DreamSampler: Unifying Diffusion Sampling and Score Distillation for Image Manipulation

Jeongsol Kim[1,*], Geon Yeong Park[1,*], and Jong Chul Ye[2]

[1] Dept. of Bio & Brain Engineering, KAIST
[2] Kim Jae Chul AI graduate school, KAIST
{jeongsol, pky3436, jong.ye}@kaist.ac.kr
* Equal contribution

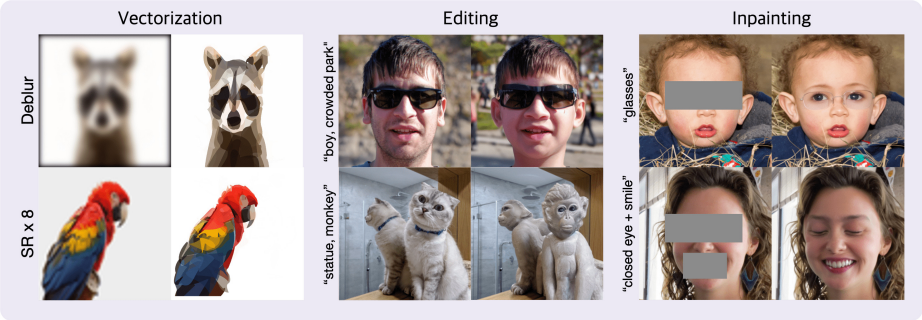

Fig. 1: DreamSampler can be used for vectorized image restoration, editing, text-guided inpainting, etc. Code: https://github.com/DreamSampler/dream-sampler

Abstract. Reverse sampling and score-distillation have emerged as main workhorses in recent years for image manipulation using latent diffusion models (LDMs). While reverse diffusion sampling often requires adjustments of LDM architecture or feature engineering, score distillation offers a simple yet powerful model-agnostic approach, but it is often prone to mode-collapsing. To address these limitations and leverage the strengths of both approaches, here we introduce a novel framework called *DreamSampler*, which seamlessly integrates these two distinct approaches through the lens of regularized latent optimization. Similar to score-distillation, DreamSampler is a model-agnostic approach applicable to any LDM architecture, but it allows both distillation and reverse sampling with additional guidance for image editing and reconstruction. Through experiments involving image editing, SVG reconstruction and etc, we demonstrate the competitive performance of DreamSampler compared to existing approaches, while providing new applications.

Keywords: Latent diffusion model · Generation · Score distillation

1 Introduction

Diffusion models [6, 26, 27] have been extensively studied as powerful generative models in recent years. These models operate by generating clean images

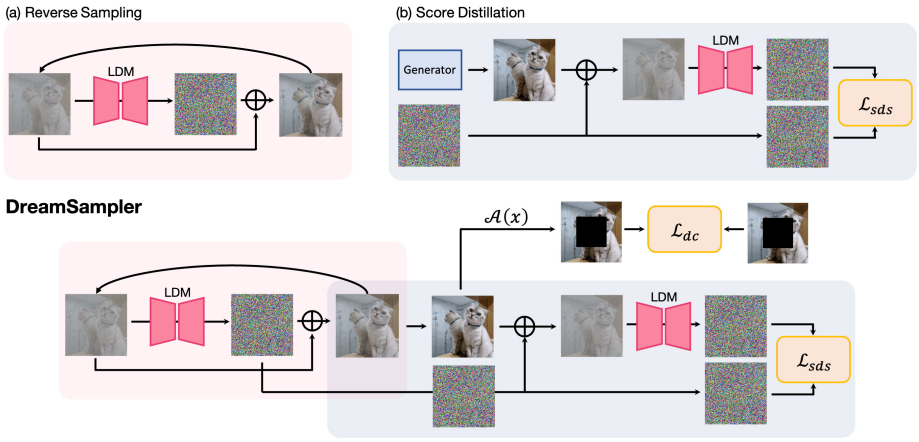

Fig. 2: DreamSampler vs (a) reverse diffusion and (b) score distillation.

from Gaussian noise through a process termed ancestral sampling. This involves progressively reducing noise by utilizing an estimated score function to guide the generation process from a random starting point towards the distribution of natural images. Reverse diffusion sampling introduces stochasticity through the reverse Wiener process within the framework of SDE [27], contributing to the prevention of mode collapse in generated samples while enhancing the fidelity [3]. Furthermore, the reverse diffusion can be flexibly regularized by various guidance gradients. For instance, classifier-guidance [3] applies classifier gradients to intermediate noisy samples, facilitating conditional image generation. Additionally, DPS [1] utilizes approximated likelihood gradients to constrain the sampling process, ensuring data consistency between the current estimated solution and given observation, thereby solving noisy inverse problems in a zero-shot manner.

On the other hand, another type of approach, called *score distillation*, utilizes diffusion model as prior knowledge for image generation and editing. For example, DreamFusion [21] leveraged 2D text-conditioned diffusion models for text-guided 3D representation learning via NeRF. Here, the diffusion model serves as the teacher model, generating gradients by comparing its predictions with the label noises and guiding the generator as the student model. The strength of the score distillation method lies in its ability to leverage the pre-trained diffusion model in a black-box manner without requiring any feature engineering, such as adjustments to the model architecture. Unfortunately, the score distillation is more often prone to mode collapsing compared to the reverse diffusion.

Although both algorithms are grounded in the same principle of diffusion models, the approaches appear to be different, making it unclearly how to synergistically combine the two approaches. To address this, we introduce a unified framework called *DreamSampler*, that seamlessly integrates two distinct approaches and take advantage of the both worlds through the lens of regularized latent optimization. Specifically, DreamSampler is model-agnostic and does not

require any feature engineering, such as adjustments to the model architecture. Moreover, in contrast to the score distillation, DreamSampler is free of mode collapsing thanks to the stochastic nature of the sampling.

