Compensation Sampling for Improved Convergence in Diffusion Models

Hui Lu[1] (✉), Albert Ali Salah[1], and Ronald Poppe[1]

Utrecht University, Princetonplein 5, 3584 CC Utrecht, The Netherlands
{h.lu1, a.a.salah, r.w.poppe}@uu.nl

Abstract. Diffusion models achieve remarkable quality in image generation, but at a cost. Iterative denoising requires many time steps to produce high fidelity images. The denoising process is crucially limited by an accumulation of the reconstruction error due to an initial inaccurate reconstruction of the target data. This leads to lower quality outputs, and slower convergence. To address these issues, we propose *compensation sampling* to guide the generation towards the target domain. We introduce a compensation term, implemented as a U-Net, which adds negligible training overhead. Our approach is flexible and we demonstrate its application in unconditional generation, face inpainting, and face de-occlusion on benchmark datasets CIFAR-10, CelebA, CelebA-HQ, FFHQ-256, and FSG. Our approach consistently yields state-of-the-art results in terms of image quality, while accelerating the denoising process to converge during training by up to an order of magnitude.

Keywords: Diffusion models · Iterative denoising · Image generation

1 Introduction

Diffusion models have achieved great success in image generation [15, 32, 35, 36]. Compared to other deep generative models such as Generative Adversarial Networks (GANs) [9, 13], diffusion models offer stable training, easy model scaling, and good distribution coverage [33]. But despite their success, diffusion models suffer from low training and inference efficiency because they typically require many time steps to converge and to generate high-quality outputs. Compared to GANs, which only require a single forward pass through the generator network, inference in diffusion models is two to three orders of magnitude slower [42]. Simply reducing the number of time steps disrupts the Gaussian assumption of the denoising process, and has been shown to reduce the synthesis quality [41, 52].

Due to the iterative sampling, inaccurate reconstruction early in the training process causes the accumulation of reconstruction errors in subsequent time steps [28]. This hinders convergence speed and the final quality of the model. To address the issue of error accumulation, several works have encoded conditionals in the denoising process to improve the initial sample quality, and to speed up the process. These approaches include encoding image features or images

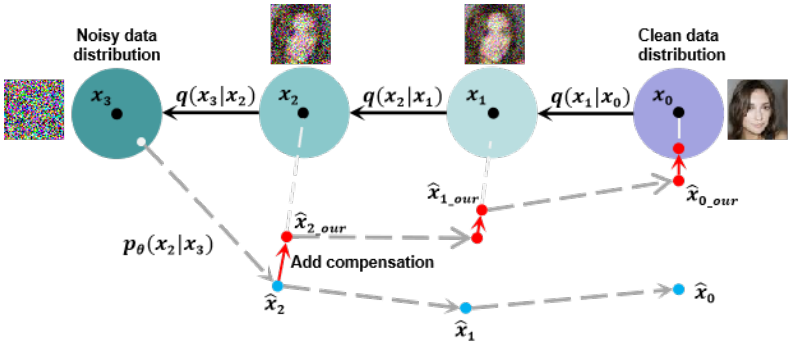

Fig. 1: Compensation (red) compared to traditional sampling (dashed) for $T = 3$. Both processes start from a noisy data distribution. Compensation sampling guides the reconstruction towards the clean data distribution for faster convergence.

generated by Variational Autoencoders (VAE) [35,36], and assigning prioritized weights on specific noise levels in the denoising process [7,8].

Conditioned denoising processes take fewer time steps to converge but the additional conditions often require significant computation cost. Conditioning is also arguably more dataset-specific, which could limit generalization. But more importantly, simply adding conditions does not address inherent issues in the denoising process. In other words, it does not avoid the error accumulation that causes slower convergence towards a sub-optimal model. The challenge thus remains to improve the efficiency during training and inference in a principled way without compromising the output quality. This is especially true for unconditional generation, in the absence of a powerful conditioning signal.

In this work, we address the efficiency issue of diffusion models by introducing *compensation sampling* (CS). It allows us to use 10 times fewer time steps during training and inference without breaking the Gaussian assumption of the denoising process. Compensation sampling can be applied in both unconditional and conditional generation tasks. An illustration of compensation sampling appears in Figure 1. At the core of our approach is the use of a learned compensation term to direct the reconstruction towards the clean data distribution, and consequently avoid error accumulation. We show that this process results in quicker training convergence, and higher-quality images. Our main contributions are:

1. **Novel sampling algorithm**. We propose the compensation sampling algorithm, with rigid mathematical derivation, that can reduce the number of time steps in diffusion models during training by an order of magnitude.
2. **State-of-the-art results**. We apply compensation sampling and achieve results that are on par with, and typically outperform, current state-of-the-art diffusion models on unconditional generation, face inpainting, and face de-occlusion on CIFAR-10, CelebA, CelebA-HQ, FFHQ, and FSG.

We discuss related work (Section 2), detail our method in Section 3, present our experiments and ablation study in Section 4, and conclude in Section 5.

2 Related Work

Denoising diffusion probabilistic models (DDPM) [15] have several advantages over other generative models such as Generative Adversarial Networks (GAN) [9, 13, 20] and Variational Autoencoders (VAE) [47]. Diffusion models transform a complex clean data distribution $p_{data}(x)$ into noise distribution $\mathcal{N}(0, I)$ and learn the reverse process to restore data from noise. Denoising Diffusion Implicit Models (DDIM) [43, 44] also iteratively solve the stochastic differential equation but with fewer time steps. Although DDIM shows impressive image generation results, the training process takes many time steps (e.g., $T = 1,000$) to converge. To this end, researchers have reconsidered the sampling process during training.

During training, the denoising starts from a low-quality image in the early stages. Current methods do not guide the network's reconstruction direction but instead rely on multiple iterations using loss to find the correct sampling direction. As a result, diffusion models often fail to find the correct reconstruction direction during early training, leading to reconstruction errors. Moreover, since iterative sampling methods are used, the network accumulates this reconstruction error [28]. To eliminate this error, the network requires many time steps, ultimately achieving convergence. Therefore, it is crucial to help the network find the reconstruction direction at the beginning.

