Rethinking Score Distillation as a Bridge Between Image Distributions

David McAllister[1*] **Songwei Ge**[2*] **Jia-Bin Huang**[2] **David W. Jacobs**[2]
Alexei A. Efros[1] **Aleksander Holynski**[1] **Angjoo Kanazawa**[1]
[1] UC Berkeley [2] University of Maryland
https://sds-bridge.github.io/

Abstract

Score distillation sampling (SDS) has proven to be an important tool, enabling the use of large-scale diffusion priors for tasks operating in data-poor domains. Unfortunately, SDS has a number of characteristic artifacts that limit its usefulness in general-purpose applications. In this paper, we make progress toward understanding the behavior of SDS and its variants by viewing them as solving an optimal-cost transport path from a source distribution to a target distribution. Under this new interpretation, these methods seek to transport corrupted images (source) to the natural image distribution (target). We argue that current methods' characteristic artifacts are caused by (1) linear approximation of the optimal path and (2) poor estimates of the source distribution. We show that calibrating the text conditioning of the source distribution can produce high-quality generation and translation results with little extra overhead. Our method can be easily applied across many domains, matching or beating the performance of specialized methods. We demonstrate its utility in text-to-2D, text-based NeRF optimization, translating paintings to real images, optical illusion generation, and 3D sketch-to-real. We compare our method to existing approaches for score distillation sampling and show that it can produce high-frequency details with realistic colors.

1 Introduction

Diffusion models have shown tremendous success in modeling complex data distributions like images [51, 54, 3, 22], videos [59, 4] and robot action policies [13]. In domains where data is plentiful, they produce state-of-the-art results. Many data modalities, however, cannot enjoy the same scaling benefits due to their lack of sufficiently large datasets. In these cases, it is useful to exploit diffusion models trained on domains with rich data sources as a prior in an optimization framework. Score Distillation Sampling (SDS) [48, 69] and its variants [70, 20, 76] are a widely adopted way to optimize parametric images, *i.e.*, images produced by a model like NeRF, with a pre-trained diffusion model. Despite being applicable to a wide range of applications, SDS is also known to suffer from several significant artifacts, such as oversaturation and oversmoothing. As such, several variants have been proposed to alleviate these artifacts [70, 76, 32], often at the cost of efficiency, diversity, or other artifacts.

In this paper, we investigate the core issues with SDS by casting the class of score distillation optimization problems as a Schrödinger Bridge (SB) problem [55, 12, 11, 42], which finds the optimal transport between two distributions. Specifically, given some images from the current optimized distribution (*e.g.*, renderings from a NeRF), applying the transport maps them to their pair images in a target distribution (*e.g.*, text-conditioned natural image distribution). The density flow formed by these mappings is transport-optimal, as defined in the SB problem. In an optimization framework, the difference between paired source and target samples, computed with an SB, can be

[*]Equal contribution.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-1064

used as a gradient to update the source. Su *et al*. [65] have shown that this path can be explicitly solved using two pre-trained diffusion models. We show that one can also compose these models as an optimizer to approximate transport paths on the fly.

Under this framework, we can understand SDS and its variants as approximating a source-to-target distribution bridge with the difference of two denoising directions. The denoising scores point to the source and target distributions respectively, with the source representing the current optimized image that updates with each optimization step.

This framing reveals two sources of errors. First, these methods are a first-order approximation of the diffusion bridge. Specifically, Gaussian noise is sampled to perturb the current optimized image, and single denoising steps, instead of the full PF-ODE simulation, are used to estimate the transport. This induces error in estimating the desired path. Recent works [34, 41] that use multi-step estimation can be explained as mitigating this error. Second, estimating the denoising direction to the current source distribution is non-trivial, since the current optimized image may not necessarily look like a real image (*e.g*., initializing with Gaussian noise or starting from a render of an untextured 3D model). Our analysis reveals that SDS approximates the current distribution with the unconditional image distribution, which is not accurate and results in a *distribution mismatch error*. We show that recent SDS variants [70, 76, 32] can be seen as proposals to improve this distribution mismatch error.

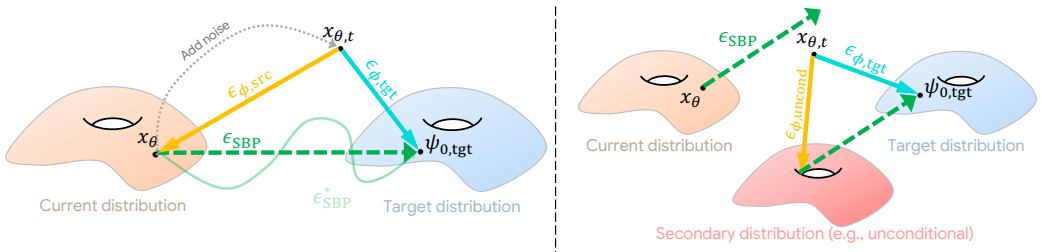
Finally, our analysis motivates a simple method that rectifies the distribution mismatch issue without additional computational overhead. Our insight is that the large-scale text-to-image diffusion models learn from billions of caption-image pairs [56], where a breadth of image corruptions are present in their training sets. They are also equipped with powerful pre-trained text encoders, which empower the models with zero-shot capacity in generating unseen concepts [53, 52]. As such, simply describing the current source distribution with text, even if it is not part of the real image manifold, can approximate the distribution of the current optimized image, leading to improved transport paths. Our simple and efficient solution can be easily applied to any existing application that uses SDS. We show that it consistently improves the visual quality in the desired domain. We comprehensively compare our approach with standard distillation sampling methods over several generation tasks, where our approach matches or outperforms the baselines.

Our contributions are as follows:

- We propose to cast the problem of using a pre-trained diffusion model as a prior in an optimization problem as solving the Schrödinger Bridge (SB) problem between two image distributions. Specifically, it can be seen as bridging the distribution of the current optimized image to the target distribution under a dual-bridge framework.
- We analyze recent SDS-based methods under the lens of our framework and explain the pros and cons of the individual methods.
- Our analysis motivates a simple yet effective alternative to SDS by using textual descriptions to specify the current optimized image distribution. It achieves consistently more realistic results than SDS, producing quality comparable with VSD [70] without its computational overhead. We compare various generation tasks to show its wall-clock efficiency and quality generations against state-of-the-art methods.