The pioneering aspect of DreamSampler is rooted in two pivotal insights. First, we demonstrate that the process of latent optimization during reverse diffusion can be viewed as a proximal update from the posterior mean by Tweedie's formula. This interpretation allows us to integrate additional regularization terms, such as measurement consistency in inverse problems, to steer the sampling procedure. Moreover, we illustrate that the loss associated with the proximal update can be conceptualized as the score distillation loss. This insight bridges a natural connection between the score-distillation methodology and reverse sampling strategies, culminating in their harmonious unification. In subsequent sections, we will explore various applications emerging from this integrated framework and demonstrate the efficacy of DreamSampler through empirical evidence.

2 Motivations

2.1 Preliminaries

In LDMs [23], the encoder $\mathcal{E}_\phi$ and the decoder $\mathcal{D}_\varphi$ are trained as auto-encoder, satisfying $\boldsymbol{x} = \mathcal{D}_\varphi(\mathcal{E}_\phi(\boldsymbol{x})) = \mathcal{D}_\varphi(\boldsymbol{z}_0)$ where $\boldsymbol{x}$ denotes clean image and $\boldsymbol{z}_0$ denotes encoded latent vector. Then, the diffusion process is defined on the latent space, which is a range space of $\mathcal{E}_\phi$. Specifically, the forward diffusion process is

$$\boldsymbol{z}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{z}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon} \tag{1}$$

where $\bar{\alpha}_t$ denotes pre-defined coefficient that manages noise scheduling, and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$ denotes a noise sampled from normal distribution. The reverse diffusion process requires a score function via a neural network (i.e. diffusion model, $\boldsymbol{\epsilon}_\theta$) trained by denoising score matching [6, 27]:

$$\min_\theta \mathbb{E}_{t,\boldsymbol{\epsilon}\sim\mathcal{N}(0,\boldsymbol{I})}\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, t)\|_2^2. \tag{2}$$

According to formulation of DDIM [2,26], the reverse sampling from the posterior distribution $p(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t, \boldsymbol{z}_0)$ could be described as

$$\boldsymbol{z}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\hat{\boldsymbol{z}}_{0|t} + \sqrt{1-\bar{\alpha}_{t-1}}\tilde{\boldsymbol{\epsilon}} \tag{3}$$

where

$$\hat{\boldsymbol{z}}_{0|t} = (\boldsymbol{z}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, t))/\sqrt{\bar{\alpha}_t} \tag{4}$$

$$\tilde{\boldsymbol{\epsilon}} = \frac{\sqrt{1-\bar{\alpha}_{t-1}-\eta^2\beta_t^2}\boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, t)}{\sqrt{1-\bar{\alpha}_{t-1}}} + \frac{\eta\beta_t\boldsymbol{\epsilon}}{\sqrt{1-\bar{\alpha}_{t-1}}}. \tag{5}$$

Here, $\hat{\boldsymbol{z}}_{0|t}$ refers to the denoised latent through Tweedie's formula, and $\tilde{\boldsymbol{\epsilon}}$ is the noise term composed of both deterministic $\boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, t)$ and stochastic term

$\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$. Note that η and β_t denote variables that controls the stochastic property of sampling. When $\eta\beta_t = 0$, the sampling is deterministic.

For text conditioning, classifier-free-guidance (CFG) [7] is widely leveraged. The estimated noise is computed by

$$\boldsymbol{\epsilon}^{\omega}_{\theta}(\boldsymbol{z}_t, t, c_{ref}) = \boldsymbol{\epsilon}_{\theta}(\boldsymbol{z}_t, t, c_{\varnothing}) + \omega[\boldsymbol{\epsilon}_{\theta}(\boldsymbol{z}_t, t, c_{ref}) - \boldsymbol{\epsilon}_{\theta}(\boldsymbol{z}_t, t, c_{\varnothing})] \tag{6}$$

where ω denotes the guidance scale, $c_{\varnothing}$ refers to the null-text embedding, and c_{ref} is the conditioning text embedding, which are encoded by pre-trained text encoder such as CLIP [22]. For simplicity, we will interchangeably use the terms $\boldsymbol{\epsilon}_{\theta}(\boldsymbol{z}_t)$, $\boldsymbol{\epsilon}_{\theta}(\boldsymbol{z}_t, t)$ and $\boldsymbol{\epsilon}^{\omega}_{\theta}(\boldsymbol{z}_t, t, c_{ref})$ unless stated otherwise.

2.2 Key Observations

Score distillation sampling (SDS) [21] and reverse diffusion [6, 26, 27] represent two distinct methodologies, each with its own pros and cons. SDS is an optimization method that focuses on minimizing score distillation loss, while reverse diffusion utilizes ancestral sampling, which stems from SDE or DDPM formulations. Although SDS is straightforward and model-agnostic, it often suffers from mode collapse due to its non-stochastic nature. Conversely, ancestral sampling typically avoids mode collapsing and generates more diverse outputs, but it is non-trivial to generalize the ancestral sampling with generic parameter space, e.g. NeRF MLP. The primary contribution of our study is the integration of these two approaches through an optimization perspective based on following key observations. The first key insight of DreamSampler is that the DDIM sampling can be interpreted as the solution of following optimization problem:

$$\boldsymbol{z}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\bar{\boldsymbol{z}} + \sqrt{1-\bar{\alpha}_{t-1}}\bar{\boldsymbol{\epsilon}}, \quad \text{where} \quad \bar{\boldsymbol{z}} = \arg\min_{\boldsymbol{z}} \|\boldsymbol{z} - \hat{\boldsymbol{z}}_{0|t}\|^2 \tag{7}$$

Although looks trivial, one of the important implications of (7) is that we can now extend the solution of (7) to include additional regularization term,

$$\min_{\boldsymbol{z}} \|\boldsymbol{z} - \hat{\boldsymbol{z}}_{0|t}\|_2^2 + \lambda_{reg}\mathcal{R}(\boldsymbol{z}) \tag{8}$$

where λ_{reg} is scalar weight for the regularization function $\mathcal{R}(\boldsymbol{z})$. For example, we can use data consistency loss to ensure that the updated variable agrees with the given observation during inverse problem solving [9].