Several works have addressed this issue based on inserting additional conditionals into the diffusion models. GDP [12] uses a protocol of conditional guidance, which enables the diffusion models to generate images with high quality. STF [54] combines the reference batch to reduce the covariance, such that they can accelerate the intermediate regime generation. QD [26] uses an additional PTQ tool to compresses the noise estimation network to accelerate the generation process. DA [36] proposes a semantic encoder to encode image features into the sampling process during training to accelerate the inference while generating images with high quality. Similarly, DiffuseVAE [35] injects VAE-generated images into the sampling process as additional conditions during training. P2 [7] prioritizes the later noise levels. An emphasis on the training of the reverse stage of these later noise levels encourages the model to reduce the accumulated error. Consequently, the convergence is sped up while better image quality is obtained.

In this paper, we depart from the idea of encoding additional conditionals into the sampling process during training. We propose a compensation algorithm for diffusion models that not only boosts the convergence during training without breaking any of the assumptions, but also improves the image quality. Our approach can be applied in various generation tasks, in both unconditional and conditional generation. Closest to our work is the recently introduced Cold Diffusion [2]. This work uses a Taylor expansion of the degradation function to represent the diffusion process, but it has to deal with diffusions that lack significant gradient information. While the method is beneficial for linear degradations, it fails to improve over the DDIM baseline because it violates the linear degradation assumption due to the addition of Gaussian noise. In contrast, our proposed approach has rigid mathematical underpinnings that adhere to the Gaussian assumption of the denoising process.

3 Compensation Sampling in Diffusion Models

We discuss common sampling practice in diffusion models before introducing our compensation sampling. We then discuss the architecture and training of the model with compensation sampling that is used in our experiments.

3.1 Common sampling in diffusion models

During training, diffusion models first degrade a complex clean data distribution $p_{data}(x)$ into noise distribution $\mathcal{N}(0, I)$ and learn the reverse process to reconstruct data from noise [15]. The diffusion process gradually corrupts clean data x_0 with predefined noise scales $0 < \beta_1 < \beta_2, ..., \beta_T < 1$, indexed by time step t $(1 \leq t \leq T)$, T the number of time steps. Corrupted data $x_1, ..., x_T$ are sampled from $x_0 \sim p_{data}(x)$ with a diffusion process, defined as the Gaussian transition:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t \mathrm{I}) \tag{1}$$

Here, q denotes the distribution of the noise image at time t. When a clean data point x_0 is provided, the noisy x_t can be obtained from the following equation, by adding noise for each time step:

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\overline{\alpha}_t}x_0, (1-\overline{\alpha}_t)\mathrm{I}) \tag{2}$$

where $\alpha_t = 1 - \beta_t$ and $\overline{\alpha}_t = \prod_{s=1}^{t} \alpha_s$. When t is close to T, x_t can be approximated as a Gaussian distribution. Diffusion models learn the reverse process to generate samples from the data distribution. The optimization objective of the reverse transition can be derived from a variational bound [22]. Ho et al. [15] employ a variational solution and assume its reverse transition kernel also subjects to a Gaussian distribution. This way, the generation process parameterizes the mean of the Gaussian transition distribution and fixes its variance as [24]:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t), \sigma_t^2 \mathrm{I}) \tag{3}$$

$$\mu_\theta(x_t) = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha}_t}}\epsilon_\theta^{(t)}(x_t)) \tag{4}$$

with $\epsilon_\theta^{(t)}$ is a function with trainable parameters $\theta(t)$ with variance σ_t^2 as a training hyper-parameter. The authors of DDIM [42] proposed an efficient sampling process that has the same training objectives as DDPM (which is described in Eq. 1-4), but with faster training. The main insight is instead of using Eq. 4 to obtain x_{t-1} from x_t, DDIM directly predicts the original clean data x_0 from x_t, and then use the predicted x_0 by Eq. 2 to generate x_{t-1}:

$$x_{t-1} = \sqrt{\overline{\alpha}_{t-1}} f_\theta(x_t, t) + \sqrt{1-\overline{\alpha}_{t-1}-\sigma_t^2}\epsilon_\theta^{(t)}(x_t) + \sigma_t^2 \mathrm{z} \tag{5}$$

where $f_\theta(x_t, t)$ is the prediction of clean data point x_0 when the noisy x_t is observed and a noise prediction model $\epsilon_\theta(x_t, t)$ is given:

$$f_\theta(x_t, t) = \frac{x_t - \sqrt{1-\overline{\alpha}_t}\epsilon_\theta^{(t)}(x_t)}{\sqrt{\overline{\alpha}_t}} \tag{6}$$

When variance σ_t is set to 0, the sampling process becomes deterministic. The non-Markovian diffusion process [42] allows the generation quality to remain unchanged with fewer denoising steps.

Although DDIM can accelerate the training process to some extent, however, since the prediction of x_0 is inaccurate at the start of training, convergence will be slow. In the conditional sampling algorithm, the solution is to iteratively predict the original data x_0 based on x_{t-1} [37, 52]:

$$p_\theta(x_{t-1}|x_t) = q(x_{t-1}|x_t, x_0 \approx \hat{x_0} = f_\theta(x_t, t)) \tag{7}$$

First, x_{t-1} is sampled using the posterior distribution in Eq. 7, and subsequently used as the input to predict the original data x_0. The above sampling process iteratively runs over t until the final result x_0 is generated.

3.2 Training limitation of previous common sampling

We first consider deterministic sampling where the noise pattern is selected prior to the generation process. The diffusion process is mathematically presented as:

$$D(x, t) = \sqrt{1 - \beta_t}x + \sqrt{\beta_t}z \tag{8}$$

with $D(x, t)$ the deterministic interpolation between data x and fixed noise pattern $z \in \mathcal{N}(0, I)$, at time step t.

To obtain the sample distribution $p_\theta(x_{t-1}|x_t)$ in Eq. 7, a neural network f_θ is typically used to predict the original data x_0, or noise ϵ. However, learning such f_θ is not easy, mainly because in the early stages of training, due to the poor initial quality of x_t, the quality of the learned $\hat{x_0}$ guided solely by a simple loss (e.g., Eq. 14) is also poor. This leads to a significant disparity between the sample distribution $p_\theta(x_{t-1}|x_t)$ learned through Eq. 7 (i.e., the distribution of x_{t-1}), and the actual ground truth $D(x, t-1)$, resulting in some reconstruction error in x_{t-1}. Furthermore, as x_{t-2} is obtained through Eq. 7 in the iterative denoising process based on the distribution of x_{t-1}, these errors will further accumulate. Subsequently, existing methods that rely solely on a simple loss function will lead to an extended convergence time and, eventually, to reduced image quality of the generated images [28].