2 Related Work

Score Distillation Sampling Modalities like 3D, 4D, sketch, and vector graphics (SVGs) lack the large-scale, diverse, and high-quality datasets needed to train a domain-specific diffusion model. In these domains, previous works explore exploiting image or video as a proxy modality [26, 16]. By computing the gradient on a proxy representation with a pretrained model, optimization in the target modality is viable with differentiable mappings, e.g. differentiable rasterization [33] for SVGs or differentiable rendering [44] for 3D objects and scenes. The seminal method, Score Distillation Sampling (SDS) [48], first proposed to apply a pretrained text-to-image diffusion model for text-to-3D generation. However, it requires a high classifier-free guidance weight and, therefore, suffers from artifacts such as over-saturation and over-smoothing. Recent works have built upon SDS to adapt it for editing tasks [30, 20, 46, 29] or more broadly improve over the original SDS formulation [28, 1, 70, 77, 76, 78]. NFSD [28] and LMC-SDS [1] inspect the individual components of the SDS gradient and propose methods to rectify the high guidance weights. However, the over-



(a) Score distillation as a bridge between two distributions.

(b) Distribution mismatch error.

Figure 1: **Optimization with diffusion models as approximation of a Schrödinger Bridge Problem (SBP).** (a) We propose to formulate optimization with diffusion models as bridging the distribution of the current optimized image x_θ to the target distribution under a dual-bridge framework (a). Current methods can be interpreted as approximating the optimal transport ϵ^*_{SBP} between these distributions via the difference between projections of a noised image $x_{\theta,t}$ onto the two distributions. This analysis reveals two sources of error: (1) these gradients are linear approximations of the optimal path, as illustrated in (a), and (2) the source distribution used for computing this approximation (*e.g.*, the unconditional distribution in SDS [48]) may not be aligned with the current distribution, illustrated in (b).

saturation problem is mitigated but not fully resolved. VSD [70] formulates the problem as particle-based variational inference and proposes to train a LoRA [24] on the fly to estimate the score of proxy distribution. We present a new framework that allows rethinking all the variants under the same lens. This framework also motivates a method that improves the quality of SDS without losing efficiency.

Visual Content Generation with SDS Since SDS was developed for text-to-3D generation, it has also been adopted to generate various other visual content such as SVGs [18, 73], sketches [72], texture [43, 6–8, 75], typography [25], 3D bodies [45], dynamic 4D scenes [2, 60, 37] and illusions [5]. Among these applications, text-to-3D has been the most active research direction. In addition to designing better distillation sampling methods [70, 77, 28], prior work has also studied the underlying 3D neural representations [74, 66, 35, 9] and leveraging multiview data to improve the 3D consistency [57, 40, 39, 49, 78]. We note that these explorations are orthogonal to our study and should be able to work jointly with our method. In this paper, we look into existing applications like text-based NeRF optimization, painting-to-real, and illusion generation. We also propose a new AR application called 3D sketch-to-real.

3 Method

In this section, we present an analytical framework that casts the score distillation sampling (SDS) family of methods as instantiations of a Schrödinger Bridge problem. We show that many recent SDS based methods can be interpreted as an online solver for the problem. That is, each SDS optimization step is a first-order approximation of a dual diffusion bridge formed by two probability flow (PF) ODEs [65]. We analyze SDS and its variants under this general framework. Then, we present a simple solution based on the analysis, which leads to significant quality improvement with little extra computational overhead.

3.1 Background

Diffusion models define a forward "noising" process that degrades data samples $\mathbf{x}$ gradually from the image distribution to noised samples $\mathbf{z}_t$, and eventually the i.i.d. Gaussian distribution [23, 62]. This process is indexed by timesteps t, where $t = 1$ indexes the full Gaussian noise distribution and $t = 0$ indexes the data distribution. A diffusion model, parameterized by ϕ, is then trained to reverse this encoding process, iteratively transforming the noise distribution into the data distribution with the following denoising objective:

$$\mathcal{L}_{\text{Diff}}(\phi, \mathbf{x}) = \mathbb{E}_{t\sim\mathcal{U}(0,1),\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[w(t)\left\|\epsilon_\phi\left(\alpha_t\mathbf{x} + \sigma_t\epsilon; y, t\right) - \epsilon\right\|_2^2\right], \tag{1}$$

where $w(t)$ is a loss weighting function, y is a conditioning text prompt, and α_t and σ_t are hyperparameters from the predefined noise schedule.

https://doi.org/10.52202/079017-1064

Probability Flow ODE. Denoising score matching [64, 27, 61] shows that the diffusion model denoising prediction can be rewritten as a score vector field:

$$\nabla_{\mathbf{x}} \log p_t(\mathbf{x}) = -\frac{1}{\sqrt{1-\alpha_t}}\epsilon_t. \tag{2}$$

Because of its special connection to marginal probability densities, the resulting ODE is named the probability flow (PF) ODE with the following expression:

$$dx = [f(\mathbf{x}, t) - \frac{1}{2}g^2(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x}))]dt, \tag{3}$$

where $f(\mathbf{x}, t)$ and $g(t)$ are pre-defined schedule parameters. This PF-ODE can be solved deterministically [63], mapping a noise sample to its corresponding data sample through the reverse process and the opposite through the forward process (inversion). This cycle-consistent conversion between image and latent representations is important in establishing dual diffusion implicit bridges.

Dual Diffusion Implicit Bridges. Dual Diffusion Implicit Bridges (DDIBs) [65] compose a diffusion inversion and generation process for solving image-to-image translation problems without requiring a paired image dataset. Instead, DDIBs use two diffusion models trained on different domains (or, analogously, one model with two different text conditions). DDIB inverts the source image into a noise latent via the forward PF-ODE and then decodes the latent in the target domain via the reverse PF-ODE. DDIBs can be interpreted as a concatenation of the Schrödinger Bridges from source-to-latent and latent-to-target, hence the dual bridges in its name. DDIBs enable solving transport between two distributions using a single pre-trained diffusion model. We build on this insight in an optimization context.

3.2 Optimization with Diffusion Model Approximates a Dual Schrödinger Bridge

Many generative vision tasks involve optimizing corrupted images to the image manifold. For example, in 3D generation, a 3D representation like NeRF is optimized to render natural images matching a prescribed text prompt. Methods like SDS enable this by using a pre-trained diffusion model as a prior. We propose formulating such optimization problems as solutions to an instantiation of a Schrödinger Bridges Problem (SBP). SBP finds cost-optimal paths between a source image distribution p_{src} and a target image distribution p_{tgt} [68, 14]. Optimizing a parametrized image toward the natural image distribution can be cast as finding the optimal paths between the current optimized image(s) and the natural image distribution. Instead of solving this problem directly, which would require training a generative model from scratch [38, 14, 10], we show that pre-trained diffusion models can be exploited as an optimizer that approximates the path. Further, the gradient computed by the existing score distillation methods can be viewed as the first-order approximation of this path. This formulation is illustrated in Figure 1.

