhou Zhao**[1*], **Jun Yu**[2], **Qi Tian**[3]

[1] Department of Computer Science and Technology, Zhejiang University
[2] School of Computer Science and Technology, Hangzhou Dianzi University
[3] Huawei Cloud & AI
ckczzj@zju.edu.cn, zhaozhou@zju.edu.cn, yujun@hdu.edu.cn, tian.qi1@huawei.com

Abstract

Diffusion models have recently exhibited remarkable abilities to synthesize striking image samples since the introduction of denoising diffusion probabilistic models (DDPMs). Their key idea is to disrupt images into noise through a fixed forward process and learn its reverse process to generate samples from noise in a denoising way. For conditional DDPMs, most existing practices relate conditions only to the reverse process and fit it to the reversal of unconditional forward process. We find this will limit the condition modeling and generation in a small time window. In this paper, we propose a novel and flexible conditional diffusion model by introducing conditions into the forward process. We utilize extra latent space to allocate an exclusive diffusion trajectory for each condition based on some shifting rules, which will disperse condition modeling to all timesteps and improve the learning capacity of model. We formulate our method, which we call ShiftDDPMs, and provide a unified point of view on existing related methods. Extensive qualitative and quantitative experiments on image synthesis demonstrate the feasibility and effectiveness of ShiftDDPMs.

Introduction and Motivation

Deep generative models such as Generative Adversarial Networks (GANs) (Goodfellow et al. 2014), Variational Autoencoders (VAEs) (Kingma and Welling 2013), autoregressive models (Van Oord, Kalchbrenner, and Kavukcuoglu 2016) and normalizing flows (Rezende and Mohamed 2015) have shown remarkable abilities to model complex data distributions and synthesize high-quality samples in various fields. Diffusion models (Sohl-Dickstein et al. 2015) are recently brought back into focus by denoising diffusion probabilistic models (DDPMs) (Ho, Jain, and Abbeel 2020), which exhibits competitive image synthesis results and has been applied in a wide range of data modalities.

Generally, DDPMs gradually disrupt images by adding noise through a fixed forward process and learn its reverse process to generate samples from noise in a denoising way. There are two main methods to achieve conditional DDPMs. One is to learn an estimator that can compute the similarity between conditions and noisy data and use it to guide pretrained unconditional DDPMs to sample towards specified conditions (Dhariwal and Nichol 2021). Another is to train a conditional DDPM from scratch by incorporating conditions into the function approximator of the reverse process. Both methods try to fit their conditional reverse process to the reversal of fixed unconditional forward process. This brings up a question: *Can we design a more effective forward process utilizing given conditions to form a new type of conditional DDPMs and benefit from it?*

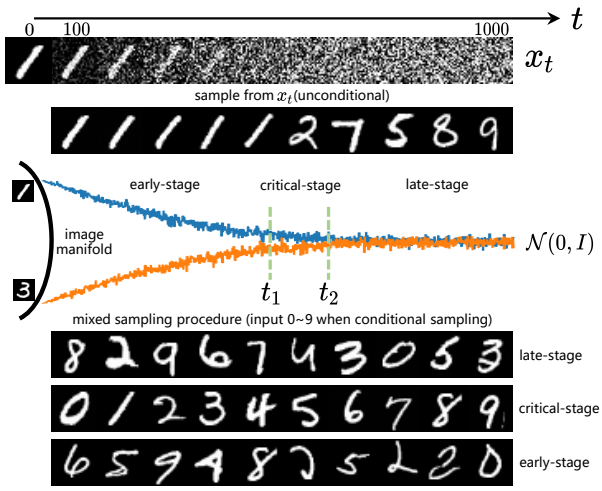

Figure 1: Exploration of the mechanism of conditional DDPMs. We grid 1000 timesteps with a step size of 50 and perform grid-search for (t_1, t_2) paris to find the shortest critical-stage that can ensure high accuracy of conditional generation. For MNIST, it is $(400, 600)$.

We investigate this question by exploring the mechanism of how conditional DDPMs achieve conditional sampling based on unconditional forward process, similar to that in PDAE (Zhang, Zhao, and Lin 2022). We conduct some experiments, shown in Figure 1. Concretely, we train an unconditional DDPM and a conditional one on MNIST (LeCun et al. 1998), respectively. The conditional one incorporates class labels (one-hot vector) into the function approximator of parameterized reverse process. The top two rows respectively show the latents $\boldsymbol{x}_t$ sampled from $\boldsymbol{x}_0$ for various t and the samples generated by the unconditional DDPM

*Corresponding author.
Copyright © 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

starting from corresponding latents. Intuitively, the latents for smaller t preserve more high-level information (such as class) of corresponding data, and they will be totally lost when t is large enough. It means that the diffusion trajectories originating from different data will get entangled, and the latents will become indistinguishable when t is large. We then divide the diffusion trajectories into three stages: early-stage ($0 \sim t_1$), critical-stage ($t_1 \sim t_2$) and late-stage ($t_2 \sim T$). Then we design a mixed sampling procedure that employs unconditional sampling but switches to conditional sampling during the specified stage. Note that the unconditional and conditional reverse process can be connected because they are trained to approximate the same forward process so that they recognize the same pattern of latents. The bottom three rows show the samples generated by three different mixed sampling procedures, where each row only employs conditional sampling for the right stage. As we can see, only the samples conditioned on input labels during critical-stage match the input class labels.

These phenomena show that, for unconditional forward process, the key to achieve conditional sampling is to shift and separate the generative trajectories of different conditions during critical-stage. Besides, to some extent, the training and sampling during early and late stages are independent of conditions and leave the condition modeling and generation to the limited critical-stage. If we can utilize extra latent space and allocate an exclusive diffusion trajectory for each condition to make the trajectories of different conditions disentangled all the time, it will disperse condition modeling to all timesteps and may improve the learning capacity of model.