3.3 Compensation sampling for accelerating training convergence

To prevent reconstruction error accumulation, we propose compensation sampling (CS). Based on Eq. 8, we define a general deterministic diffusion process:

$$D(x, t) = g(t)x + f(t)z \tag{9}$$

with $g(t)$ and $f(t)$ functions that define the noise scales, x is the input, $z \in \mathcal{N}(0, I)$. Considering the forward diffusion process, x_t is defined as:

$$x_t = g(t)x_0 + f(t)z \tag{10}$$

During the training of the denoising process, the compensation sampling algorithm relies on a learnable initial sample reconstruction model R :

$$\hat{x_0} = R(x_t, t) \tag{11}$$

With $\hat{x_0}$, we define the compensation weight $w(t)$ as the component to evaluate the noise scales difference at time step t:

$$w(t) = g(t) - g(t-1) \tag{12}$$

We then obtain the perfect inverse of x_{t-1} from x_t:

$$\begin{aligned} x_{t-1} &= x_t - D(\hat{x_0}, t) + D(\hat{x_0}, t-1) + w(t)(\hat{x_0} - x_0) \\ &= g(t)x_0 - g(t)\hat{x_0} + g(t-1)\hat{x_0} + f(t-1)z + w(t)(\hat{x_0} - x_0) \\ &= \hat{x_0}(g(t-1) - g(t)) + w(t)\hat{x_0} + g(t)x_0 - w(t)x_0 + f(t-1)z \\ &= g(t)x_0 - w(t)x_0 + f(t-1)z \\ &= g(t-1)x_0 + f(t-1)z \\ &= D(x_0, t-1) \end{aligned} \tag{13}$$

From Eqs. 9–12, it follows that $x_t = D(x_0, t)$ for all $t < T$, regardless of R during training, i.e., the generated x_{t-1} will be the same as the ground truth $D(x, t-1)$. This means that the accumulated error will be highly alleviated since the iterates x_t will be the same as when R is a perfect inverse for the degradation D. We notice that Cold Diffusion [2] also adds the term $D(\hat{x_0}, t) + D(\hat{x_0}, t-1)$ during sampling. However, only using this term has been shown to yield worse performance than DDIM [42]. Instead, our algorithm mathematically realizes the perfect inverse to reduce the accumulated error. We define $w(t)(\hat{x_0} - x_0)$ as the *compensation term*, which is based on the differences between $\hat{x_0}$ and x_0, as well as $w(t)$. So the compensation term can be seen as addressing the remaining shortcomings in reconstructing x_0 under the current time step t and the difference in noise scales $w(t)$. Our ablation studies provide insight into the practical operation of the term during training.

We use the final image x_0 in Eq. 12 but, during generation, we do not have access to ground truth x_0. Consequently, we cannot directly calculate the compensation term. To circumvent this issue, we use a lightweight U-Net model [38] during training as a compensation module to learn the compensation term. It is worth noting that we only utilize a single training epoch to train the compensation module to further save computation cost. There is significant diminishing return in further training of this module, which we avoid. We show in the supplementary material that further training will trade-off increased precision for decreased recall, but the FID will remain roughly the same. After one epoch of training, the compensation module indicates the direction of the denoising process, which is sufficient to guide x_t towards the original data distribution and to avoid error accumulation.

With Eqs. 9–12, compensation sampling realizes improved convergence of diffusion models during training. Importantly, our approach is not limited to fixed

noise scales $\beta_1, \beta_2, ..., \beta_T$ or fixed Gaussian noise patterns, but can be applied to any diffusion pattern. Also, since the compensation module is lightweight, it brings negligible computation cost during inference.

Algorithm 1: Training using compensation sampling

Input: $x_0, T \leftarrow$ sample image, total time step
Input: $E \leftarrow$ epochs to train compensation module
% Diffusion process
for $t = 1$ **to** T
    $x_t = D(x_0, t)$ ▷ Eqs. 9 & 10
end for
% Denoising process
for $t = T$ **to** 1 **step** -1
    $\hat{x}_0 = R(x_t, t)$ ▷ Eqs. 11 & 14
% Train compensation module
    for $e = 1$ **to** E
        CT = U-Net($\hat{x}_0$, t)
    end for
    $x_{t-1} = x_t - D(\hat{x}_0, t) + D(\hat{x}_0, t-1) + CT$ ▷ Eq. 13
    $x_t = x_{t-1}$
    $t = t - 1$
end for

3.4 Training and inference of compensation diffusion model

To make a fair comparison with recent diffusion models with common sampling, we opt to use the popular Ablated Diffusion Model (ADM) [9] as a backbone. ADM is based on the U-Net architecture [38] with residual blocks and self-attention layers in the low-resolution feature maps. In our main experiments, we use our compensation sampling in DDIM [42], and term the resulting model DDIM+CS. See architecture details, computation and parameter comparisons of the original U-Net and compensation module in the supplementary material.

The training process pipeline is provided in pseudocode form in Algorithm 1. To train ADM, we use the inner iteration training scheme [45], i.e., during a single training of ADM, the compensation module can be trained multiple times (default is one, see supplementary material for other options) using L_1 loss, which is a common loss function that measures the absolute difference between the predicted and actual compensation term. With ADM, we reduce the training time steps T to 100, which is a ten-fold reduction compared to the original ADM. Since the ADM can be interpreted as an equally weighted sequence of denoising modules $\epsilon_\theta(x_t, t)$, we train by optimizing loss function L_{DM} with respect to θ:

$$L_{DM} = \sum_{t=1}^{T} \mathbb{E}_{x_0, \epsilon_t} \left[\|\epsilon_\theta(x_t, t) - \epsilon_t\|_2^2 \right] \tag{14}$$

with x_0 the original image, and $\epsilon_t \in \mathbb{R}^{3\times H\times W} \sim \mathcal{N}(0, I)$.