Let $\mathbf{x}_\theta \in \mathbb{R}^d$ represent a parametric image, *i.e.*, an image produced differentiably by a model with parameter θ, such as a NeRF. To leverage the pretrained diffusion model, we add noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ to obtain a latent at timestep t:

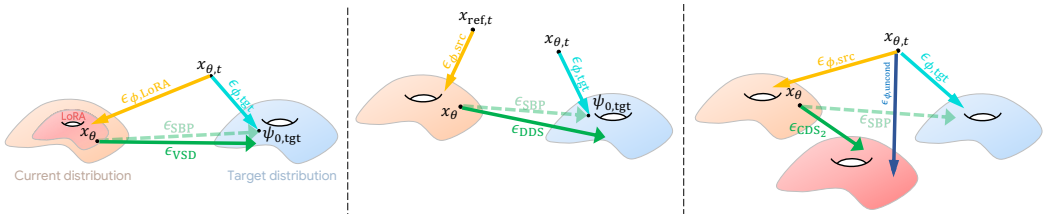
$$\mathbf{x}_{\theta,t} = \alpha_t \mathbf{x}_\theta + \sigma_t \epsilon \tag{4}$$

Suppose that $\psi_{t',\text{src}}$ and $\psi_{t',\text{tgt}}$ denote the paths obtained by solving the PF ODE as in Eq. 3 from t to 0, both starting from $\mathbf{x}_{\theta,t}$, such that $\psi_{0,\text{src}} \in p_{\text{src}}$, $\psi_{0,\text{tgt}} \in p_{\text{tgt}}$, $\psi_{t,\text{src}} = \psi_{t,\text{tgt}} = \mathbf{x}_{\theta,t}$. This forms a dual diffusion bridge [65] from $\psi_{0,\text{src}}$ to $\psi_{0,\text{tgt}}$. We approximate this path *per-iteration* using a pretrained diffusion model. We denote the displacement of this path as:

$$\epsilon^*_{\text{SBP}} = \psi_{0,\text{tgt}} - \psi_{0,\text{src}}. \tag{5}$$

Fully simulating this bridge involves solving two PF ODEs, which invokes dozens of neural function evaluations (NFEs) to estimate the gradient of each iteration. Instead, one can estimate each half of the bridge with a single-step prediction by computing two denoising directions $\epsilon_{\phi,\text{src}}$ and $\epsilon_{\phi,\text{tgt}}$. We thus obtain a first-order approximation of a dual diffusion bridge with the difference vector:

$$\epsilon_{\text{SBP}} = \epsilon_{\phi,\text{tgt}} - \epsilon_{\phi,\text{src}}, \tag{6}$$



(a) Variational Score Distillation (b) Delta Distillation Sampling (c) Classifier Score Distillation (2nd term)

Figure 2: **Comparision of SDS variants under our analysis**. We illustrate the major gradient components of different SDS variants and provide a straightforward comparison with ϵ_{SBP}.

which is subject to the following sources of errors.

1. **First-order approximation error**. Instead of performing full PF-ODE simulations, the single-step noising and prediction are less accurate and induce errors. Recent work ISM [34] can be interpreted as reducing this error with a multi-step simulation to obtain $\mathbf{x}_{\theta,t}$.
2. **Source distribution mismatch**. The dual diffusion bridge relies on $\epsilon_{\phi,\text{src}}$ accurately estimating the distribution of the current sample, $\mathbf{x}_\theta$. A series of works can be viewed as improving this error [70, 28, 76] by computing more accurate $\epsilon_{\phi,\text{src}}$.

We show that $\epsilon_{\phi,\text{tgt}} - \epsilon_{\phi,\text{src}}$ is an effective gradient when both the source and target distribution are well expressed. Next, we discuss the popular score distillation methods under this analysis. We argue that their characteristic artifacts can largely be understood due to the errors above.

3.3 Analyzing Existing Score Distillation Methods

We analyze SDS and its variants through our framework by inspecting each component in the computed gradient. For notation, y_{tgt} is the text prompt representing the target distribution, and $\varnothing$ denotes the unconditional prompt. For each method, we present its gradient update and discuss its implications.

Score Distillation Sampling [48]:

$$\epsilon_{\text{SDS}} = \epsilon_\phi\left(\mathbf{x}_{\theta,t}; \varnothing, t\right) + s \cdot \left(\epsilon_\phi\left(\mathbf{x}_{\theta,t}; y_{\text{tgt}}, t\right) - \epsilon_\phi\left(\mathbf{x}_{\theta,t}; \varnothing, t\right)\right) - \epsilon,$$

where s is the strength of classifier-free guidance. When s is small, the ϵ functions as an averaging term to regress the image to the mean. However, the SDS gradient has been shown to work best with extreme values of classifier-free guidance s like 100. We can rewrite the gradient to emphasize how the conditional-unconditional delta dominates at high CFG scales.

$$\epsilon_{\text{SDS}} = \underbrace{s \cdot \left(\epsilon_\phi\left(\mathbf{x}_{\theta,t}; y_{\text{tgt}}, t\right) - \epsilon_\phi\left(\mathbf{x}_{\theta,t}; \varnothing, t\right)\right)}_{\text{Dominant when } s \gg 1} + \epsilon_\phi\left(\mathbf{x}_{\theta,t}; \varnothing, t\right) - \epsilon,$$

Experimentally, we produce very similar results at high CFG with or without the non-dominant terms. We argue that SDS should be interpreted through the dominant term, which fits within our analysis. Under this interpretation, the unconditional direction $\phi\left(\mathbf{x}_{\theta,t}; \varnothing, t\right)$ approximates the source distribution of $\mathbf{x}_\theta$ poorly, instead representing images of any identity with low contrast and geometric artifacts. Figure 1(b) illustrates the effect of a poor approximation. The bridge from the unconditional to conditional distribution leads to the characteristic oversaturation and smoothing of SDS results.

Delta Distillation Sampling [20]:

$$\epsilon_{\text{DDS}} = \epsilon_\phi\left(\mathbf{x}_{\theta,t}; y_{\text{tgt}}, t\right) - \epsilon_\phi\left(\mathbf{x}_{\text{ref},t}; y_{\text{src}}, t\right),$$

where $\mathbf{x}_{\text{ref},t}$ is a noised version of a reference image in the image editing task. As shown in Figure 2 (b), this increases the *source distribution mismatch* since $\epsilon_{\phi,\text{src}}$ is not calculated based on the current optimized image $\mathbf{x}_{\theta,t}$.

Noise Free Score Distillation [28]:

$$\epsilon_{\text{NFSD}} = \left(\epsilon_\phi\left(\mathbf{x}_{\theta,t}; \varnothing, t\right) - (t < 0.2) \cdot \epsilon_\phi\left(\mathbf{x}_{\theta,t}; y_{\text{neg}}, t\right)\right) + s \cdot \left(\epsilon_\phi\left(\mathbf{x}_{\theta,t}; y_{\text{tgt}}, t\right) - \epsilon_\phi\left(\mathbf{x}_{\theta,t}; \varnothing, t\right)\right),$$

https://doi.org/10.52202/079017-1064

where the strength of classifier-free guidance s is set to 7.5 and y_{neg} ="unrealistic, blurry, low quality ...". NFSD greatly reduces the guidance strength while it is observed to perform very similarly to SDS in practice. We can better explain this phenomenon since the prompt y_{neg} does not accurately describe the source distribution as it omits the image's content. In addition, the second component with weight $s = 7.5$ still forms the major part of the gradient, which is the dominant term in SDS.