Recently, Grad-TTS (Popov et al. 2021) and PriorGrad (Lee et al. 2021) introduce conditional forward process with data-dependent priors for audio diffusion models and enable more efficient training than those with unconditional forward process. However, their differences and connections have not been discussed, and there has not been a comprehensive exploration of this kind of methods, especially for image diffusion models. In this work, we systematically study how to design controllable diffusion trajectories according to conditions and its effect for conditional diffusion models. Our main contributions contain:

- We systemically introduce conditional forward process for diffusion models and provide a unified point of view on existing related approaches.
- By shifting diffusion trajectories, ShiftDDPMs improve the utilization rate of latent space and the learning capacity of model.
- We demonstrate the feasibility and effectiveness of ShiftDDPMs on various image synthesis tasks with extensive experiments.

Related Works

Diffusion models (Sohl-Dickstein et al. 2015; Ho, Jain, and Abbeel 2020) are an emerging family of generative models and have exhibited remarkable abilities to synthesize high-quality samples. Numerous studies (Song et al. 2020; Song, Meng, and Ermon 2020; Dhariwal and Nichol 2021; Liu et al. 2022) and applications (Chen et al. 2020; Saharia et al. 2022; Huang et al. 2022a,b; Ye et al. 2022, 2023) have further improved and expanded diffusion models. Among existing practices of conditional diffusion models, only Grad-TTS (Popov et al. 2021) and PriorGrad (Lee et al. 2021) involve conditions in forward process but, nonetheless, they are totally different methods. We will demonstrate their differences under the point of view of ShiftDDPMs.

ShiftDDPMs

Background

DDPMs (Ho, Jain, and Abbeel 2020) employ a forward process that sequentially destroys data distribution $q(\boldsymbol{x}_0)$ into $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ with Markov diffusion kernels defined by a fixed variance schedule $\{\beta_t\}_{t=1}^T$:

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t\boldsymbol{I}), \tag{1}$$

which admits sampling $\boldsymbol{x}_t$ from $\boldsymbol{x}_0$ for any timestep t in closed form:

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\boldsymbol{I}). \tag{2}$$

Then a parameterized Markov chain is trained to fit the reversal of forward process, denoising an arbitrary Gaussian noise to a data sample:

$$p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t), \boldsymbol{\Sigma}_\theta(\boldsymbol{x}_t, t)). \tag{3}$$

Training is performed by maximizing the model log likelihood with some parameterization and simplication:

$$L(\theta) = \mathbb{E}_{t,\boldsymbol{x}_0,\epsilon}\left[\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\|^2\right]. \tag{4}$$

See Appendix A for full details of DDPMs.

Conditional Forward Process

We aim to shift the diffusion trajectories in some way related to conditions. An intuitive way is to directly rewrite the Gaussian distribution in Eq.(2) as:

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0, \boldsymbol{c}) = \mathcal{N}(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + k_t \cdot \boldsymbol{E}(\boldsymbol{c}), (1-\bar{\alpha}_t)\boldsymbol{\Sigma}(\boldsymbol{c})). \tag{5}$$

Specifically, $k_t \cdot \boldsymbol{E}(\boldsymbol{c})$ is the cumulative mean shift of diffusion trajectories at t-th step, where k_t is a shift coefficient schedule that decides the shift mode and $\boldsymbol{E}(\cdot)$ is a function which we call shift predictor that maps conditions into the latent space. $\boldsymbol{\Sigma}(\boldsymbol{c})$ is a diagonal covariance matrix, where $\boldsymbol{\Sigma}(\cdot)$ is some function similar to $\boldsymbol{E}(\cdot)$. Comparing the diffusion trajectories to water pipes, then $k_t \cdot \boldsymbol{E}(\boldsymbol{c})$ is employed to change their directions and $\boldsymbol{\Sigma}(\boldsymbol{c})$ is employed to change their size in latent space. Note that both $\boldsymbol{E}(\cdot)$ and $\boldsymbol{\Sigma}(\cdot)$ can be fixed or trainable. In our experiments on image synthesis, trainable $\boldsymbol{\Sigma}(\cdot)$ leads to complex training and sampling procedure, unstable training and poor results, so we fix $\boldsymbol{\Sigma}(\boldsymbol{c}) = \boldsymbol{I}$ like that in Eq.(2). For generalization, we still use $\boldsymbol{\Sigma}(\boldsymbol{c})$ in our derivations. For simplicity, we use following substitution:

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0, \boldsymbol{c}) = \mathcal{N}(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \boldsymbol{s}_t, (1-\bar{\alpha}_t)\boldsymbol{\Sigma}), \tag{6}$$

where $\boldsymbol{s}_t = k_t \cdot \boldsymbol{E}(\boldsymbol{c})$ and $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}(\boldsymbol{c})$. We will discuss how to choose k_t and $\boldsymbol{E}(\cdot)$ in later sections.

With Eq.(6), we can derive corresponding forward diffusion kernels (See proof in Appendix A):

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{c}) = \mathcal{N}(\sqrt{\alpha_t}\boldsymbol{x}_{t-1} + \boldsymbol{s}_t - \sqrt{\alpha_t}\boldsymbol{s}_{t-1}, \beta_t\boldsymbol{\Sigma}), \tag{7}$$

where $\boldsymbol{s}_0 = \boldsymbol{0}$ (i.e. $k_0 = 0$). Intuitively, our forward diffusion kernels introduce a small perturbation conditioned on $\boldsymbol{c}$ to original ones shown in Eq.(1).