During inference, the input is x_t, and the output is x_0. However, we do not require the compensation module during inference. This is evidenced by our investigation in the ablation study and by Figure 5 which reveal that, once the diffusion model has converged, the compensation term's value approaches zero. Consequently, our inference process aligns entirely with that of DDIM, both utilizing Eqs. 5 and 6 for sampling.

4 Experiments

We experiment on benchmark datasets CIFAR-10 [23], CelebA [29], CelebA-HQ [17], FFHQ [19], and FSG [6]. We address unconditional generation, conditional face inpainting, and face de-occlusion. We then analyze the training speed-up and present our ablation study. Additional results appear in the supplementary material.

4.1 Unconditional generation

Experiment setting. We evaluate on CelebA-64 (200k images, 64×64), FFHQ-256 (70k images, 256×256), and CIFAR-10 (60k images, 32×32 resolution). CelebA-64 and FFHQ-256 are frontal face datasets. CIFAR-10 contains images of 10 different classes such as airplane and cat. For fair comparison, we use the training hyper-parameters from DDIM [42]. We use Gaussian noise as our corruption mechanism and adopt the fixed linear variance schedule $\beta_1, ..., \beta_T$ as in previous works [15, 18, 34, 42] for the diffusion process in Eq. 1. Given the quicker convergence of compensation sampling, we reduce the training time steps T to 100. Both compensation and diffusion model use the Adam optimizer.

Evaluation metrics. How to quantitatively assess how accurately a generated distribution mimics the training data distribution remains an open research topic. We employ the widely used FID [14]. The FID score measures the KL divergence between two Gaussian distributions in the Inception-V3 feature space, computed by comparing real reference samples to generated samples. We randomly generate 50k images to compute the FID score (FID-50k) with the same implementation as in EDM [18]. Note that we treat the U-Net and the denoising network as separate network function evaluations (NFEs), as they are updated within one time step, we increase the NFE count by 2.

CelebA & FFHQ. We focus on face synthesis using diffusion models and GANs in Table 1. We compare the reported FID-50k performance from the published papers with the number of function evaluations (NFE) during generation. The top part of the table contains diffusion-based methods including our baselines Cold Diffusion [2] and DDIM [44] with common sampling, whereas the bottom part summarizes the performance of GANs.

We first discuss regular training, before moving to patch-based training. With regular training at NFE=1,000, our DDIM+CS achieves the best performance of all tested diffusion models. On CelebA, we outperform baselines DDIM with

Methods	Backbone \ NFE	CelebA-64					FFHQ-256		
		10	20	50	100	1,000	100	200	500
Cold Diffusion [2]	ADM	32.81	20.45	12.93	7.13	7.84	31.32	29.53	28.11
DDIM [42]	ADM	17.33	13.73	9.17	6.53	4.88	17.89	11.26	8.41
P2 Diffusion [7]	ADM	20.37	16.11	11.96	9.04	7.22	16.78	10.38	6.97
D2C [40]	NVAE+U-Net	17.32	11.46	6.80	5.70	5.15	13.04	9.85	7.94
Diffusion Autoencoder [36]	ADM	12.92	10.18	7.05	5.30	4.97	15.33	8.80	5.81
Analytic-DDIM [4]	ADM	15.62	10.45	6.13	4.29	3.13	14.84	9.02	5.98
DPM-Solver [30]	ADM	5.83	3.13	3.10	3.11	3.08	10.82	8.47	8.40
SN-DDIM [3]	ADM	10.20	6.77	3.83	3.04	2.90	14.82	8.79	5.44
F-PNDM [27]	ADM	7.71	5.51	3.34	2.81	2.86	17.51	8.23	4.79
DDIM+CS (**ours**)	ADM	7.80	5.11	2.23	2.11	1.98	11.89	7.31	4.02
DPM-Solver+CS (**ours**) [30]	ADM	<u>5.22</u>	<u>2.35</u>	2.22	2.12	2.04	<u>9.73</u>	<u>4.66</u>	4.01
PDM [49] (patch-wise training)	ADM	35.88	8.36	<u>1.77</u>	<u>1.86</u>	<u>1.82</u>	23.55	6.47	<u>3.13</u>
PDM+CS (**ours**) (patch-wise training)	ADM	**4.93**	**1.97**	**1.42**	**1.44**	**1.38**	**6.11**	**3.52**	**2.57**
U-Net GAN [39]	U-Net	19.31					10.90		
VQGAN [11]	CNN+Transformer	12.70					9.60		
GANFormer2 [1]	CNN+Transformer	6.87					7.77		
Diffusion StyleGAN2 [50]	StyleGAN	<u>1.69</u>					<u>3.73</u>		
Diffusion StyleGAN2+CS (**ours**) [50]	StyleGAN	**1.21**					**2.95**		

Table 1: Unconditional image generation results. Comparison on FID-50k with the state-of-the-art on CelebA-64 and FFHQ-256. Top part of the table contains baselines DDIM and Cold Diffusion, the middle part are state-of-the-art diffusion models, and the bottom part summarizes the performance of GAN models. Best results for diffusion and GAN models in **bold**, second best are <u>underlined</u>.

common sampling and Cold Diffusion by 59% (4.88→1.98) and 75% (7.84→1.98). For FFHQ-256 at NFE=500, gains of 52% and 86% are achieved. More importantly, all tested models at NFE=1,000 are also outperformed by DDIM+CS with NFE=100 or even NFE=50. This indicates that we obtain higher quality images with much lower training time. We analyze this in Section 4.4.

We also obtain better performance at various NFEs compared with other approaches. For example, when we apply our compensation sampling in DPM-Solver (DPM-Solver+CS), we obtain improved performance on both CelebA-64 and FFHQ-256.

Patch-wise training as used in PDM [49] allows for even better results owing to the more performant generation scheme. Our resulting PDM+CS obtains state-of-the-art performance on both CelebA-64 (FID-50k=1.38 at NFE=1,000) and FFHQ-256 (FID-50k=2.57 at NFE=500).

For GANs, the superior performance of Diffusion StyleGAN2 can be attributed to the powerful backbone. When applying our compensation diffusion method to this backbone (Diffusion StyleGAN2+CS), we obtain the best performance with an improvement of 28% (1.69→1.21) on CelebA-64 and 21% (3.73→2.95) on FFHQ-256.