Classifier Score Distillation [76]:

$$\epsilon_{\text{CSD}} = w_1 \cdot (\epsilon_\phi(\mathbf{x}_{\theta,t}; y_{\text{tgt}}, t) - \epsilon_\phi(\mathbf{x}_{\theta,t}; \varnothing, t)) + w_2 \cdot (\epsilon_\phi(\mathbf{x}_{\theta,t}; \varnothing, t) - \epsilon_\phi(\mathbf{x}_{\theta,t}; y_{\text{src}}, t)),$$

where w_1 and w_2 are hyperparameters. As shown in Figure 2 (c), the second term approximates the bridge from the source distribution to the unconditional distribution, which is not ideal since it does not point to the target distribution. It explains the observation made by the authors [76] that this undermines the alignment with the text prompt. Therefore, the authors always anneal w_2 to 0 during the optimization. However, we show this often reintroduces the SDS artifacts in practice.

Variational Score Distillation [70, 32]:

$$\epsilon_{\text{VSD}} = \epsilon_\phi(\mathbf{x}_{\theta,t}; \varnothing, t) + s \cdot (\epsilon_\phi(\mathbf{x}_{\theta,t}; y_{\text{tgt}}, t) - \epsilon_\phi(\mathbf{x}_{\theta,t}; \varnothing, t)) - \epsilon_{LoRA}(\mathbf{x}_{\theta,t}; y_{\text{tgt}}, t).$$

Out of all the discussed methods, VSD attempts to minimize the *source distribution mismatch* error most directly by test-time finetuning a copy of the diffusion model with LoRA on the current set of $\mathbf{x}_\theta$. Note that in the original paper, the use of LoRA was motivated based on a particle-based variational framework. Our analysis enables an alternative understanding of VSD. As shown in Figure 2 a), this approach is well-justified in our dual diffusion bridge framework. However, training a LoRA *every iteration* is computationally expensive, adds complexity, and introduces its own low-rank approximation errors. Given this insight, we propose a simple yet efficient approach to mitigating source distribution without LoRA.

3.4 Mitigating Source Distribution Mismatch with Textual Descriptions

Our analysis reveals that the LoRA model in VSD most closely approximates the distribution of the current optimized parametrized image, addressing the distribution mismatch error. Unfortunately, it incurs $200 - 300\%$ runtime overhead on top of SDS, making it impractical, despite its significant performance gains. With this understanding, we propose a simple approach that better expresses the source distribution. Our insight is that pre-trained diffusion models have learned the distribution of natural and corrupted images through a combination of powerful text representation and enormous image-caption datasets. We find that by simply describing image corruptions with a text prompt, we can improve our estimate of the source distribution.

Specifically, we propose to use the gradient

$$\epsilon_{\text{ours}} = w \cdot (\epsilon_\phi(\mathbf{x}_{\theta,t}; y_{\text{tgt}}, t) - \epsilon_\phi(\mathbf{x}_{\theta,t}; y_{\text{src}}, t)),$$

where we get y_{src} by adding descriptions of the current image distribution to y_{tgt} (the base prompt). The remaining question is how to set this description. In generation tasks, we propose a simple two-stage solution.

1. We use ϵ_{SDS} to produce a generation with the method's characteristic artifacts:
2. We switch to optimization with our gradient, ϵ_{ours}, to transport the image parameter toward the natural image distribution.

To describe the artifacts produced by SDS, we append the descriptors ", `oversaturated, smooth, pixelated, cartoon, foggy, hazy, blurry, bad structure, noisy, malformed`" and drop the descriptors of the high-quality generation. This description y_{src} does not require hand-crafting based on problem domains—it is fixed across all shown examples and use cases. As shown in Appendix Figure A5, we explored searching for other prompts but did not find that variations in these descriptions made a big difference.

In editing tasks, we have an initialization that y_{src} describes accurately. In such cases, we omit the first SDS stage and only apply our gradient to optimization. We also append a "domain descriptor." For instance, in painting-to-real, this is simply ", painting" to represent the initial distribution.

Figure 3: **Text-to-image generation results with COCO Captions.** We compare different score distillation methods for generating images with COCO captions by optimizing a randomly initialized image. DDIM sampling indicates the lower bound that the diffusion model can achieve. VSD [70] and our method generate the least color artifacts while ours is more efficient than VSD.

Table 1: **Zero-shot FID comparison with different score distillation methods.** We report FID scores of text-to-image generation using 5K captions randomly sampled from the COCO dataset. The best score distillation result is indicated in **bold**, while the second best is underlined.

	DDIM (lower bound)	SDS [48]	NFSD [28]	CSD [76]	VSD [70]	Ours
Zero-Shot FID (↓)	49.12	86.02	91.70	89.96	**59.22**	67.89
Zero-Shot CLIP FID (↓)	16.56	28.39	29.25	27.07	**18.86**	20.31
Time per Sample (mins)	0.05	**4.48**	7.20	6.21	16.02	**4.48**

While the use of such negative prompting has been explored before, such as in NFSD, our analysis motivates a principled way to incorporate it into score distillation. We find that these simple modifications significantly narrow the quality gap between SDS and resource-intensive methods like VSD. We verify this finding experimentally with qualitative results and quantitative comparisons across applicable tasks.

4 Experiments

In this section, we test our proposed method on several generation problems where SDS is adopted. We compare against SDS and other task-specific baselines. Note that our goal is not to show another state-of-the-art text-to-3D generation method, but to verify our findings, where the proposed score distillation approach based on textual description efficiently improves the results by mitigating the source distribution mismatch error. We first perform a thorough experiment in a controlled setting on text-to-image generation. Then, we compare it on text-guided NeRF optimization to SDS and VSD and evaluate the painting-to-real image translation task against image editing baselines. Please see more results in the appendix, including additional qualitative results and comparison, ablation studies and our method's application to optical illusion generation and 3D-sketch-to-real task.