Second key insight arises from the important connection between (7) and the score-distillation loss. Specifically, using the forward diffusion to generate $\boldsymbol{z}_t$ from the clean latent $\boldsymbol{z}$:

$$\boldsymbol{z}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{z} + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \tag{9}$$

the objective function in (7) can be converted to:

$$\|\boldsymbol{z} - \hat{\boldsymbol{z}}_{0|t}\|^2 = \left\| \frac{\boldsymbol{z}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}}{\sqrt{\bar{\alpha}_t}} - \frac{\boldsymbol{z}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_{\theta}(\boldsymbol{z}_t, t)}{\sqrt{\bar{\alpha}_t}} \right\|^2 \tag{10}$$

$$= \frac{1-\bar{\alpha}_t}{\bar{\alpha}_t}\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\theta}(\boldsymbol{z}_t, t)\|^2, \tag{11}$$

which is equivalent to the score-distillation loss up to a constant scaling factor.

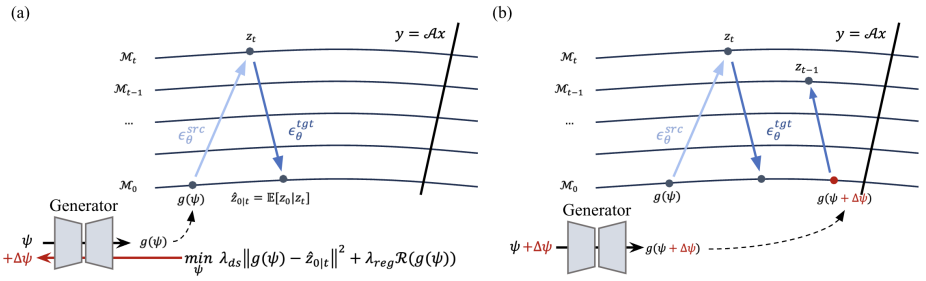

Fig. 3: Unified framework of DreamSampler. (a) Distillation step where the gradient is computed from regularized latent optimization problem. (b) Reverse sampling step where estimated noise by diffusion model is added to the updated generation.

3 DreamSampler

From Section 2.2, it is evident that DDIM sampling inherently includes 'score-distillation' optimization, although it is conducted with $\boldsymbol{z}$ instead of generic parameters ψ. Inspired by this observation, we aim to generalize this optimization-based sampling with arbitrary generic parameter ψ, as in conventional score distillation sampling protocols.

3.1 General Formulation

Suppose that $g(\psi)$ denotes a generated data by an arbitrary generator g and parameter ψ. Inspired by the two key insights described in the previous section, the sampling process of DreamSampler at timestep t is given by[3]

$$\boldsymbol{z}_t = \sqrt{\bar{\alpha}_t} g(\psi_t) + \sqrt{1-\bar{\alpha}_t}\tilde{\boldsymbol{\epsilon}}, \tag{12}$$

where the noise $\tilde{\boldsymbol{\epsilon}}$ is defined as in (5), and

$$\psi_t = \arg\min_{\psi} \|g(\psi) - \hat{\boldsymbol{z}}_{0|t}\|^2 + \lambda_{reg}\mathcal{R}(g(\psi)), \tag{13}$$

$$\hat{\boldsymbol{z}}_{0|t} = \left(\boldsymbol{z}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, t)\right)/\sqrt{\bar{\alpha}_t}. \tag{14}$$

It implies that the score distillation sampling and reverse sampling can be integrated based on this generalized latent optimization framework with proper generator and regularization functions. In the following sections, we further delineate the special cases of DreamSampler.

3.2 DreamSampler with External Generators

Similar to DreamFusion [21], for any differentiable generator g, DreamSampler can feasibly update parameters ψ by leveraging the diffusion model and sharing

[3] Here, we omit the encoder $\mathcal{E}_\phi$ in $\mathcal{E}_\phi(g(\psi))$ for notational simplicity. The encoder maps the generated image $g(\psi)$ to the latent space.

Algorithm 1 Score Distillation

Require: T, ζ, g, ψ, $\mathcal{E}_\phi$, $\{\bar{\alpha}_t\}_{t=1}^T$
$\boldsymbol{z}_0 \leftarrow \mathcal{E}_\phi(\boldsymbol{x}_0)$
for $i = T$ **to** 1 **do**
  $t \sim U[0, T]$
  $\tilde{\boldsymbol{\epsilon}} \sim \mathcal{N}(0, \boldsymbol{I})$
  $\boldsymbol{z}_t \leftarrow \sqrt{\bar{\alpha}_t}\boldsymbol{z}_0 + \sqrt{1-\bar{\alpha}_t}\tilde{\boldsymbol{\epsilon}}$
  $\hat{\boldsymbol{\epsilon}}_\theta \leftarrow \boldsymbol{\epsilon}_\theta^\omega(\boldsymbol{z}_t, t, c)$
  $\nabla_\psi \mathcal{L}_{ds} \leftarrow \tilde{\boldsymbol{\epsilon}} - \hat{\boldsymbol{\epsilon}}_\theta$
  $\psi \leftarrow \psi - \zeta \nabla_\psi \mathcal{L}_{ds}$
  $\boldsymbol{z}_0 \leftarrow \mathcal{E}_\phi(g(\psi))$
end for
return ψ

Algorithm 2 DreamSampler

Require: T, ζ, g, ψ, $\mathcal{E}_\phi$, $\{\bar{\alpha}_t\}_{t=1}^T$
$\boldsymbol{z}_0 \leftarrow \mathcal{E}_\phi(\boldsymbol{x}_0), \boldsymbol{\epsilon}_\theta(\boldsymbol{z}_{T+1}) := \boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$
for $i = T$ **to** 1 **do**
  $t \leftarrow i$, $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$
  $\tilde{\boldsymbol{\epsilon}} \leftarrow \frac{\sqrt{1-\bar{\alpha}_{t-1}-\eta^2\beta_t^2}\hat{\boldsymbol{\epsilon}}_\theta + \eta\beta_t\boldsymbol{\epsilon}}{\sqrt{1-\bar{\alpha}_t}}$
  $\boldsymbol{z}_t \leftarrow \sqrt{\bar{\alpha}_t}\boldsymbol{z}_0 + \sqrt{1-\bar{\alpha}_t}\tilde{\boldsymbol{\epsilon}}$
  $\hat{\boldsymbol{\epsilon}}_\theta \leftarrow \boldsymbol{\epsilon}_\theta^\omega(\boldsymbol{z}_t, t, c)$
  $\nabla_\psi \mathcal{L}_{ds} \leftarrow \tilde{\boldsymbol{\epsilon}} - \hat{\boldsymbol{\epsilon}}_\theta$
  $\psi \leftarrow \psi - \zeta[\nabla_\psi \mathcal{L}_{ds} + \lambda_{reg}\nabla_\psi \mathcal{R}(\boldsymbol{z})]$
  $\boldsymbol{z}_0 \leftarrow \mathcal{E}_\phi(g(\psi))$
end for
return ψ

the same sampling process. To emphasize the distinctions between the original score distillation algorithms and DreamSampler, we conduct a line-by-line comparison of the pseudocode in Algorithm 1 and Algorithm 2.