CIFAR-10. To validate our approach on general image synthesis, we report on CIFAR-10 in Table 2. We evaluate with NFE=35, in line with EDM [18] and PFGM++ [53]. Compared to approaches trained with 10 times more time steps, our DDIM+CS achieves competitive results with much less training time (see Table 5). When trained with $T = 1,000$, DDIM+CS (FID-50k=1.57) outper-

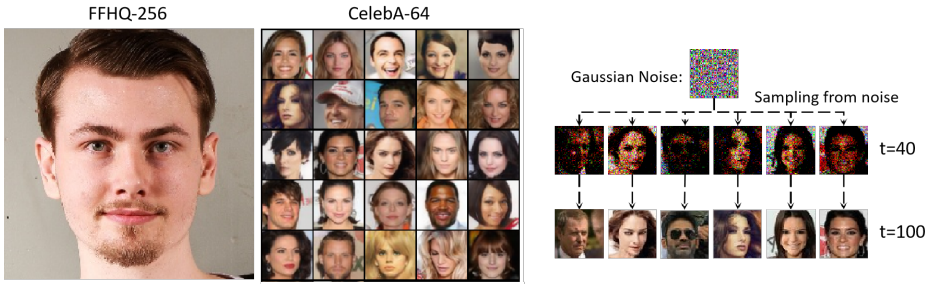

Fig. 2: Visualization of our compensation diffusion results. (left) The unconditional generation results of FFHQ-256 (256×256 resolution) and CelebA-64 (64×64 resolution) datasets (NFE = 100). (right) Illustration of the diversity of our compensation sampling on random Gaussian noise, visualized at time steps 40 and 100.

forms the current state-of-the-art. We observe that the previous best approach PFGM++ [53] has a significantly more complex backbone, as it combines diffusion models with Poisson Flow Generative Models. When we apply compensation sampling in PFGM++ (PFGM++ + CS), a lower FID score is achieved (1.91→1.74) with the same NFE, but 10 times fewer time steps. When we increase the time steps to $T = 1,000$, we obtain the state-of-the-art result of 1.50.

Qualitative evaluation. Images generated by DDIM with compensation sampling for CelebA-64 and FFHQ-256 appear in Figure 2 (left). Images generated for CIFAR-10 are shown in the supplementary material. For different resolutions, our method generates images with realistic details. We show the stochastic process with a wide diversity in outputs in Figure 2 (right). We present more examples, including failure cases, in the supplementary material.

Method	**FID-50k ↓**
DDPM (T=1,000, NFE=1,000) [15]	3.17
DDIM (T=1,000, NFE=1,000) [42]	3.95
DPM-Solver (T=1,000, NFE = 44) [30]	3.48
DiffuseVAE-72M (T=1,000, NFE=1,000) [35]	2.62
DDPM++ (T=1,000, NFE=1,000) [21]	2.56
LSGM (T=1,000, NFE=138) [48]	2.10
EDM (T=1,000, NFE=35) [18]	1.97
PFGM++ (T=1,000, NFE=35) [53]	1.91
DPM-Solver+CS (**ours**) (T=100, NFE = 35)	1.93
DDIM+CS (**ours**) (T=100, NFE=35)	2.01
DDIM+CS (**ours**) (T=1,000, NFE=35)	<u>1.57</u>
PFGM++ + CS (**ours**) (T=100, NFE=35)	1.74
PFGM++ + CS (**ours**) (T=1,000, NFE=35)	**1.50**

Table 2: Unconditional image generation results on **CIFAR-10**. Best result in **bold**.

Method	**PSNR ↑**	**SSIM ↑**
Occluded image	10.4764	0.6425
CycleGAN [60]	13.7667	0.6459
DeepFill [56]	14.5140	0.7029
OA-GAN [10]	17.1828	0.7215
FSG-GAN [6]	21.1112	0.7936
Cascade GAN [57]	26.4736	0.8422
SRNet [55]	27.0031	0.8493
DDIM	19.3211	0.7308
DDIM+CS (**ours**)	**31.3842**	**0.8699**

Table 3: Face de-occlusion on FSG. Best results in **bold**.

Methods	Half		Completion		Expand		Thick Line		Medium Line	
	LPIPS ↓	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓	FID ↓
CoModGAN [59]	0.445	37.72	0.406	43.77	0.671	93.48	0.091	5.82	0.105	5.86
LaMa [46]	0.342	33.82	0.315	25.72	0.538	86.21	0.080	5.47	0.077	5.18
CDE [5]	0.344	29.33	0.302	19.07	0.508	71.99	0.079	4.77	0.070	4.33
RePaint [32]	0.435	41.28	0.387	37.96	0.665	92.03	**0.059**	5.08	**0.028**	4.97
MAT [25]	0.331	32.55	0.280	20.63	0.479	82.37	0.080	5.16	0.077	4.95
GLaMa [31]	0.327	30.76	0.289	18.61	0.481	80.44	0.081	5.83	0.080	5.10
FcF [16]	0.305	27.95	0.378	31.91	0.502	73.24	0.086	4.63	0.071	4.42
DDIM	0.377	33.25	0.328	29.30	0.446	57.93	0.088	8.44	0.112	9.02
DDIM+CS (**ours**)	**0.272**	**20.37**	**0.259**	**15.33**	**0.372**	**39.05**	0.079	**4.21**	0.064	**3.58**

Table 4: Results of face inpainting on CelebA-HQ-256. DDIM with our compensation sampling shows consistent improvement over state-of-the-art methods, for both LPIPS and FID-50k metrics. Best results in **bold**.

4.2 Face inpainting

The goal of face inpainting is to restore missing or damaged parts in a face image, resulting in a complete facial image. It is a conditional image generation task.

Experiment setting. We perform the inpainting experiment with DDIM+CS on CelebA-HQ-256 with NFE = 100 and 40 training epochs for the compensation module to reduce the output diversity. Results with different numbers of epochs for the compensation module appear in the supplementary material. All other training hyper-parameters are the same as for the unconditional generation experiment. We employ the same schedule of corruption transforms as in previous works [5, 32]. Each training image of the reconstruction model is corrupted with a synthetically generated mask. The superimposition process starts with input images x_0, which are iteratively masked for T steps via multiplication with a sequence of masks $m \in [0, 1]$. We follow [31, 32] and use five mask types: *Half*, *Completion*, *Expand*, *Thick Line*, and *Medium Line*, see Figure 3.