4.1 Zero-Shot Text-to-Image Generation with Score Distillation

To verify our analysis of existing SDS variants and the proposed method, we perform text-to-image generation by optimizing an image of size $64 \times 64 \times 4$ in the Stable Diffusion latent space [70, 28] (We explore other base models like MVDream [57] and SDXL [47] in Appendix Figure A3). The benefit of choosing image generation as the evaluation task is that its generation quality has the least confounding variables among other tasks. (*e.g.,* in text-to-3D, many designs like regularizations [77], initialization [35], 3D representations [9, 67, 74, 66], and 2D prior models [57, 40, 39, 49, 78] could affect the final quality.)

https://doi.org/10.52202/079017-1064

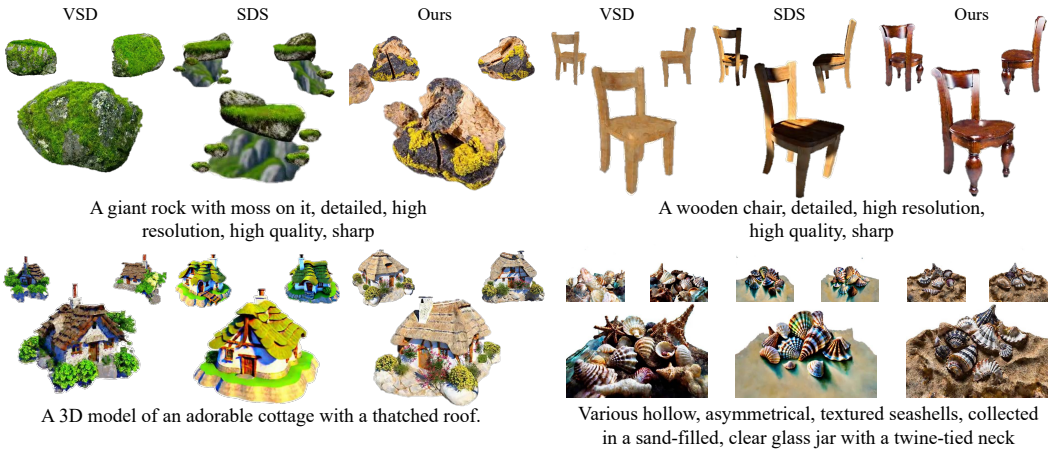


Figure 4: **Text-guided NeRF optimization with different score distillation methods.** We make a fair comparison of SDS and VSD for text-to-3D generation. For each generation, we show three uniformly sampled views. SDS results like the cottage and pepper mill still suffer from over-saturation problems, while ours and VSD can produce realistic details, color, and texture.

We use the MS-COCO [36] dataset for the evaluation. Consistent with the prior study [3], we randomly sample 5K captions from the COCO validation set as conditions for generating images. For each caption, we optimize a randomly initialized the image with the score distillation gradients. We compare our method with several SDS variants including SDS [48], NFSD [28], CSD [76], and VSD [70]. For all the methods, we use the same learning rate of 0.01 and optimize for $2,500$ steps where we generally observe convergence. We compute the zero-shot FID [21] and CLIP FID scores [31] between these generated images and the ground truth images. We also report results generated by DDIM with 20 steps as a lower bound for renference.

We report the FID scores and the time to optimize one image in Table 1. Among all the score distillation methods, VSD [70] achieves the lowest FID scores. However, it requires training a LoRA along the optimization process. Instead, ours achieves a comparable FID score with over $3\times$ faster speed. We visualize random examples generated by different score distillation methods in Figure 3. We notice that SDS and NSFD suffer from the over-saturation and over-smoothness issues. CDS has slightly fewer color artifacts. VSD and ours generate the samples that most closely resemble the DDIM sampling.

4.2 Text-guided NeRF Optimization

We now evaluate the text-to-3D generation problem, where we intentionally aim to exclude variables that could affect the generation quality other than the score distillation methods. We use the Three-Studio [19] repository to optimize a NeRF with settings tuned for ProlificDreamer stage 1 (NeRF optimization) [70]. Note that we do not perform stages 2 and 3, *i.e*. geometry fine-tuning and texture refinement. Specifically, we initialize the NeRF with the method proposed by Magic3D [35], use the regularization losses on the sparsity and opacity, and optimize for 25K steps. We adopt the native SDS and VSD guidance implementations for comparison. In Appendix Figure A4, we evaluate our methods with additional text-to-3D systems, including Fantasia3D [9], Magic3D [35] and CSD [76].

We first show visual comparisons of different score distillation methods in Figure 4. We notice that SDS tends to generate fewer details, as shown by the rock and chair examples, and sometimes suffers from over-saturation issues, as in 2D, as demonstrated by the cottage and seashell examples. Instead, both VSD and ours can generate highly photo-realistic 3D objects, while ours does not require training a LoRA model and shares a similar computational cost as SDS.

Table 2: **Quantitative comparisons of NeRF optimization**. We measure the average CLIP similarity of rendered views using SDS, VSD and our.

	ViT-L/14	ViT-B/16	ViT-B/32
SDS [48]	0.2811	0.3196	0.3139
VSD [70]	0.2837	0.3292	0.3166
Ours	0.2848	0.3282	0.3148

https://doi.org/10.52202/079017-1064

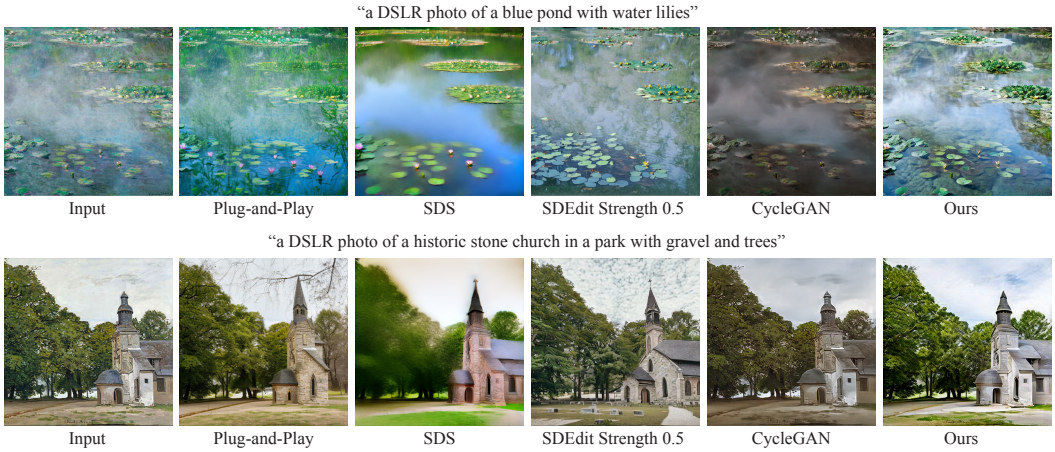


Figure 5: **Painting-to-Real comparison.** We compare our gradient in optimization to image restoration and image-conditional generation baselines. While SDEdit produces convincing textures, it is difficult to find a strength value that balances structure and quality. Other baselines fail to reproduce natural image quality, while our method produces the best combination of quality and faithfulness.