First, DreamSampler follows the timestep schedule of the reverse sampling process, while distillation sampling algorithms use uniformly random timestep for optimization. This provides us with a novel potential for further refinement in utilizing various time schedulers to improve reconstruction quality or accelerate sampling. Second, as DreamSampler is built upon the general proximal optimization framework, it is compatible with additional regularization functions. From this design, one can explore various applications of the proposed distillation sampling. For example, by defining the regularization function as a data consistency term, one can constrain the generator to reconstruct the true image that aligns with the given measurement for inverse imaging.

Figure 3 illustrates the sampling process of DreamSampler. At each timestep, the generated image $g(\psi)$ is mapped to a noisy manifold by incorporating the estimated noise from the previous timestep, and new noise is subsequently estimated by the diffusion model. The distillation gradient is then computed between these two estimated noises and utilized to update the generator parameters.

3.3 DreamSampler for Image Editing

As DreamSampler is a general framework, we can reproduce other existing algorithms by properly defining $g(\psi)$, $\hat{\boldsymbol{z}}_{0|t}$, and $\mathcal{R}$. As a representative example, here we derive Delta Denoising Score (DDS) [4] for image editing task and discuss its potential extension from the perspective of DreamSampler.

The main assumption of DDS is to decompose the SDS gradient [21] into text component and bias component, where only the text component contains information to be edited according to given text prompt while the bias component includes preserved information. To remove the bias component from distillation gradient, DDS leverages the difference of two conditional predicted noises, $\boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, t, c_{tgt}) - \boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, t, c_{src})$, where c_{tgt} denotes description of editing direction

and c_{src} denotes description of the original image. Specifically, DDS update to clean latent reads [4]

$$\bar{\boldsymbol{z}} = \boldsymbol{z} - \gamma[\boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, t, c_{tgt}) - \boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, t, c_{src})]. \tag{15}$$

In the context of Dreamsampler, let the generator $g(\psi) := \boldsymbol{z}$ be a clean latent. Then, the following Theorem 1 shows that the one-step DDS update can be reproduced with DreamSampler, by defining the regularization function $\mathcal{R}(\boldsymbol{z})$ as Euclidean distance from the posterior mean conditioned on text.

Theorem 1. *Supposed c_{src} in (15) be defined as the null-text, i.e. $c_{src} = c_\varnothing$ and consider text-conditioned posterior mean:*

$$\hat{\boldsymbol{z}}_{0|t}(c) = \mathbb{E}[\boldsymbol{z}_0|\boldsymbol{z}_t, c] = (\boldsymbol{z}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, t, c))/\sqrt{\bar{\alpha}_t}. \tag{16}$$

Then, DDS update in (15) can be obtained from the following latent optimization:

$$\min_{\boldsymbol{z}} \|\boldsymbol{z} - \hat{\boldsymbol{z}}_{0|t}(c_\varnothing)\|^2 + \gamma R(\boldsymbol{z}), \quad \textit{where} \quad R(\boldsymbol{z}) := \frac{\|\boldsymbol{z} - \hat{\boldsymbol{z}}_{0|t}(c_{tgt})\|^2}{(1-\gamma)} \tag{17}$$

Furthermore, it is equivalent to Tweedie's formula with CFG, i.e.:

$$\hat{\boldsymbol{z}}^\gamma_{0|t}(c_{tgt}) := \frac{\boldsymbol{z}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}^\gamma_\theta(\boldsymbol{z}_t, t, c_{tgt})}{\sqrt{\bar{\alpha}_t}} \tag{18}$$

where $\boldsymbol{\epsilon}^\gamma_\theta(\boldsymbol{z}_t, t, c_{tgt}) = \boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, t, c_\varnothing) + \gamma[\boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, t, c_{tgt}) - \boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, t, c_\varnothing)]$.

Theorem 1 reveals that the one step latent optimization (17) of DreamSampler reproduces the DDS update (15). That being said, the main advantages of DreamSampler stems from the added noise to updated source image $\boldsymbol{z}_0$. Specifically, in contrast to the original DDS method that adds newly sampled Gaussian noise to $\boldsymbol{z}_0$, DreamSampler adds the estimated noise by $\boldsymbol{\epsilon}_\theta$ in the previous timestep of reverse sampling. Initiated from the inverted noise, reverse sampling do not deviate significantly from the reconstruction trajectory even though source prompt is not given, because the latent optimization (17) represents a proximal problem that regulates the sampling process. Consequently, DreamSampler only requires text prompt that describe the editing direction for the real imaging editing.

Finally, it is noteworthy that the equivalent interpretation (18) could be readily extended to spatially localized distillation by computing the solution as

$$\hat{\boldsymbol{z}}^\gamma_{0|t} = \left(\boldsymbol{z}_t - \sqrt{1-\bar{\alpha}_t}\sum_i \mathcal{M}_i \odot \boldsymbol{\epsilon}^\gamma_\theta(\boldsymbol{z}_t, t, c^{(i)}_{tgt})\right)/\sqrt{\bar{\alpha}_t} \tag{19}$$

[4] The update term is equivalent to (3) in [4] when $\theta = \boldsymbol{z}$, $\hat{\boldsymbol{y}} = c_{src}$, and $\boldsymbol{y} = c_{tgt}$.

where $\mathcal{M}_i$ denotes the pixel-wise mask, $\odot$ is element-wise multiplication, and $c_{tgt}^{(i)}$ denote ith mask and corresponding text prompt pair. The entire process for the real image editing via DreamSampler is described in Algorithm 3. Remark that the Algorithm 3 is equivalent to the Algorithm 2 through Theorem 1, by setting $\eta\beta_t = 0$ to achieve the deterministic sampling, and by setting $\omega = 0$ to ensure that the latent optimization problem is solved during unconditional sampling. Line 9 of the Algorithm 2 is disappeared since we are assuming that $\psi = \boldsymbol{z}_0$ and g as identity mapping.