Evaluation metrics. Following recent image inpainting literature, we use Learned Perceptual Image Patch Similarity (LPIPS) [58] and FID as similarity metrics. Compared to PSNR and SSIM [51], LPIPS and FID are more suited to measure the performance of inpainting for large masks [31].

Quantitative evaluation. Results of diffusion and GAN-based approaches appear in Table 4. We report results from the papers, and use open source implementations to calculate missing numbers. DDIM+CS outperforms other approaches in almost all cases. Despite slightly worse LPIPS scores than RePaint for the Thick Line and Medium Line masks (0.059→0.079, 0.028→0.064), we show better performance for other masks. Moreover, we consistently outperform all tested methods on FID-50k score. Models trained with Expand masks give worse results than other mask types, which is consistent with observations in GLaMa [31]. Still, our approach significantly reduces both LPIPS and FID scores.

Qualitative evaluation. Inpainting results appear in Figure 3 and the supplementary material. Compared to state-of-the-art approaches LaMa [46] and GLaMa [31], our generated images are more detailed and have fewer flaws.

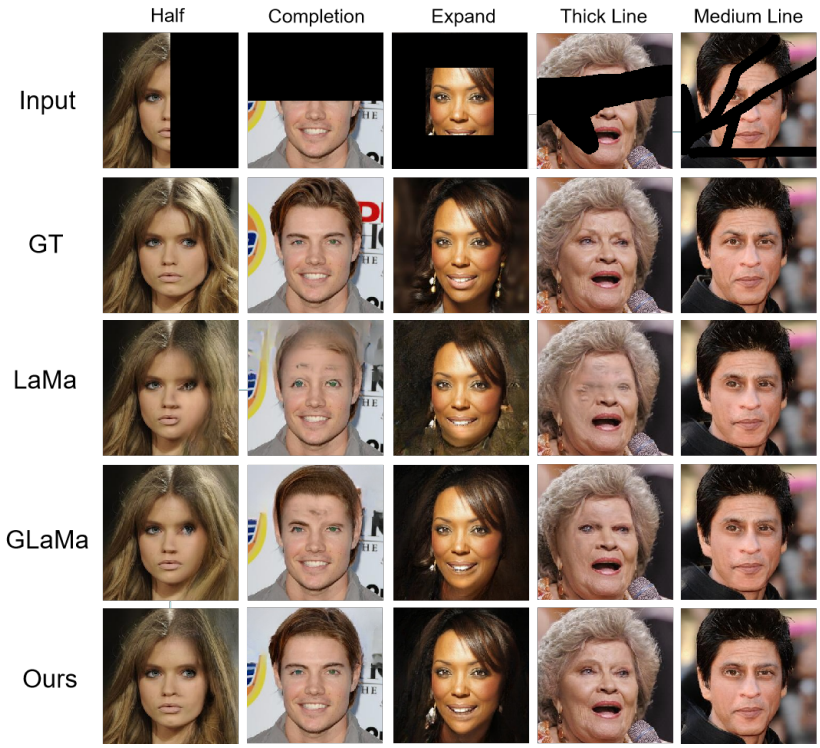

Fig. 3: Visualization of inpainting results on CelebA-HQ-256. Compared to state-of-the-art methods, our method generates more realistic images.

4.3 Face de-occlusion

We further experiment with face de-occlusion. Compared to face inpainting, face de-occlusion is more challenging since the missing contents of the face images are not black pixels but pixels with values potentially similar to faces. During the reconstruction process, the models therefore not only have to infer which parts belong to the face, but also fill in the content of the missing parts.

Experiment setting. We train and evaluate our DDIM+CS using FSG (200k images) [6]. The face images are synthesized with common occlusion objects that are semantically placed relative to facial landmarks. We use the same settings as in the face inpainting experiments.

Evaluation metrics. While face de-occlusion has been addressed with GANs, we believe ours is the first work to apply diffusion models to face de-occlusion without any additional information such as a segmentation mask. To compare to other works, we use PSNR and SSIM as metrics.

We summarize the results in Table 3. DDIM with compensation sampling shows significantly better performance than any of the tested GANs. For exam-

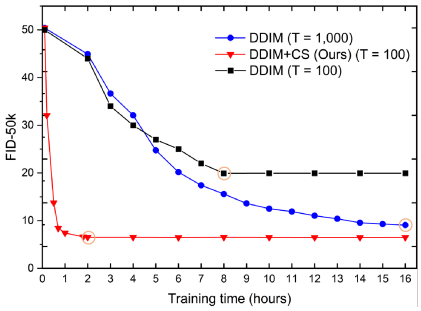

Fig. 4: Training time for unconditional face generation on CelebA-64. DDIM and DDIM+CS are trained with 1,000 and 100 time steps, respectively.

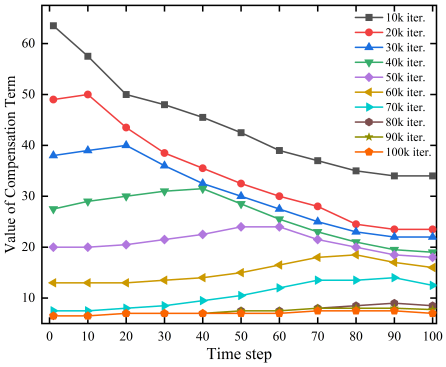

Fig. 5: Compensation term value per training iteration, unconditional face generation on FFHQ-256, DDIM+CS $T = 100$.

ple, DDIM+CS achieves 20% lower PSNR compared to the best tested SRNet. More examples are included in the supplementary material.

4.4 Training computation cost

We consistently observe competitive or better results when using compensation sampling with 10 times fewer time steps. Here, we investigate the benefit of our algorithm in accelerating the training process. Figure 4 clearly illustrates the value of our contribution. It shows FID-50k scores during the DDIM training process with common and compensation sampling for unconditional generation on CelebA-64. DDIM+CS ($T = 100$) takes only 2 hours to converge to a FID-50k of 2.11, while DDIM ($T = 1,000$) takes 16 hours to achieve 4.88.