With Eq.(6) and Eq.(7), the posterior distributions of forward steps for $t > 1$ can be derived from Bayes' rule (See proof in Appendix A):

$$\begin{aligned} q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0, \boldsymbol{c}) = \mathcal{N}(&\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\boldsymbol{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\boldsymbol{x}_t \\ &- \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\boldsymbol{s}_t + \boldsymbol{s}_{t-1}\,,\ \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t\boldsymbol{\Sigma})\,. \end{aligned} \tag{8}$$

Parameterized Reverse Process

The reverse process starts at $p(\boldsymbol{x}_T) = \mathcal{N}(\boldsymbol{s}_T, \boldsymbol{\Sigma})$, which is an approximation of $q(\boldsymbol{x}_T|\boldsymbol{x}_0, \boldsymbol{c})$, and employs parameterized kernels $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{c})$ to fit $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0, \boldsymbol{c})$.

According to Eq.(6), $\boldsymbol{x}_0$ can be represented as:

$$\boldsymbol{x}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}\big(\boldsymbol{x}_t - \boldsymbol{s}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}\big)\,, \tag{9}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$. Then we take it into Eq.(8) and derive the posterior mean of forward steps:

$$\begin{aligned} \mathbb{E}\big[q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0, \boldsymbol{c})\big] =& \frac{1}{\sqrt{\alpha_t}}(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}) \\ &- \frac{1}{\sqrt{\alpha_t}}\boldsymbol{s}_t + \boldsymbol{s}_{t-1}\,, \end{aligned} \tag{10}$$

where all things are available except $\boldsymbol{\epsilon}$. We can employ a model $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)$ to predict $\boldsymbol{\epsilon}$. Note that there is no need to feed $\boldsymbol{c}$ into $\boldsymbol{\epsilon}_\theta$ because we have encoded it into condition-dependent trajectories (i.e., in $\boldsymbol{x}_t$) so that the model does not need its guidance.

Further improvements come from another parameterization because $\boldsymbol{\epsilon}$ in Eq.(9) is given by:

$$\boldsymbol{\epsilon} = \frac{\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0}{\sqrt{1-\bar{\alpha}_t}} - \frac{\boldsymbol{s}_t}{\sqrt{1-\bar{\alpha}_t}}\,, \tag{11}$$

where the second term is available. Therefore we can employ a model $\boldsymbol{g}_\theta(\boldsymbol{x}_t, t)$ to predict the first term for training. We find this parameterization achieves better performance than predicting $\boldsymbol{\epsilon}$ directly. Then we can get the predicted posterior distributions parameterized by θ:

$$\begin{aligned} p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{c}) = \mathcal{N}(&\frac{1}{\sqrt{\alpha_t}}\left[\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{g}_\theta(\boldsymbol{x}_t, t)\right] \\ &- \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\boldsymbol{s}_t + \boldsymbol{s}_{t-1}, \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t\boldsymbol{\Sigma}). \end{aligned} \tag{12}$$

Algorithm 1: Training

repeat
  $\boldsymbol{x}_0, \boldsymbol{c} \sim q(\boldsymbol{x}_0)$
  $t \sim \text{Uniform}(\{1, \dots, T\})$
  $\boldsymbol{s}_t = k_t \cdot \boldsymbol{E}(\boldsymbol{c})\,,\ \boldsymbol{\Sigma} = \boldsymbol{\Sigma}(\boldsymbol{c})$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$
  $\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \boldsymbol{s}_t + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$
  Optimize $\|\frac{\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0}{\sqrt{1-\bar{\alpha}_t}} - \boldsymbol{g}_\theta(\boldsymbol{x}_t, t)\|^2_{\boldsymbol{\Sigma}^{-1}}$
until converged

Algorithm 2: Sampling

$\boldsymbol{s}_T = k_T \cdot \boldsymbol{E}(\boldsymbol{c})\,,\ \boldsymbol{\Sigma} = \boldsymbol{\Sigma}(\boldsymbol{c})$
$\boldsymbol{x}_T \sim \mathcal{N}(\boldsymbol{s}_T, \boldsymbol{\Sigma})$
for $t = T, \dots, 1$ **do**
  $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$ if $t > 1$, else $\boldsymbol{z} = \boldsymbol{0}$
  $\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\big[\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{g}_\theta(\boldsymbol{x}_t, t)\big]$
    $- \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\boldsymbol{s}_t + \boldsymbol{s}_{t-1} + \sqrt{\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t}\boldsymbol{z}$
end for
return $\boldsymbol{x}_0$

Training Objective

With our conditional forward process and corresponding reverse process, our training objective can be represented as (See proof in Appendix A):

$$L = c + \sum_{t=1}^{T} \gamma_t \mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{\epsilon}}\left[\|\frac{\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0}{\sqrt{1-\bar{\alpha}_t}} - \boldsymbol{g}_\theta(\boldsymbol{x}_t, t)\|^2_{\boldsymbol{\Sigma}^{-1}}\right], \tag{13}$$

where c is some constant, $\boldsymbol{x}_0 \sim q(\boldsymbol{x_0})$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$, $\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \boldsymbol{s}_t + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$, $\|\boldsymbol{x}\|^2_{\boldsymbol{\Sigma}^{-1}} = \boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x}$, $\gamma_1 = \frac{1}{2\alpha_1}$ and $\gamma_t = \frac{\beta_t}{2\alpha_t(1-\bar{\alpha}_{t-1})}$ for $t \geq 2$. During training, we follow DDPMs (Ho, Jain, and Abbeel 2020) to adopt the simplified training objective by uniformly sampling t between 1 and T and ignoring loss weight γ_t. Algorithm 1 and Algorithm 2 describe our training and sampling procedure. Note that $\boldsymbol{E}(\cdot)$ and $\boldsymbol{\Sigma}(\cdot)$ will be optimized along with θ if they are trainable.