3.4 DreamSampler for Inverse Problems

DreamSampler can leverage multiple regularization terms to precisely constrain the sampling process to solve inverse problems. For example, we can solve the text-guided image inpainting task by defining the regularization function as

$$\mathcal{R}(\boldsymbol{z}) = (1-\gamma)\|\mathcal{M} \odot (\boldsymbol{z} - \hat{\boldsymbol{z}}_{0|t}(c_{tgt}))\|^2 + \gamma\|\boldsymbol{y} - \mathcal{A}\mathcal{D}_\varphi(\boldsymbol{z})\|^2, \tag{20}$$

where $\mathcal{M}$ denotes the operation to create the measurement by masking out the target region. This regularization term implies that the sample image satisfies data consistency for regions where the true signal is preserved, while guiding the masked region to reflect the target text prompt. When we solve the entire latent optimization problem, we follows two-step approach of TReg [9]. For the details, refer to the appendix. The main difference with TReg is that we separate the text-guidance from the data consistency term. In other words, we initialize the $\boldsymbol{z}$ for the data consistency term as $\hat{\boldsymbol{z}}_{0|t}(c_\varnothing)$ while TReg uses $(1-\omega)\hat{\boldsymbol{z}}_{0|t}(c_\varnothing) + \omega\hat{\boldsymbol{z}}_{0|t}(c_{tgt})$ where the ω denotes the CFG scale. This difference allows DreamSampler to solve the inpainting problem with different text-guidance for each masked region, by combining localized distillation approach introduced in Section 3.3.

4 Experimental Results

4.1 Image Restoration through Vectorization

As a novel application that other methods have not explored, here we present an image vectorization from blurry measurement using DreamSampler. In this sce-

Algorithm 3 DreamSampler for Image Editing

Require: source image $\boldsymbol{x}$, image encoder $\mathcal{E}_\phi$, latent diffusion model $\boldsymbol{\epsilon}_\theta$, null-text embedding $c_\varnothing$, conditioning text embedding c_{tgt}.
  $\boldsymbol{z}_0 \leftarrow \mathcal{E}_\phi(\boldsymbol{x})$
  $\boldsymbol{z}_T \leftarrow \text{Inversion}(\boldsymbol{z}_0)$
  for $t \in [T, 0]$ **do**
    $\hat{\boldsymbol{\epsilon}}_\theta \leftarrow \boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, t, c_\varnothing) + \gamma[\boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, t, c_{tgt}) - \boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, t, c_\varnothing)]$
    $\bar{\boldsymbol{z}} \leftarrow (\boldsymbol{z}_t - \sqrt{1-\bar{\alpha}_t}\hat{\boldsymbol{\epsilon}}_\theta)/\sqrt{\bar{\alpha}_t}$
    $\boldsymbol{z}_t \leftarrow \sqrt{\bar{\alpha}_{t-1}}\bar{\boldsymbol{z}} + \sqrt{1-\bar{\alpha}_{t-1}}\boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, t, c_\varnothing)$
  end for

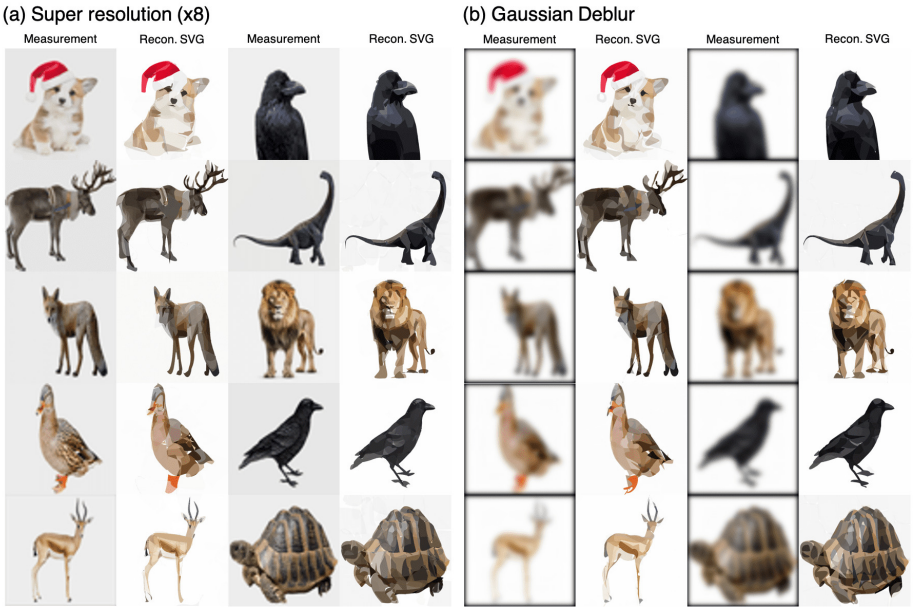

Fig. 4: Representative results for image vectorization task with image reconstruction.

nario, the generator $g(\cdot)$ corresponds to a differentiable rasterizer (DiffVG [11]), and the parameters ψ of the generator consist of path parameters comprising Scalable Vector Graphics (SVG).