Method	Property	CIFAR-10	CelebA-64	FFHQ-256
DDIM ($T = 1,000$)	GFLOPS	7.76	15.52	248.17
	Total time	7h	16h	26h
	FID-50k	3.95	4.88	8.41
DDIM+CS (**ours**) ($T = 100$)	GFLOPS	7.78	15.58	249.07
	Total time	0.4h	2.0h	5.3h
	FID-50k	2.01	2.11	4.02

Table 5: Training time for DDIM and ours on different datasets/resolutions.

Method	FID-50k
DDIM	17.89
Cold Diffusion	31.32
DDIM+CS	11.89
DDIM+CS (Train)	11.84

Table 6: Effects of compensation term.

In Table 5, we report training times and FLOPs for three datasets with different resolutions. DDIM+CS is consistently much faster owing to the fewer required time steps. The limited increase in FLOPs demonstrates that the training of the compensation module has a negligible effect on the computation cost.

4.5 Ablation study

Compensation term value during training. We study how the compensation term varies during training for unconditional generation on FFHQ-256 of DDIM+CS. In Figure 5, we show the average compensation term value over all training images every 10 time steps with $T = 100$ steps. We make the following conclusions. First, the value of the compensation term continuously decreases at each time step. The accumulated error is higher in early training, and more compensation is needed. Second, the highest value gradually moves from time step 1 to 100 as training progresses. The compensation initially aims at recovering fine details, before focusing on the more difficult task of generating structure [7].

Effect of compensation term during training and generation, on unconditional generation on FFHQ-256, $T = 100$. We compare with common sampling (DDIM [44]) and Cold Diffusion [2]. For DDIM+CS, we investigate the setting in which the compensation term is used during training but not generation (DDIM+CS (Train)), and the setting where it is used in both (DDIM+CS).

From Table 6, we observe the performance improvement when using compensation sampling. This difference is mainly due its use during training. When also using the compensation term in the generation phase, the FID-50k score slightly improves (11.89→ 11.84). The worse performance of Cold Diffusion might be caused by incorrect assumptions about the noise distribution. These results demonstrate that the compensation term is beneficial in producing higher-quality outputs. However, the added value of the compensation term in inference is minimal. This makes sense, since its value decreases as the training progresses, see Figure 5. Using a small residual term during inference has a limited effect.

5 Discussion and Conclusions

We have introduced compensation sampling, a novel algorithm to guide the training of diffusion models. The main limitation is that our method is currently applied on score-based linear generative models, in the future, we will deploy our method into nonlinear diffusion models. Also, our reverse process is deterministic rather than stochastic. This may affect the quality of generated images, because deterministic processes may not capture all the variability of the target distribution. However, our compensation term is freely learnable by the network, which introduces some variability in the target distribution during this process.

Our innovation has three main benefits. First, training diffusion models with our approach converge to a better solution due to the reduction of accumulated error. Second, by guiding the convergence, we can reduce the number of time steps up to an order of magnitude. Finally, our approach is no free lunch, but addresses error accumulation with negligible computational cost in linear diffusion models, the most common type of diffusion model. We show this on unconditional face generation, face inpainting, and face de-occlusion. Our results on benchmark datasets consistently demonstrate superior performance and increased efficiency compared to state-of-the-art diffusion and GAN models. Our sampling approach is general and can be used in a wide range of diffusion models.

References

1. Arad Hudson, D., Zitnick, L.: Compositional transformers for scene generation. Advances in Neural Information Processing Systems **34**, 9506–9520 (2021)
2. Bansal, A., Borgnia, E., Chu, H.M., Li, J., Kazemi, H., Huang, F., Goldblum, M., Geiping, J., Goldstein, T.: Cold diffusion: Inverting arbitrary image transforms without noise. Advances in Neural Information Processing Systems **36** (2024)
3. Bao, F., Li, C., Sun, J., Zhu, J., Zhang, B.: Estimating the optimal covariance with imperfect mean in diffusion probabilistic models. In: International Conference on Machine Learning. pp. 1555–1584. PMLR (2022)
4. Bao, F., Li, C., Zhu, J., Zhang, B.: Analytic-DPM: An analytic estimate of the optimal reverse variance in diffusion probabilistic models. In: International Conference on Learning Representations (2022)
5. Batzolis, G., Stanczuk, J., Schönlieb, C.B., Etmann, C.: Conditional image generation with score-based diffusion models. arXiv preprint arXiv:2111.13606 (2021)
6. Cheung, Y.M., Li, M., Zou, R.: Facial structure guided GAN for identity-preserved face image de-occlusion. In: International Conference on Multimedia Retrieval. pp. 46–54 (2021)
7. Choi, J., Lee, J., Shin, C., Kim, S., Kim, H., Yoon, S.: Perception prioritized training of diffusion models. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11472–11481 (2022)
8. Daras, G., Delbracio, M., Talebi, H., Dimakis, A., Milanfar, P.: Soft Diffusion: Score matching with general corruptions. Transactions on Machine Learning Research (TMLR) (2023)
9. Dhariwal, P., Nichol, A.: Diffusion models beat GANs on image synthesis. Advances in Neural Information Processing Systems **34**, 8780–8794 (2021)
10. Dong, J., Zhang, L., Zhang, H., Liu, W.: Occlusion-aware GAN for face de-occlusion in the wild. In: IEEE International Conference on Multimedia and Expo (ICME). pp. 1–6 (2020)
11. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12873–12883 (2021)
12. Fei, B., Lyu, Z., Pan, L., Zhang, J., Yang, W., Luo, T., Zhang, B., Dai, B.: Generative diffusion prior for unified image restoration and enhancement. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9935–9946 (2023)
13. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems. vol. 27 (2014)
14. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local Nash equilibrium. Advances in neural information processing systems **30** (2017)
15. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in Neural Information Processing Systems **33**, 6840–6851 (2020)
16. Jain, J., Zhou, Y., Yu, N., Shi, H.: Keys to better image inpainting: Structure and texture go hand in hand. In: IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 208–217 (2023)
17. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of GANs for improved quality, stability, and variation. In: International Conference on Learning Representations (2018)