Intuitive Interpretation

Assume that $\boldsymbol{\Sigma} = \boldsymbol{I}$ and $\boldsymbol{x}'_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$, DDPMs employ $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}'_t, \boldsymbol{c}, t)$ to predict $\epsilon = \frac{\boldsymbol{x}'_t - \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0}{\sqrt{1-\bar{\alpha}_t}}$, while ShiftDDPMs employ $\boldsymbol{g}_\theta(\boldsymbol{x}'_t + \boldsymbol{s}_t, t)$ to predict $\frac{\boldsymbol{x}'_t + \boldsymbol{s}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0}{\sqrt{1-\bar{\alpha}_t}}$. They are trained to predict the same pattern of objective but with different input (i.e. $\frac{\text{input} - \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0}{\sqrt{1-\bar{\alpha}_t}}$). Compared with DDPMs, ShiftDDPMs transfer input condition $\boldsymbol{c}$ onto diffusion trajectories by shifting $\boldsymbol{x}'_t$ to $\boldsymbol{x}'_t + \boldsymbol{s}_t$, which allows conditional training and sampling without feeding $\boldsymbol{c}$ into the network. For DDPMs, only the training and sampling during limited critical-stage plays a key role for condition modeling and generation, while ShiftDDPMs disperse it to all timesteps and improve the utilization rate of latent space, which may lead to a better performance.

Furthermore, if $\boldsymbol{E}(\cdot)$ is trainable, it will be optimized to find an optimal shift in latent space to specialize the diffu-

sion trajectories of different conditions and make them disentangle as much as possible. The term $\boldsymbol{d}_t = -\frac{1}{\sqrt{\alpha_t}}\boldsymbol{s}_t + \boldsymbol{s}_{t-1}$ in Eq.(10) will amend the sampling trajectories in every step to ensure they can finally fall on the data manifold.

Next, we will show that the forward process of Grad-TTS (Popov et al. 2021) and PriorGrad (Lee et al. 2021) correspond to a special choice of k_t, respectively.

Prior-Shift

Grad-TTS (Popov et al. 2021) proposes a score-based text-to-speech generative model with the prior mean predicted by text encoder and aligner. Specifically, it defines a forward process satisfying the following SDE:

$$\mathrm{d}\boldsymbol{X}_t = \frac{1}{2}(\boldsymbol{\mu} - \boldsymbol{X}_t)\beta_t \mathrm{d}t + \sqrt{\beta_t}\mathrm{d}\boldsymbol{W}_t\,, \tag{14}$$

where $\boldsymbol{\mu}$ corresponds to $\boldsymbol{E}(\boldsymbol{c})$ of our system ($\boldsymbol{E}(\cdot)$ represents the parameterized text encoder and aligner, $\boldsymbol{c}$ represents the input text). We show that $k_t = 1 - \sqrt{\bar{\alpha}_t}$ match a discretization of Eq.(14) (See proof in Appendix A). For forward process, k_t increases from 0 to 1 and leads $\boldsymbol{x}_t$ to shift to $\boldsymbol{\mu}$ as t increases. For reverse process, we have:

$$\boldsymbol{d}_t = (1 - \frac{1}{\sqrt{\alpha_t}})\boldsymbol{\mu}\,, \tag{15}$$

where $1 - \frac{1}{\sqrt{\alpha_t}} < 0$ because the reverse process starts from $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{I})$ and it needs to eliminate the cumulative shift $\boldsymbol{\mu}$ of forward process. From the view of diffusion trajectories, Grad-TTS changes the ending point of trajectories, so we name the shift mode as Prior-Shift.

Note that Grad-TTS still takes $\boldsymbol{\mu}$ as an additional input to the score estimator, but we have stated that it is unnecessary. However, doing this will get at least not worse results, but also introduces additional parameter and computation.

Data-Normalization

PriorGrad (Lee et al. 2021) employs a forward process as follows:

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}(\boldsymbol{x}_0 - \boldsymbol{\mu}) + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}\,, \tag{16}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. Obviously, $k_t = -\sqrt{\bar{\alpha}_t}$ satisfies Eq.(16). For forward process, it first normalizes $\boldsymbol{x}_0$ by subtracting its corresponding prior mean $\boldsymbol{\mu}$ and then trains a diffusion model on normalized $\boldsymbol{x}_0$ with prior $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. For reverse process, we have:

$$\boldsymbol{d}_1 = \boldsymbol{\mu}\,,\ \boldsymbol{d}_{t>1} = \mathbf{0}. \tag{17}$$

Intuitively, the reverse process starts from $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ and has no amendments all the time except the last step, where it adds prior mean $\boldsymbol{\mu}$ to the output (denormalization). From the view of diffusion trajectories, PriorGrad resets the starting point of trajectories on the data manifold, so we name the shift mode as Data-Normalization.

Unlike Prior-Shift that disperses the cumulative shift to all points on the diffusion trajectories, Data-Normalization does not disentangle the diffusion trajectories so that it must feed $\boldsymbol{c}$ into the network to guide sampling. However, by carefully designing $\boldsymbol{\Sigma}$, it can achieve the same precision with a simpler network and have a faster convergence rate under some constraints (Lee et al. 2021). Data-Normalization is more suitable for variance-sensitive data such as audio.