We also perform a quantitative evaluation and user study on the NeRFs optimized based on 31 different text prompts. Note that this number is similar to the choice of existing works on the text-to-3D task [34, 32, 15]. However, different from these works that ignore the confounding 3D variables that contribute to the generation quality, we disentangle this by isolating the score distillation method as the only comparison variable. We follow these works to evaluate the generation quality with CLIP [50]. We report the CLIP similarity in Table 2. Our method consistently outperforms SDS and achieves comparable results with VSD. In addition, in a user study consisting of 37 users, shown pairwise comparisons of rotating 3D renders (*i.e.,* comparisons of our result and a random choice of VSD or SDS, with the prompt: "For a text-to-3D system, given the prompt *[p]*, which result would you be happiest with?"), our results were chosen in 75.7% of all responses.

4.3 Painting-to-Real

We examine our method's ability to serve as a general-purpose realism prior. Paintings are "near-manifold" images, meaning they do not possess natural image statistics but live near the image distribution in image space. An effective image prior should guide a painting toward a nearby natural image through optimization.

We initialize a latent image by encoding scans of the artwork through Stable Diffusion's encoder. We specify a prompt for each painting to condition the diffusion model and then apply the second optimization stage of our method (SDS stage omitted). We experimented with automatically generating prompts via pretrained vision language models but found the results inconsistent, so we leave this to future work. Since the large image datasets used to train diffusion models contain artwork, we append the domain descriptor ", painting" to y_{src} to optimize away from this distribution.

While SDS is proposed to leverage a pretrained text-to-image diffusion model as an image prior, its artifacts make it ineffective in practice. In comparison, our method realistically synthesizes details and relights the image naturally. We observe that SDS methods diverge more easily in 2D experiments than in 3D but that the issue can be mostly resolved with tuning. A future goal is to formulate a gradient that can be applied idempotently [58]. We compare with image reconstruction baselines in Figure 5 and provide a small gallery of painting-to-real results in Figure 6.

5 Discussion on Solving the Linear Approximation Error

As we have shown that reducing the distribution mismatching error can significantly improve the generation quality of the score distillation optimization, it is natural to ask whether one can also reduce the first-order approximation error, induced by linear bridge estimation, to improve the results further. Several recent studies, including SDI [41] and ISM [34], can be viewed as mitigating

https://doi.org/10.52202/079017-1064

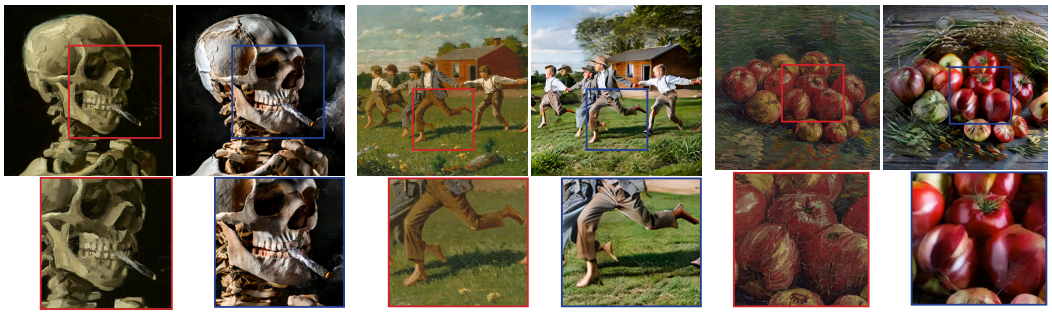


Figure 6: **Painting-to-Real results.**We show selected Painting-to-Real samples with diverse art styles and subjects. Initialization images are shown on the left, optimized images are shown on the right.

Table 3: **Reducing first-order approximation error improves generation quality.** Using full PF-ODE simulation ("Full-path") to replace single-step prediction improves visual quality in all settings.

Approximate the bridge	Single-step			Full-path		
Estimate source distribution	Uncond. (SDS)	Bridge	LoRA (VSD)	Uncond.	Bridge	LoRA
Zero-Shot COCO FID (↓)	86.02	67.89	59.22	63.31	60.07	55.65

this error by replacing the single-step estimation with a multi-step estimation to an intermediate timestep. Under our framework, one can estimate the entire dual bridge by solving both PF-ODE paths. Specifically, via inversion, one can solve the PF-ODE path from $\psi_{0,\text{src}}$ to $x_{\theta,T}$, and then walk to the $\psi_{0,\text{tgt}}$ via sampling. In this way, it is possible to obtain the most accurate gradient direction with little approximation error $\epsilon^*_{\text{SBP}} = w \cdot (\psi_{0,\text{tgt}} - \psi_{0,\text{src}})$. We refer to this approach as "full path". Note that this resolves the linear approximation error, and it is independent of handling the source approximation error, which could be addressed via the discussed text description or LoRA.

However, solving the inversion ODE is not trivial [27]. We noticed that the inversion can exaggerate the distribution mismatch error and cause the optimization to get stuck at a local optimum at the beginning of the optimization. Instead, the stochasticity of the single-step methods often shows more robustness to the input image. Therefore, we first perform the single-step score distillation optimization to obtain reasonable results and then switch to solving the full bridge. We also anneal the timestep endpoint of the bridge throughout the optimization. With this approach, we can now explore addressing both the first and second sources of error. The first source (linear approximation) has "full-path," and the second source (source distribution mismatch error) has "Bridge" or "LoRA". We find that using the "full-path" multi-step (mitigating linear approximation error) always outperforms the single-step methods, achieving a lower FID, as shown in Table 3. However, the same trend does not fully transfer to the text-to-3D experiments. We observe that solving the entire bridge typically introduces additional artifacts and makes the optimization less stable. We leave the best way of leveraging this gradient for future research exploration.

6 Conclusion

We present an analysis that formulates the use of a pre-trained diffusion model in an optimization framework as seeking an optimal transport between two distributions. Under this lens, we analyze SDS variants with a unified framework. We also develop a simple approach based on textual descriptions that work comparably well to the best-performing approach, VSD, without its significant computational burden. However, neither approach has yet to achieve the quality and diversity of images generated by the reverse process. We hope that our analysis enables the development of a more sophisticated solution that can one day achieve the same quality and diversity as the reverse process in an optimization framework. Combining our proposed method with multi-step approximations like ISM [34] or schedules like DreamFlow [32] could mitigate the first-order approximation error and further improve the efficiency, which is an interesting future research direction. With the rise of high-quality video diffusion models, we anticipate that the question of how to effectively use such models as a prior in various problems will become even more important.

Acknowledgment. We thank Matthew Tancik, Jiaming Song, Riley Peterlinz, Ayaan Haque, Ethan Weber, Konpat Preechakul, Amit Kohli and Ben Poole for their helpful feedback and discussion. This project is supported in part by a Google research scholar award, IARPA DOI/IBC No. 140D0423C0035, and NSF grant No. IIS-2213335. The views and conclusions contained herein are those of the authors and do not represent the official policies or endorsements of these institutions.