18. Karras, T., Aittala, M., Aila, T., Laine, S.: Elucidating the design space of diffusion-based generative models. arXiv preprint arXiv:2206.00364 (2022)
19. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4401–4410 (2019)
20. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of StyleGAN. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8110–8119 (2020)
21. Kim, D., Shin, S., Song, K., Kang, W., Moon, I.C.: Soft truncation: A universal training technique of score-based diffusion model for high precision score estimation. In: International Conference on Machine Learning. pp. 11201–11228. PMLR (2022)
22. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
23. Krizhevsky, A.: Learning multiple layers of features from tiny images. Tech. rep., University of Toronto (2009)
24. Li, S., Zheng, G., Wang, H., Yao, T., Chen, Y., Ding, S., Li, X.: Entropy-driven sampling and training scheme for conditional diffusion generation. arXiv preprint arXiv:2206.11474 (2022)
25. Li, W., Lin, Z., Zhou, K., Qi, L., Wang, Y., Jia, J.: MAT: Mask-aware transformer for large hole image inpainting. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10758–10768 (2022)
26. Li, X., Liu, Y., Lian, L., Yang, H., Dong, Z., Kang, D., Zhang, S., Keutzer, K.: Q-diffusion: Quantizing diffusion models. In: IEEE/CVF International Conference on Computer Vision. pp. 17535–17545 (2023)
27. Liu, L., Ren, Y., Lin, Z., Zhao, Z.: Pseudo numerical methods for diffusion models on manifolds. arXiv preprint arXiv:2202.09778 (2022)
28. Liu, X., Gong, C., Liu, Q.: Flow straight and fast: Learning to generate and transfer data with rectified flow. arXiv preprint arXiv:2209.03003 (2022)
29. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: IEEE International Conference on Computer Vision. pp. 3730–3738 (2015)
30. Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., Zhu, J.: DPM-Solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. Advances in Neural Information Processing Systems **35**, 5775–5787 (2022)
31. Lu, Z., Jiang, J., Huang, J., Wu, G., Liu, X.: GLaMa: Joint spatial and frequency loss for general image inpainting. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1301–1310 (2022)
32. Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., Van Gool, L.: Repaint: Inpainting using denoising diffusion probabilistic models. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11461–11471 (2022)
33. Moghadam, P.A., Van Dalen, S., Martin, K.C., Lennerz, J., Yip, S., Farahani, H., Bashashati, A.: A morphology focused diffusion probabilistic model for synthesis of histopathology images. In: IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2000–2009 (2023)
34. Nash, C., Menick, J., Dieleman, S., Battaglia, P.W.: Generating images with sparse representations. arXiv preprint arXiv:2103.03841 (2021)
35. Pandey, K., Mukherjee, A., Rai, P., Kumar, A.: DiffuseVAE: Efficient, controllable and high-fidelity generation from low-dimensional latents. arXiv preprint arXiv:2201.00308 (2022)

36. Preechakul, K., Chatthee, N., Wizadwongsa, S., Suwajanakorn, S.: Diffusion autoencoders: Toward a meaningful and decodable representation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10619–10629 (2022)
37. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 (2022)
38. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. pp. 234–241 (2015)
39. Schonfeld, E., Schiele, B., Khoreva, A.: A U-net based discriminator for generative adversarial networks. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8207–8216 (2020)
40. Sinha, A., Song, J., Meng, C., Ermon, S.: D2C: Diffusion-decoding models for few-shot conditional generation. Advances in Neural Information Processing Systems **34**, 12533–12548 (2021)
41. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: International Conference on Machine Learning. pp. 2256–2265 (2015)
42. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: International Conference on Learning Representations (2021)
43. Song, Y., Ermon, S.: Generative modeling by estimating gradients of the data distribution. Advances in Neural Information Processing Systems **32** (2019)
44. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456 (2020)
45. Such, F.P., Rawal, A., Lehman, J., Stanley, K., Clune, J.: Generative teaching networks: Accelerating neural architecture search by learning to generate synthetic training data. In: International Conference on Machine Learning. pp. 9206–9216 (2020)
46. Suvorov, R., Logacheva, E., Mashikhin, A., Remizova, A., Ashukha, A., Silvestrov, A., Kong, N., Goka, H., Park, K., Lempitsky, V.: Resolution-robust large mask inpainting with Fourier convolutions. In: IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2149–2159 (2022)
47. Vahdat, A., Kautz, J.: NVAE: A deep hierarchical variational autoencoder. Advances in Neural Information Processing Systems **33**, 19667–19679 (2020)
48. Vahdat, A., Kreis, K., Kautz, J.: Score-based generative modeling in latent space. Advances in Neural Information Processing Systems **34**, 11287–11302 (2021)
49. Wang, Z., Jiang, Y., Zheng, H., Wang, P., He, P., Wang, Z., Chen, W., Zhou, M.: Patch diffusion: Faster and more data-efficient training of diffusion models. arXiv preprint arXiv:2304.12526 (2023)
50. Wang, Z., Zheng, H., He, P., Chen, W., Zhou, M.: Diffusion-GAN: Training GANs with diffusion. In: International Conference on Learning Representations (2023)
51. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing **13**(4), 600–612 (2004)
52. Xiao, Z., Kreis, K., Vahdat, A.: Tackling the generative learning trilemma with denoising diffusion GANs. arXiv preprint arXiv:2112.07804 (2021)
53. Xu, Y., Liu, Z., Tian, Y., Tong, S., Tegmark, M., Jaakkola, T.: PFGM++: Unlocking the potential of physics-inspired generative models. arXiv preprint arXiv:2302.04265 (2023)

54. Xu, Y., Tong, S., Jaakkola, T.S.: Stable target field for reduced variance score estimation in diffusion models. In: International Conference on Learning Representations (2023)
55. Yin, X., Huang, D., Fu, Z., Wang, Y., Chen, L.: Segmentation-reconstruction-guided facial image de-occlusion. In: IEEE International Conference on Automatic Face and Gesture Recognition (FG). pp. 1–8 (2023)
56. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Generative image inpainting with contextual attention. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 5505–5514 (2018)
57. Zhang, N., Liu, N., Han, J., Wan, K., Shao, L.: Face de-occlusion with deep cascade guidance learning. IEEE Transactions on Multimedia (2022)
58. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 586–595 (2018)
59. Zhao, S., Cui, J., Sheng, Y., Dong, Y., Liang, X., Chang, E.I., Xu, Y.: Large scale image completion via co-modulated generative adversarial networks. arXiv preprint arXiv:2103.10428 (2021)
60. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: IEEE International Conference on Computer Vision. pp. 2223–2232 (2017)