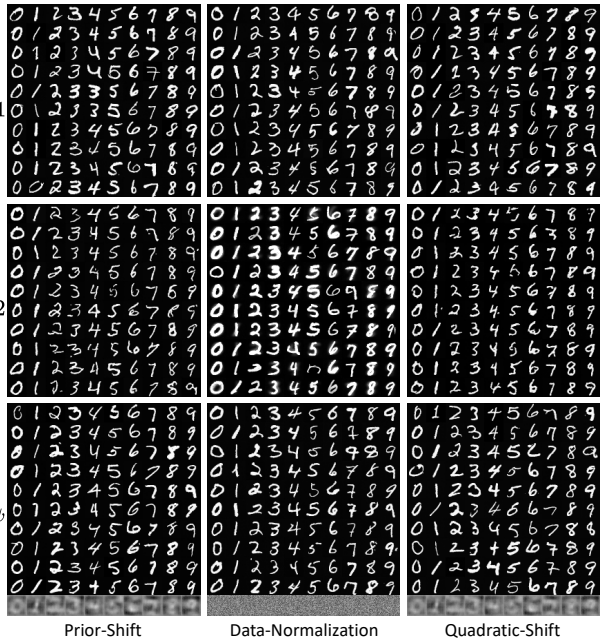

Figure 2: 32×32 conditional MNIST samples for different shift modes with different shift predictors. The last row visualize the learned $\boldsymbol{E}_\psi(\cdot)$.

Quadratic-Shift

Except for Prior-Shift, we propose a shift mode to disentangle the diffusion trajectories of different conditions by making the concave trajectories shown in Figure 1 convex. In this case, we don't change their starting or ending point, and $\boldsymbol{E}(\boldsymbol{c})$ becomes a middle point, where they first progress to it and then go away from it. Therefore k_t should be similar to some quadratic function opening downwards with $k_1 \approx 0$ and $k_T \approx 0$. Empirically, we choose $k_t = \sqrt{\bar{\alpha}_t}(1 - \sqrt{\bar{\alpha}_t})$. We name the shift mode as Quadratic-Shift.

Experiments

In this section, we conduct several conditional image synthesis experiments with ShiftDDPMs. Note that we always set $\boldsymbol{\Sigma}(\boldsymbol{c}) = \boldsymbol{I}$. Full implementation details of all experiments can be found in Appendix B.

Effectiveness of Conditional Sampling

We first verify the effectiveness of ShiftDDPMs with three shift modes on toy dataset MNIST (LeCun et al. 1998). We employ two fixed shift predictors ($\boldsymbol{E}_1(\cdot)$ and $\boldsymbol{E}_2(\cdot)$) and a trainable one ($\boldsymbol{E}_\psi(\cdot)$ with parameters ψ), mapping a one-hot vector $\boldsymbol{c}$ to a 32×32 matrix. Specifically, $\boldsymbol{E}_1(\cdot)$ takes 10 evenly spaced numbers over $[-1, 1]$ and expands each number into a 32×32 matrix. $\boldsymbol{E}_2(\cdot)$ takes the mean of all training

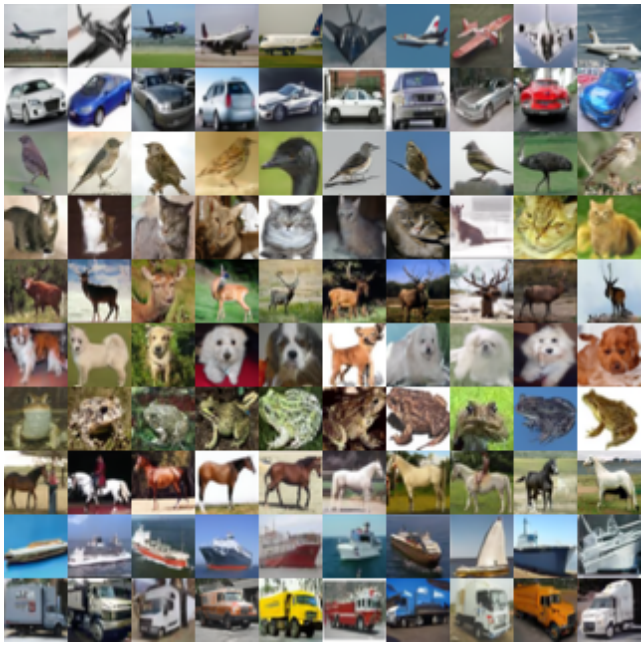

Figure 3: 32×32 conditional CIFAR-10 samples for Quadratic-Shift.

data belonging to the specified class. $\boldsymbol{E}_\psi(\cdot)$ employs stacked transposed convolution layers to compute the matrix.

Figure 2 presents the conditional MNIST samples for different shift modes with different shift predictors. As we can see, all models work for conditional generation, and the visualization of learned $\boldsymbol{E}_\psi(\boldsymbol{c})$ for Prior-Shift and Quadratic-Shift contain the general shape of corresponding class, which means that they learn specialized trajectories for different conditions. Data-Normalization must feed $\boldsymbol{c}$ into the model so it may ignore the shift.

Despite the success of the fixed shift predictor on MNIST, we get poor sample results when modeling complex data distribution such as CIFAR-10. Therefore we will always employ trainable shift predictor $\boldsymbol{E}_\psi$ with parameter ψ in the following experiments.

Sample Quality

We further evaluate ShiftDDPMs on CIFAR-10 (Krizhevsky and Hinton 2009). For a fair comparison, we retrain a DDPM as baseline (our DDPM) and then use the same experimental settings and resources to train other models. We train a traditional conditional DDPM (cond. DDPM) by incorporating class labels into the function approximator of reverse process. Moreover, we train a time-dependent classifier (Sohl-Dickstein et al. 2015; Song et al. 2020; Dhariwal and Nichol 2021) on noisy images and use its gradients to guide (our DDPM) to sample towards specified class (cls. DDPM). For ShiftDDPMs, we train three models, including Prior-Shift, Data-Normalization, and Quadratic-Shift, all with trainable shift predictors. Furthermore, we employ another two models (cond. Prior-Shift and cond. Quadratic-Shift) by incorporating class labels into the reverse process of Prior-Shift and Quadratic-Shift, with the same method with (cond. DDPM). Figure 3 presents some conditional CIFAR-10 samples generated by Quadratic-Shift. Table 1 shows Inception Score, FID, negative log likelihood for these models.