Specifically, we address a text-guided SVG inverse problem, acknowledging that low-quality, noisy measurements can detract from the detail and aesthetic quality of vector designs. Here, our goal is to accurately reconstruct SVG paths for the parameter ψ that, when rasterized, align closely with the provided measurements $\boldsymbol{y}$ and text conditions $c_{\boldsymbol{y}}$, related through the forward measurement operator $\mathcal{A}$. In this context, the regularizer $\mathcal{R}$ in (14) corresponds to the data consistency term. Forward measurement operators are specified as follows: (**a**) For super-resolution, bicubic downsampling is performed with scale $\times 8$. (**b**) For Gaussian blur, the kernel has size 61×61 with a standard deviation of 5. Then, the latent optimization framework of the SVG inverse problem is defined as:

$$\min_{\psi}(1-\gamma)\lambda_{SDS}\|\mathcal{E}(g(\psi)) - \hat{\boldsymbol{z}}_{0|t}(c_{\boldsymbol{y}})\|^2 + \gamma\lambda_{DC}\|\boldsymbol{y} - \mathcal{A}g(\psi)\|^2, \tag{21}$$

where we found out that $\gamma = \bar{\alpha}_t$ works well in practice. SVG primitives ψ are initialized with radius 20, random fill color, and opacity uniformly sampled between 0.7 and 1, following [8]. In this paper, we use closed Bézier curves for iconography artistic style. For the optimization of (21), we use Adam optimizer with (β_1, β_2)=(0.9, 0.9). For the text condition $c_{\boldsymbol{y}}$, we append a suffix to the object[5]. Additional experimental details are provided in the appendix.

[5] e.g. "{*a cute fox*}, *minimal flat 2d vector icon. lineal color. on a white background. trending on artstation.*"

Fig. 5: Qualitative comparison of SVG reconstruction. For baselines, we first obtain an initial reconstruction using PSLD [25], vectorize it with LIVE [15], and refine the output vector with VectorFusion [8] or PDS [10]. DreamSampler outperforms this multi-step approach by simultaneously solving the inverse problem and updating SVG parameters via score distillation.

Comparatively, the proposed framework is evaluated against a multi-stage baseline approach involving initial rasterized image reconstruction using the state-of-the-art solver (PSLD [24]) that is based on the latent diffusion model. Then, we vectorize the reconstruction using the off-the-shelf Layer-wise Image Vectorization program (LIVE [15]). An optional step includes refining the SVG output via latent score distillation sampling algorithms, exemplified by VectorFusion [8] and Posterior Distillation Sampling (PDS, [10]).

Figure 4 illustrates that the proposed framework achieves high-quality SVG reconstructions with semantic alignment closely matching the specified text condition $c_{\boldsymbol{y}}$. In contrast, Figure 5 highlights the deficiencies of the multi-stage baseline methods, particularly its inability to retain detailed fidelity. Errors accumulate during the initial restoration process, resulting in blurriness and undesirable path overlap in the vector outputs. While the solver may achieve adequate reconstructions, the subsequent vectorization step disregards text caption $c_{\boldsymbol{y}}$, leading to the potential loss of details and semantic coherence. Additional VectorFusion fine-tuning loses consistency with the measurement. Conversely, DreamSampler effectively restores SVGs by directly utilizing latent-space diffusion and text conditions for both vectorization and reconstruction, ensuring the preservation of detail and contextual relevance.

4.2 Real Image Editing

For real image editing via DreamSampler, we leverage the Stable-Diffusion v1.5 provided by HuggingFace. We use linear time schedule and set NFE to 200

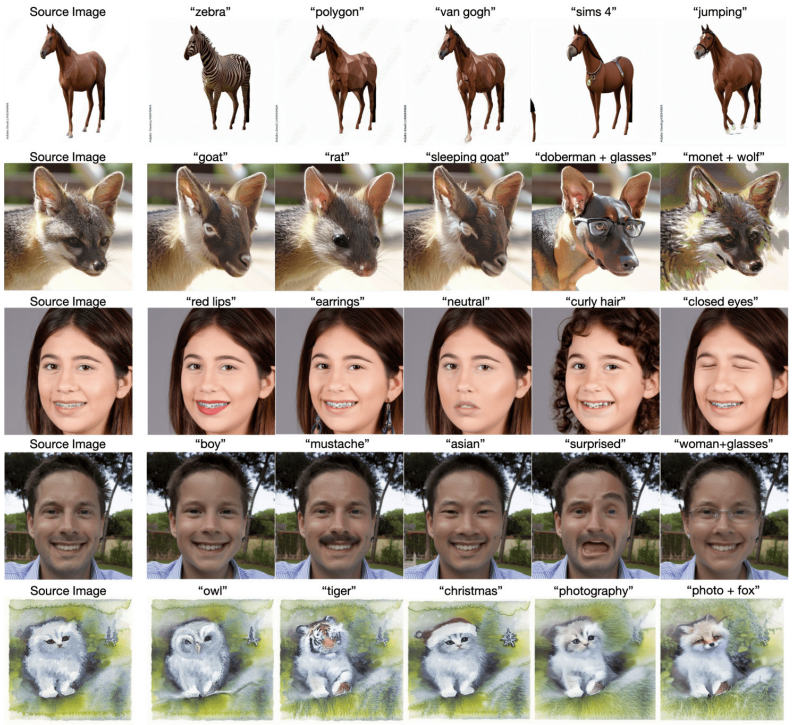

Fig. 6: Representative results for real image editing via distillation through reverse sampling. We leverage source images from various domains and the caption above each image denotes text prompt reflecting the editing direction. The result demonstrates that distillation could be effectively conducted during the reverse sampling.

for both the DDIM inversion and reverse sampling. For more details on hyperparameter setting, please refer to the appendix.

To demonstrate the ability of DreamSampler in real image editing, we leverage source images from various domains, including photographs of animals, human faces and drawings. All results in Figure 6 show that DreamSampler accurately reflects the provided text prompts for editing. Specifically, DreamSampler does not change bias components, which are intended not to be edited by text prompt. For instance, in 3rd and 4th rows of Figure 6, features such as braces and background including striped shirt, are well-maintained while the text prompts are accurately reflected. Moreover, DreamSampler effectively reflects multiple editing directions simultaneously such as "doberman" + "wearing glasses", "woman" + "wearing glasses" or "photography" + "fox". We next conduct a qualitative comparison with the original DDS algorithm to show the improvement from integration of distillation into reverse sampling. As illustrated in Figure 7, DreamSampler is capable of editing according to text prompt robustly across various domains of source images. Furthermore, DreamSampler achieves better fidelity and bias component preservation compared to the original DDS.