References

[1] Thiemo Alldieck, Nikos Kolotouros, and Cristian Sminchisescu. Score distillation sampling with learned manifold corrective, 2024.

[2] Sherwin Bahmani, Ivan Skorokhodov, Victor Rong, Gordon Wetzstein, Leonidas Guibas, Peter Wonka, Sergey Tulyakov, Jeong Joon Park, Andrea Tagliasacchi, and David B. Lindell. 4d-fy: Text-to-4d generation using hybrid score distillation sampling. *arXiv preprint arXiv:2311.17984*, 2023.

[3] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Qinsheng Zhang, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, et al. eDiff-I: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022.

[4] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *CVPR*, 2023.

[5] Ryan Burgert, Xiang Li, Abe Leite, Kanchana Ranasinghe, and Michael Ryoo. Diffusion illusions: Hiding images in plain sight, June 2023.

[6] Tianshi Cao, Karsten Kreis, Sanja Fidler, Nicholas Sharp, and KangXue Yin. Texfusion: Synthesizing 3d textures with text-guided image diffusion models. In *ICCV*, 2023.

[7] Dave Zhenyu Chen, Haoxuan Li, Hsin-Ying Lee, Sergey Tulyakov, and Matthias Nießner. Scenetex: High-quality texture synthesis for indoor scenes via diffusion priors. *arXiv preprint arXiv:2311.17261*, 2023.

[8] Dave Zhenyu Chen, Yawar Siddiqui, Hsin-Ying Lee, Sergey Tulyakov, and Matthias Nießner. Text2tex: Text-driven texture synthesis via diffusion models. *arXiv preprint arXiv:2303.11396*, 2023.

[9] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In *ICCV*, 2023.

[10] Tianrong Chen, Guan-Horng Liu, and Evangelos A Theodorou. Likelihood training of schrödinger bridge using forward-backward sdes theory. In *ICLR*, 2022.

[11] Yongxin Chen and Tryphon Georgiou. Stochastic bridges of linear systems. *IEEE Transactions on Automatic Control*, 61(2):526–531, 2016.

[12] Yongxin Chen, Tryphon T. Georgiou, and Michele Pavon. On the relation between optimal transport and schrödinger bridges: A stochastic control viewpoint. *Journal of Optimization Theory and Applications*, 169:671–691, 2014.

[13] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *RSS*, 2023.

[14] Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. In *NeurIPS*, 2021.

[15] Wenqi Dong, Bangbang Yang, Lin Ma, Xiao Liu, Liyuan Cui, Hujun Bao, Yuewen Ma, and Zhaopeng Cui. Coin3d: Controllable and interactive 3d assets generation with proxy-guided conditioning. *arXiv preprint arXiv:2405.08054*, 2024.

 https://doi.org/10.52202/079017-1064

[16] Kevin Frans, Lisa Soros, and Olaf Witkowski. Clipdraw: Exploring text-to-drawing synthesis through language-image encoders. *Advances in Neural Information Processing Systems*, 35:5207–5218, 2022.

[17] Daniel Geng, Inbum Park, and Andrew Owens. Visual anagrams: Generating multi-view optical illusions with diffusion models. *CVPR*, 2024.

[18] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *CVPR*, pages 10696–10706, 2022.

[19] Yuan-Chen Guo, Ying-Tian Liu, Ruizhi Shao, Christian Laforte, Vikram Voleti, Guan Luo, Chia-Hao Chen, Zi-Xin Zou, Chen Wang, Yan-Pei Cao, and Song-Hai Zhang. threestudio: A unified framework for 3d content generation. `https://github.com/threestudio-project/threestudio`, 2023.

[20] Amir Hertz, Kfir Aberman, and Daniel Cohen-Or. Delta denoising score. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2328–2337, 2023.

[21] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017.

[22] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

[23] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[24] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022.

[25] Shir Iluz, Yael Vinker, Amir Hertz, Daniel Berio, Daniel Cohen-Or, and Ariel Shamir. Word-as-image for semantic typography. *SIGGRAPH*, 2023.

[26] Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *CVPR*, pages 867–876, 2022.

[27] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *arXiv preprint arXiv:2206.00364*, 2022.

[28] Oren Katzir, Or Patashnik, Daniel Cohen-Or, and Dani Lischinski. Noise-free score distillation. *arXiv preprint arXiv:2310.17590*, 2023.

[29] Subin Kim, Kyungmin Lee, June Suk Choi, Jongheon Jeong, Kihyuk Sohn, and Jinwoo Shin. Collaborative score distillation for consistent visual editing. In *NeurIPS*, 2023.

[30] Juil Koo, Chanho Park, and Minhyuk Sung. Posterior distillation sampling. *arXiv preprint arXiv:2311.13831*, 2023.

[31] Tuomas Kynkäänniemi, Tero Karras, Miika Aittala, Timo Aila, and Jaakko Lehtinen. The role of imagenet classes in fréchet inception distance. In *ICLR*, 2022.

[32] Kyungmin Lee, Kihyuk Sohn, and Jinwoo Shin. Dreamflow: High-quality text-to-3d generation by approximating probability flow. In *ICLR*, 2024.

[33] Tzu-Mao Li, Michal Lukáč, Michaël Gharbi, and Jonathan Ragan-Kelley. Differentiable vector graphics rasterization for editing and learning. *ACM Transactions on Graphics (TOG)*, 39(6):1–15, 2020.

[34] Yixun Liang, Xin Yang, Jiantao Lin, Haodong Li, Xiaogang Xu, and Yingcong Chen. Luciddreamer: Towards high-fidelity text-to-3d generation via interval score matching. In *CVPR*, 2023.

[35] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *CVPR*, 2023.

[36] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[37] Huan Ling, Seung Wook Kim, Antonio Torralba, Sanja Fidler, and Karsten Kreis. Align your gaussians: Text-to-4d with dynamic 3d gaussians and composed diffusion models, 2024.

[38] Guan-Horng Liu, Arash Vahdat, De-An Huang, Evangelos A Theodorou, Weili Nie, and Anima Anandkumar. I2sb: image-to-image schrödinger bridge. In *ICML*, 2023.

[39] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *CVPR*, 2023.

[40] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. In *ICLR*, 2024.

[41] Artem Lukoianov, Haitz Sáez de Ocáriz Borde, Kristjan Greenewald, Vitor Campagnolo Guizilini, Timur Bagautdinov, Vincent Sitzmann, and Justin Solomon. Score distillation via reparametrized ddim. *arXiv preprint arXiv:2405.15891*, 2024.

[42] Christian Léonard. A survey of the schrödinger problem and some of its connections with optimal transport, 2013.

[43] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. In *CVPR*, pages 12663–12673, 2023.

[44] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[45] Lea Muller, Vickie Ye, Georgios Pavlakos, Michael J. Black, and Angjoo Kanazawa. Generative proxemics: A prior for 3D social interaction from images. 2024.