Model	IS↑	FID↓	NLL↓
Unconditional			
DDPM	9.46	3.17	≤ 3.75
our DDPM	9.52	3.13	≤ 3.72
Conditional			
cond. DDPM	9.59	3.12	≤ 3.74
cls. DDPM	9.17	5.85	–
Prior-Shift	9.54	3.06	≤ 3.71
cond. Prior-Shift	9.65	3.06	≤ 3.70
Data-Normalization	9.14	5.51	–
Quadratic-Shift	9.67	3.05	$\leq \mathbf{3.69}$
cond. Quadratic-Shift	**9.74**	**3.02**	≤ 3.70

Table 1: Quantitative results of conditional sample quality on CIFAR-10. NLL measured in bits/dim.

As we can see, our retrained unconditional DDPM is slightly better than the original one with the help of improved settings. With the help of conditional knowledge, conditional DDPM outperforms unconditional DDPM. Classifer-guided DDPM has poor results because it is sensitive to the classifier. Data-Normalization has an unstable training process and poor results, which means that it is not suitable for image synthesis. Both Prior-Shift and Quadratic-Shift outperform conditional DDPM, which proves that conditional forward process can improve the learning capacity of ShiftDDPMs. Although incorporating class labels can slightly improve their performance, it also introduces additional computational and parameter complexity.

Adaption to DDIM for Fast Sampling

DDIMs (Song, Meng, and Ermon 2020) generalize the forward process of DDPMs to non-Markovian process with an equivalent objective for training, which enables us to employ an accelerated reverse process with pre-trained DDPMs. Fortunately, ShiftDDPMs can be adapted to ShiftDDIMs. Specifically, we can generate $\boldsymbol{x}_{t-1}$ from $\boldsymbol{x}_t$ via:

$$\begin{aligned}\boldsymbol{x}_{t-1} = &\frac{1}{\sqrt{\alpha_t}}\left[\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{g}_\theta(\boldsymbol{x}_t, t)\right] + \boldsymbol{s}_{t-1} \\ &+ \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\cdot\left[\boldsymbol{g}_\theta(\boldsymbol{x}_t,t) - \frac{\boldsymbol{s}_t}{\sqrt{1-\bar{\alpha}_t}}\right] + \sigma_t\boldsymbol{\epsilon}_t,\end{aligned} \tag{18}$$

where $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ (See proof in Appendix A).

Then we employ $\tau = \{\tau_1, \cdots, \tau_S\}$, which is an increasing sub-sequence of $[1, \cdots, T]$ of length S, for accelerated sampling. The corresponding variance become $\sigma_{\tau_i}(\eta) = \eta\sqrt{\frac{1-\bar{\alpha}_{\tau_{i-1}}}{1-\bar{\alpha}_{\tau_i}}}\sqrt{1-\frac{\bar{\alpha}_{\tau_i}}{\bar{\alpha}_{\tau_{i-1}}}}$, where η is a hyperparameter that we can directly control. Figure 4 and Table 2 presents the conditional CIFAR-10 samples generated by

Figure 4: 32×32 conditional CIFAR-10 samples for Quadratic-Shift. We use fixed input and noise during sampling.

S		10	20	50	100
η	0.0	14.25	7.95	5.22	3.93
	0.2	14.16	7.88	5.30	4.06
	0.5	16.96	9.12	6.18	4.43
	1.0	25.33	11.67	9.81	5.70
$\hat{\sigma}$		264.32	118.61	36.24	10.95

Table 2: FID of conditional sample quality on CIFAR-10 for Quadratic-Shift.

Quadratic-Shift mode and its FID with different sampling steps and η. ShiftDDIMs can still keep competitive FID even though it only samples for 100 steps.

Interpolation of Diffusion Trajectories

DDPMs (Ho, Jain, and Abbeel 2020) show that one can interpolate the latents of two source data, decode the interpolated latent by the reverse process and get a sample similar to the interpolation of two source data. Inspired by this phenomenon, we can try to interpolate the diffusion trajectories of different conditions, which is equivalent to interpolating between different $\boldsymbol{s}_t$, such as $\hat{\boldsymbol{s}}_t = \lambda \cdot k_t \cdot \boldsymbol{E}_\psi(\boldsymbol{c}_1) + (1-\lambda) \cdot k_t \cdot \boldsymbol{E}_\psi(\boldsymbol{c}_2)$ for two different conditions $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$. In theory, $\boldsymbol{s}_t$ decide the direction of diffusion trajectories and the interpolated $\hat{\boldsymbol{s}}_t$ will take the median direction, which can lead the reverse process to generate the samples with the mixed features of $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$.

We verify this idea by conducting the experiments of attribute-to-image (Yan et al. 2016) on LFW dataset (Huang et al. 2008). Specifically, it requires us to generate facial images according to the input attributes. Each image ($\boldsymbol{x}_0$) in LFW corresponds to a 73-dim real-valued vector ($\boldsymbol{c}$), where the value of each dimension represents the degree of some attribute such as male, beard and so on. We employ Quadratic-Shift with a trainable shift predictor to train on the training set and evaluate it on the test set. Figure 5 presents some samples, which shows that ShiftDDPMs can learn a meaningful shift (like a heatmap of the face), and the generated images are consistent with the ground truth in labeled face attributes. Figure 6 presents the interpolations generated by Quadratic-Shift. The interpolations smoothly transition from one side to the other, which verifies our assumptions about the disentangled diffusion trajectories.