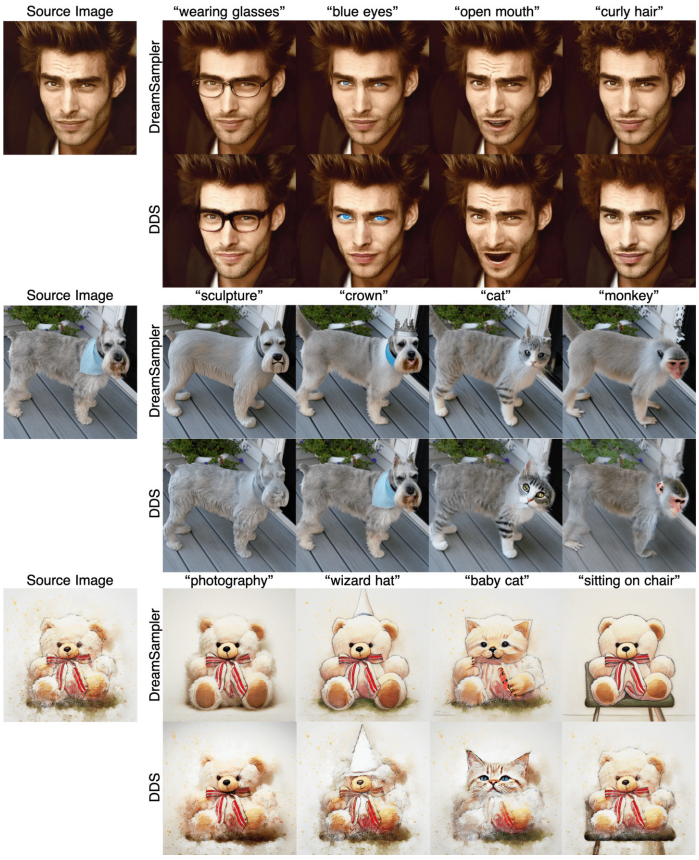

Fig. 7: Qualitative comparison with DDS for the real image editing. Both DreamSampler and DDS edit images following target text, but DreamSampler achieves higher fidelity with preserving bias component.

For the quantitative comparison, we compare DreamSampler against diffusion-based editing algorithms [5, 16, 18], following the experimental setups of [19, 20]. We use the prompt "a photograph of {}" with the source and target objects inserted. For the CFG scale, we use $0.15\bar{\alpha}_t$ across all cases. Table 1 demonstrates that DreamSampler outperforms most baselines in image editing tasks. Specifically, DreamSampler generates glasses more naturally in the "cat $\rightarrow$ cat w/ glasses" task, improves fidelity in source image edits compared to baseline algorithms as shown in Figure S10. Some baselines achieve high CLIP accuracy by focusing on the generation of glasses, regardless of its natural appearance.

4.3 Text-guided Image Inpainting

For the text-guided image inpainting task, we also use Stable-Diffusion v1.5, linear time schedule, and 200 NFE. In addition to solving (20), we apply DPS [1]

Table 1: Comparison to diffusion-based editing methods. Dist for DINO-ViT Structure Distance. Baseline results are from [19].

Method	Cat→Dog		Horse→Zebra		Cat→ Cat w/ glasses	
	CLIP-Acc ↑	Dist ↓	CLIP-Acc ↑	Dist ↓	CLIP-Acc ↑	Dist ↓
SDEdit + word swap	71.2%	0.081	92.2%	0.105	34.0%	0.082
DDIM + word swap	72.0%	0.087	94.0%	0.123	37.6%	0.085
prompt to prompt	66.0%	0.080	18.4%	0.095	69.6%	0.081
p2p-zero	92.4%	0.044	75.2%	0.066	71.2%	0.028
EBCA	93.7%	0.040	90.4%	0.061	81.1%	0.052
DreamSampler (Ours)	90.3%	**0.029**	**95.2%**	**0.038**	48.3%	**0.025**

Table 2: Comparison to text-conditioned diffusion-based inpainting solvers. **Bold**: the best score, Underline: the second best.

Method	Glasses			Smile		
	PSNR ↑	FID ↓	CLIP-sim ↑	PSNR ↑	FID ↓	CLIP-sim ↑
Stable-Inpaint	19.82	54.26	0.281	25.33	**19.22**	**0.249**
TReg	21.97	61.04	**0.288**	26.71	24.48	**0.249**
DreamSampler (Ours)	**24.61**	**27.10**	0.263	**27.90**	24.33	0.242

steps during sampling by following TReg [9] to enhance the consistency of masked region and other regions. We generate measurements by masking out two rectangular regions on the eyes and mouth, where the masked region is determined based on the averaged face of the 1k FFHQ validation set. The size of measurement is 512×512. We solve the inpainting problem by giving text prompts "a photography of face wearing glasses" and "a photography of face with smile". Figure 8 shows that DreamSampler fills masked region by reflecting given text prompt accurately. While the text guidance is applied inside the mask, data consistency gradients combined with the reconstruction by inversion is applied outside the mask, which results in superior fidelity of the output image. Note that we display the image output as is without any post-processing such as projection[6]. The bottom row of Figure 8 depict the solution for inpainting problem with two different masks and distinct text guidance. Through localized distillation gradient, DreamSampler generates solutions according to the provided guidance. DreamSampler generates masked regions with better robustness than TReg. Additionally, in the case of multiple masks, DreamSampler successfully reflects text in the correct regions via localized distillation gradients, whereas TReg fails to meet one of the conditions. For more results, refer to appendix.

We also evaluate both the quality of reconstructions (PSNR, FID) and the accuracy of the text guidance (CLIP similarity) on 1k FFHQ validation set. For baselines, we select diffusion-based image inpainting models [14, 23] and a text-guided inverse problem solver [9]. Specifically, Stable-Inpaint [23] is a fine-tuned StableDiffusion model specialized for inpainting task. Table 2 shows that DreamSampler outperforms the baselines in terms of PSNR and FID score while

[6] For linear operator $\mathcal{A}$ and measurement $\boldsymbol{y}$, the projection means $\mathcal{A}^\top \mathcal{A}\boldsymbol{y} + (\boldsymbol{I} - \mathcal{A}^\top \mathcal{A})\boldsymbol{x}$.