Figure 5: 64×64 conditional LFW samples for Quadratic-Shift. From left to right are ground truth image (from test set), generated image and learned $\boldsymbol{E}_\psi(\boldsymbol{c})$.

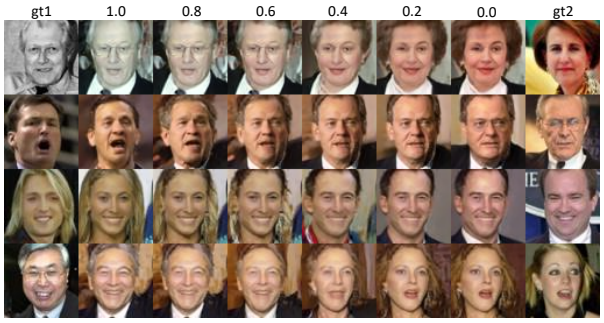

Figure 6: 64×64 conditional LFW interpolations for Quadratic-Shift. We use fixed input and noise during sampling.

Image Inpainting

Except for class-conditional image synthesis, we conduct some image-to-image synthesis experiments. Compared with enumerable class label, image space is almost infinite and it is a challenge to assign a unique trajectory for each instance. To prove the capacity of ShiftDDPMs, we conduct image inpainting experiments using Irregular Mask Dataset (Liu et al. 2018) with three image datasets: CelebA-HQ (Liu et al. 2015), LSUN-church (Yu et al. 2015) and Places2 (Zhou et al. 2017). We employ Quadratic-Shift mode and a UNet based architecture as a shift predictor, which takes as input the masked image and predicts the shift. Figure 7 presents some inpainting samples. As we can see, ShiftDDPMs predict a template of complete image based on the masked one, which guides the trajectory to generate consistent and diverse completions. To further evaluate ShiftDDPMs on image inpainting, we follow prior works (Yu et al. 2019; Liu et al. 2018; Zhang et al. 2020) by reporting FID on Places2 dataset. We choose several GAN-based models: Contextual Attention (Yu et al. 2018), EdgeConnect (Nazeri et al. 2019) and StructureFlow (Ren et al. 2019) as baselines. Besides, we take score-based inpainting method proposed in (Song et al. 2020) as another baseline. Table 3 presents the quantitative results, and ShiftDDPMs achieve competitive results comparable to prior GAN-based methods. In addition, ShiftDDPMs also outperform the score-based inpainting method, showing that the extra utilization of the latent space to some extent improves the learning capacity of

Figure 7: 256 × 256 inpainting samples from CelebA-HQ and LSUN-church test set for Quadratic-Shift.

Mask Percentage	0-20%	20-40%	40-60%
Contextual Attention	4.8586	18.4190	37.9432
EdgeConnect	3.0097	7.2635	19.0030
StructureFlow	2.9420	7.0354	22.3803
DDPM (score)	2.0665	6.6129	17.3601
Quadratic-Shift	**1.8314**	**6.2915**	**14.9667**

Table 3: FID of inpainting results on Places2 dataset.

diffusion models.

Text-to-Image

We conduct text-to-image (text2img) experiments on CUB dataset (Wah et al. 2011). We employ Quadratic-Shift mode and a network as shift predictor to generate shift from the pre-trained sentence embeddings. Figure 8 presents some generated samples. We can see that the shift predictor can predict a meaningful template according to text and guide the trajectory to generate text-consistent images. We choose several GAN-based models GAN-INT-CLS (Reed et al. 2016), StackGAN (Zhang et al. 2017), StackGAN++ (Zhang et al. 2018) and AttnGAN (Xu et al. 2018) as baselines. Besides, we take traditional conditional diffusion method as another baseline, which only incorporates sentence embeddings into the function approximator of parameterized reverse process. Table 4 presents some quantitative results, and ShiftDDPMs achieve competitive results comparable to prior GAN-based methods and traditional conditional diffusion model.

More Choice of k_t

The choice of k_t is flexible. For Prior-Shift, any schedules of k_t monotonically increasing from 0 to 1 can be applied on Prior-Shift. We have tried with following three types k_t: $\frac{t}{T}$, $(\frac{t}{T})^2$ and $\sin(\frac{t\pi}{2T} - \frac{\pi}{2})$ and they all work well. Furthermore, k_t can also be piecewise:

$$k_t = \begin{cases} 0 & t < 0.4T \\ \frac{t-0.4T}{0.6T} & \text{otherwise} \end{cases}. \tag{19}$$

One can also design other reasonable k_t. We leave empirical investigations of k_t as future work.

Figure 8: 256 × 256 text2img samples from CUB test set for Quadratic-Shift. From left to right are text, learned shift, generated sample and ground truth, respectively.

Methods	IS	FID
GAN-INT-CLS	2.88	68.79
StackGAN	3.70	51.89
StackGAN++	3.82	15.30
AttnGAN	4.36	-
cond. DDPM	4.18	14.79
Quadratic-Shift	**4.42**	**14.26**

Table 4: IS and FID of text2img results on CUB dataset.

Conclusion

In this work, we propose a novel and flexible conditional diffusion model called ShiftDDPMs by introducing conditional forward process with controllable condition-dependent diffusion trajectories. We analyze the differences of existing related methods under the point of view of ShiftDDPMs and first apply them on image synthesis. With ShiftDDPMs, we can achieve a better performance and learn some interesting features in latent space. Extensive qualitative and quantitative experiments on image synthesis demonstrate the feasibility and effectiveness of ShiftDDPMs.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant No.62020106007, No.U21B2040, No.62222211 and No.202100023), Zhejiang Natural Science Foundation (LR19F020006), Zhejiang Electric Power Co., Ltd. Science and Technology Project No.5211YF220006 and Yiwise.

References

Chen, N.; Zhang, Y.; Zen, H.; Weiss, R. J.; Norouzi, M.; and Chan, W. 2020. WaveGrad: Estimating gradients for waveform generation. *arXiv preprint arXiv:2009.00713*.

Dhariwal, P.; and Nichol, A. 2021. Diffusion models beat gans on image synthesis. *arXiv preprint arXiv:2105.05233*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *arXiv preprint arXiv:2006.11239*.

Huang, G. B.; Mattar, M.; Berg, T.; and Learned-Miller, E. 2008. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*.

Huang, R.; Lam, M. W.; Wang, J.; Su, D.; Yu, D.; Ren, Y.; and Zhao, Z. 2022a. FastDiff: A Fast Conditional Diffusion Model for High-Quality Speech Synthesis. *arXiv preprint arXiv:2204.09934*.

Huang, R.; Zhao, Z.; Liu, H.; Liu, J.; Cui, C.; and Ren, Y. 2022b. Prodiff: Progressive fast diffusion model for high-quality text-to-speech. *arXiv preprint arXiv:2207.06389*.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Lee, S.-g.; Kim, H.; Shin, C.; Tan, X.; Liu, C.; Meng, Q.; Qin, T.; Chen, W.; Yoon, S.; and Liu, T.-Y. 2021. PriorGrad: Improving Conditional Denoising Diffusion Models with Data-Driven Adaptive Prior. *arXiv preprint arXiv:2106.06406*.

Liu, G.; Reda, F. A.; Shih, K. J.; Wang, T.-C.; Tao, A.; and Catanzaro, B. 2018. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 85–100.

Liu, L.; Ren, Y.; Lin, Z.; and Zhao, Z. 2022. Pseudo Numerical Methods for Diffusion Models on Manifolds. In *International Conference on Learning Representations*.

Liu, Z.; Luo, P.; Wang, X.; and Tang, X. 2015. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, 3730–3738.

Nazeri, K.; Ng, E.; Joseph, T.; Qureshi, F.; and Ebrahimi, M. 2019. Edgeconnect: Structure guided image inpainting using edge prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 0–0.

Popov, V.; Vovk, I.; Gogoryan, V.; Sadekova, T.; and Kudinov, M. 2021. Grad-tts: A diffusion probabilistic model for text-to-speech. *arXiv preprint arXiv:2105.06337*.

Reed, S.; Akata, Z.; Yan, X.; Logeswaran, L.; Schiele, B.; and Lee, H. 2016. Generative adversarial text to image synthesis. In *International Conference on Machine Learning*, 1060–1069. PMLR.

Ren, Y.; Yu, X.; Zhang, R.; Li, T. H.; Liu, S.; and Li, G. 2019. Structureflow: Image inpainting via structure-aware appearance flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 181–190.

Rezende, D.; and Mohamed, S. 2015. Variational inference with normalizing flows. In *International conference on machine learning*, 1530–1538. PMLR.

Saharia, C.; Chan, W.; Saxena, S.; Li, L.; Whang, J.; Denton, E.; Ghasemipour, S. K. S.; Ayan, B. K.; Mahdavi, S. S.; Lopes, R. G.; et al. 2022. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. *arXiv preprint arXiv:2205.11487*.

Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, 2256–2265. PMLR.

Song, J.; Meng, C.; and Ermon, S. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B. 2020. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

Van Oord, A.; Kalchbrenner, N.; and Kavukcuoglu, K. 2016. Pixel recurrent neural networks. In *International Conference on Machine Learning*, 1747–1756. PMLR.

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology.

Xu, T.; Zhang, P.; Huang, Q.; Zhang, H.; Gan, Z.; Huang, X.; and He, X. 2018. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1316–1324.

Yan, X.; Yang, J.; Sohn, K.; and Lee, H. 2016. Attribute2image: Conditional image generation from visual attributes. In *European Conference on Computer Vision*, 776–791. Springer.

Ye, Z.; Jiang, Z.; Ren, Y.; Liu, J.; He, J.; and Zhao, Z. 2023. GeneFace: Generalized and High-Fidelity Audio-Driven 3D Talking Face Synthesis. *arXiv preprint arXiv:2301.13430*.

Ye, Z.; Zhao, Z.; Ren, Y.; and Wu, F. 2022. SyntaSpeech: Syntax-aware Generative Adversarial Text-to-Speech. *arXiv preprint arXiv:2204.11792*.

Yu, F.; Seff, A.; Zhang, Y.; Song, S.; Funkhouser, T.; and Xiao, J. 2015. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*.

Yu, J.; Lin, Z.; Yang, J.; Shen, X.; Lu, X.; and Huang, T. S. 2018. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5505–5514.

Yu, J.; Lin, Z.; Yang, J.; Shen, X.; Lu, X.; and Huang, T. S. 2019. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4471–4480.

Zhang, H.; Xu, T.; Li, H.; Zhang, S.; Wang, X.; Huang, X.; and Metaxas, D. N. 2017. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, 5907–5915.

Zhang, H.; Xu, T.; Li, H.; Zhang, S.; Wang, X.; Huang, X.; and Metaxas, D. N. 2018. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(8): 1947–1962.

Zhang, Z.; Zhao, Z.; and Lin, Z. 2022. Unsupervised Representation Learning from Pre-trained Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems*.

Zhang, Z.; Zhao, Z.; Zhang, Z.; Huai, B.; and Yuan, J. 2020. Text-guided image inpainting. In *Proceedings of the 28th ACM International Conference on Multimedia*, 4079–4087.

Zhou, B.; Lapedriza, A.; Khosla, A.; Oliva, A.; and Torralba, A. 2017. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6): 1452–1464.