Fig. 8: Qualitative comparison for text guided image inpainting task. DreamSampler can generate more realistic images according to given text prompt.

achieving comparable CLIP similarity. Stable-Inpaint achieves a lower FID score, while DreamSampler, serving as a zero-shot inpainting solver, achieves a higher PSNR. This suggests superior reconstruction quality of Dreamsampler through data consistency update.

5 Conclusion

We presented DreamSampler, a unified framework of reverse sampling and score distillation by taking the advantages of each algorithms. Specifically, we connected two distinct algorithms under the perspective of latent optimization problem. Consequently, we introduced a generalized optimization framework, which offers new design space to solve various applications. Especially, DreamSampler enables to combine various regularization functions to constrain the sampling process. Additionally, we provided three applications including image vectorization with reconstruction, real image editing, and image inpainting. DreamSampler could be extended to other algorithms by defining appropriate regularization functions. The codebase is available to public at https://github.com/DreamSampler/dream-sampler.

Potential negative social impact. The performance of algorithms established on DreamSampler heavily depends on the prior of diffusion model. Hence, the proposed method basically influenced by the potential negative impacts of LDM itself. Thus, proper political regulation is required to mitigate these risks.

Acknowledgments This work was supported by the National Research Foundation of Korea under Grant RS-2024-00336454, by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT)) (No. RS-2022-II220984, Development of Artificial Intelligence Technology for Personalized Plug-and-Play Explanation and Verification of Explanation, and No.2019-0-00075, Artificial Intelligence Graduate School Program(KAIST)), by Culture, Sports and Tourism R&D Program through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports and Tourism in 2023, by Field-oriented Technology Development Project for Customs Administration funded by the Korea government (the Ministry of Science & ICT and the Korea Customs Service) through the National Research Foundation (NRF) of Korea under Grant NRF2021M3I1A1097910.

References

1. Chung, H., Kim, J., Mccann, M.T., Klasky, M.L., Ye, J.C.: Diffusion posterior sampling for general noisy inverse problems. arXiv preprint arXiv:2209.14687 (2022) 2, 12
2. Chung, H., Lee, S., Ye, J.C.: Fast diffusion sampler for inverse problems by geometric decomposition. arXiv preprint arXiv:2303.05754 (2023) 3
3. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. Advances in neural information processing systems **34**, 8780–8794 (2021) 2
4. Hertz, A., Aberman, K., Cohen-Or, D.: Delta denoising score. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2328–2337 (2023) 6, 7
5. Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-Or, D.: Prompt-to-prompt image editing with cross attention control. arXiv preprint arXiv:2208.01626 (2022) 12
6. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020) 1, 3, 4
7. Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598 (2022) 4
8. Jain, A., Xie, A., Abbeel, P.: Vectorfusion: Text-to-svg by abstracting pixel-based diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1911–1920 (2023) 9, 10, 18
9. Kim, J., Park, G.Y., Chung, H., Ye, J.C.: Regularization by texts for latent diffusion inverse solvers. arXiv preprint arXiv:2311.15658 (2023) 4, 8, 13, 19
10. Koo, J., Park, C., Sung, M.: Posterior distillation sampling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13352–13361 (2024) 10
11. Li, T.M., Lukáč, M., Gharbi, M., Ragan-Kelley, J.: Differentiable vector graphics rasterization for editing and learning. ACM Transactions on Graphics (TOG) **39**(6), 1–15 (2020) 9
12. Liu, G.H., Vahdat, A., Huang, D.A., Theodorou, E.A., Nie, W., Anandkumar, A.: I2sb: Image-to-image schrodinger bridge. arXiv preprint arXiv:2302.05872 (2023) 25
13. Liu, Y., Lin, C., Zeng, Z., Long, X., Liu, L., Komura, T., Wang, W.: Syncdreamer: Generating multiview-consistent images from a single-view image. arXiv preprint arXiv:2309.03453 (2023) 25

14. Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., Van Gool, L.: Repaint: Inpainting using denoising diffusion probabilistic models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11461–11471 (2022) 13
15. Ma, X., Zhou, Y., Xu, X., Sun, B., Filev, V., Orlov, N., Fu, Y., Shi, H.: Towards layer-wise image vectorization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16314–16323 (2022) 10
16. Meng, C., Song, Y., Song, J., Wu, J., Zhu, J.Y., Ermon, S.: Sdedit: Image synthesis and editing with stochastic differential equations. arXiv preprint arXiv:2108.01073 (2021) 12
17. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021) 24
18. Mokady, R., Hertz, A., Aberman, K., Pritch, Y., Cohen-Or, D.: Null-text inversion for editing real images using guided diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6038–6047 (2023) 12
19. Park, G.Y., Kim, J., Kim, B., Lee, S.W., Ye, J.C.: Energy-based cross attention for bayesian context update in text-to-image diffusion models. Advances in Neural Information Processing Systems **36** (2024) 12, 13
20. Parmar, G., Kumar Singh, K., Zhang, R., Li, Y., Lu, J., Zhu, J.Y.: Zero-shot image-to-image translation. In: ACM SIGGRAPH 2023 Conference Proceedings. pp. 1–11 (2023) 12
21. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. arXiv preprint arXiv:2209.14988 (2022) 2, 4, 5, 6
22. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021) 4
23. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022) 3, 13
24. Rout, L., Raoof, N., Daras, G., Caramanis, C., Dimakis, A., Shakkottai, S.: Solving linear inverse problems provably via posterior sampling with latent diffusion models. Advances in Neural Information Processing Systems **36** (2024) 10
25. Rout, L., Raoof, N., Daras, G., Caramanis, C., Dimakis, A.G., Shakkottai, S.: Solving linear inverse problems provably via posterior sampling with latent diffusion models. arXiv preprint arXiv:2307.00619 (2023) 10
26. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020) 1, 3, 4
27. 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) 1, 2, 3, 4
28. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018) 25
29. Zhu, J., Zhuang, P.: Hifa: High-fidelity text-to-3d with advanced diffusion guidance. arXiv preprint arXiv:2305.18766 (2023) 25, 27