[46] Hyelin Nam, Gihyun Kwon, Geon Yeong Park, and Jong Chul Ye. Contrastive denoising score for text-guided latent diffusion image editing. *CVPR*, 2022.

[47] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis, 2023.

[48] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *ICLR*, 2023.

[49] Guocheng Qian, Jinjie Mai, Abdullah Hamdi, Jian Ren, Aliaksandr Siarohin, Bing Li, Hsin-Ying Lee, Ivan Skorokhodov, Peter Wonka, Sergey Tulyakov, and Bernard Ghanem. Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors. In *ICLR*, 2024.

[50] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[51] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[52] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

 https://doi.org/10.52202/079017-1064

[53] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, 2021.

[54] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, 2022.

[55] E. Schrödinger. Sur la théorie relativiste de l'électron et l'interprétation de la mécanique quantique. *Annales de l'institut Henri Poincaré*, 2(4):269–310, 1932.

[56] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *NeurIPS*, 2022.

[57] Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. MVDream: Multi-view diffusion for 3d generation. In *ICLR*, 2024.

[58] Assaf Shocher, Amil V Dravid, Yossi Gandelsman, Inbar Mosseri, Michael Rubinstein, and Alexei A Efros. Idempotent generative network. In *ICLR*, 2024.

[59] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data. In *ICLR*, 2023.

[60] Uriel Singer, Shelly Sheynin, Adam Polyak, Oron Ashual, Iurii Makarov, Filippos Kokkinos, Naman Goyal, Andrea Vedaldi, Devi Parikh, Justin Johnson, et al. Text-to-4d dynamic scene generation. *arXiv preprint arXiv:2301.11280*, 2023.

[61] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015.

[62] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021.

[63] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022.

[64] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

[65] Xuan Su, Jiaming Song, Chenlin Meng, and Stefano Ermon. Dual diffusion implicit bridges for image-to-image translation. In *ICLR*, 2022.

[66] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation, 2023.

[67] Christina Tsalicoglou, Fabian Manhardt, Alessio Tonioni, Michael Niemeyer, and Federico Tombari. Textmesh: Generation of realistic 3d meshes from text prompts. *arXiv preprint arXiv:2304.12439*, 2023.

[68] Gefei Wang, Yuling Jiao, Qian Xu, Yang Wang, and Can Yang. Deep generative learning via schrödinger bridge. In *ICML*, 2021.

[69] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A. Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *CVPR*, pages 12619–12629, June 2023.

[70] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *NeurIPS*, 2023.

[71] Tong Wu, Guandao Yang, Zhibing Li, Kai Zhang, Ziwei Liu, Leonidas Guibas, Dahua Lin, and Gordon Wetzstein. Gpt-4v(ision) is a human-aligned evaluator for text-to-3d generation. In *CVPR*, 2024.

[72] Ximing Xing, Chuang Wang, Haitao Zhou, Jing Zhang, Qian Yu, and Dong Xu. Diffsketcher: Text guided vector sketch synthesis through latent diffusion models. In *NeurIPS*, 2023.

[73] Ximing Xing, Haitao Zhou, Chuang Wang, Jing Zhang, Dong Xu, and Qian Yu. Svgdreamer: Text guided svg generation with diffusion model. *arXiv preprint arXiv:2312.16476*, 2023.

[74] Taoran Yi, Jiemin Fang, Junjie Wang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. Gaussiandreamer: Fast generation from text to 3d gaussians by bridging 2d and 3d diffusion models, 2023.

[75] Kim Youwang, Tae-Hyun Oh, and Gerard Pons-Moll. Paint-it: Text-to-texture synthesis via deep convolutional texture map optimization and physically-based rendering. In *CVPR*, 2024.

[76] Xin Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Song-Hai Zhang, and Xiaojuan Qi. Text-to-3d with classifier score distillation, 2023.

[77] Junzhe Zhu and Peiye Zhuang. Hifa: High-fidelity text-to-3d generation with advanced diffusion guidance, 2023.

[78] Zi-Xin Zou, Weihao Cheng, Yan-Pei Cao, Shi-Sheng Huang, Ying Shan, and Song-Hai Zhang. Sparse3d: Distilling multiview-consistent diffusion for object reconstruction from sparse views, 2023.

 https://doi.org/10.52202/079017-1064

Figure A1: **3D sketch-to-real.** We introduce a conditional generation task in 3D where a coarse human-drawn mesh is optimized into a high-quality mesh. While SDS and our gradient both adhere to the prompt and shape conditions, our method produces higher fidelity colors and texture.

Appendix

In this appendix, we discuss the additional experiment details and provide more visual results, including optical illusion sketch, text-based NeRF optimization, and 3D paint-to-real results. We also perform an ablation study of our method.

A Additional Experimental Setup

In this section, we describe our experimental setups in more detail.

Text-to-image generation with score distillation. We use the *stable-diffusion-v2-1-base* model by default for our experiments if not specified. For CSD, we follow the original paper [76] to use $w_1 = w_2 = 40$ at the initialization steps and anneal $w_2 = 0$ within the first 500 steps. We use $s = 100$ for SDS and $s = 7.5$ for NFSD and VSD, which are consistent with the best practice. We use $s = 40$ and $w = 25$ for our method. And we optimize with ϵ_{SDS} loss for 500 iterations and then switch to ϵ_{ours} for the rest of $2,000$ iterations. For all the methods, we use a learning rate of 0.01, and we use a learning rate of $1e-4$ to train the LoRA in VSD.

Text-guided NeRF optimization with score distillation. For our method, we optimize with ϵ_{SDS} loss for $20,000$ iterations and then switch to ϵ_{ours} for the rest of $5,000$ iterations. We use $s = 100$ and $w = 1$ for our method. We find that a high s is necessary to establish geometry in the first stage of the text-to-3D setting, but our method is not too sensitive to this hyperparameter in 2D. We use the rest of the learning rates and regularization strengths as the default settings.

B More Visual Results

In this section, we provide extra visual results. Specifically, we show 3D sketch-to-real and optical illusion generation as additional applications of our method. We also report more comparisons and ablation studies of text-based NeRF optimzition.

B.1 Additional Applications

3D Sketch-to-Real Head-mounted displays with hand tracking are a natural platform for a sort of "3D sketching," where 3D primitives trail from your hand like ink from a pen. The resulting coarse mesh is structurally accurate but lacks geometric or texture detail. To this end, we propose a new application that transfers these 3D sketches to more realistic versions. We extend our text-to-3D solution to generate these details.

We first fit an implicit SDF volume to multi-view renders of the mesh, then apply our gradient with the same schedule as in text-based NeRF optimization. We lower the learning rate for geometry parameters to prevent divergence from the guiding sketch. Holding other hyperparameters equal, we compare our gradient and the SDS gradient in Figure A1.

https://doi.org/10.52202/079017-1064

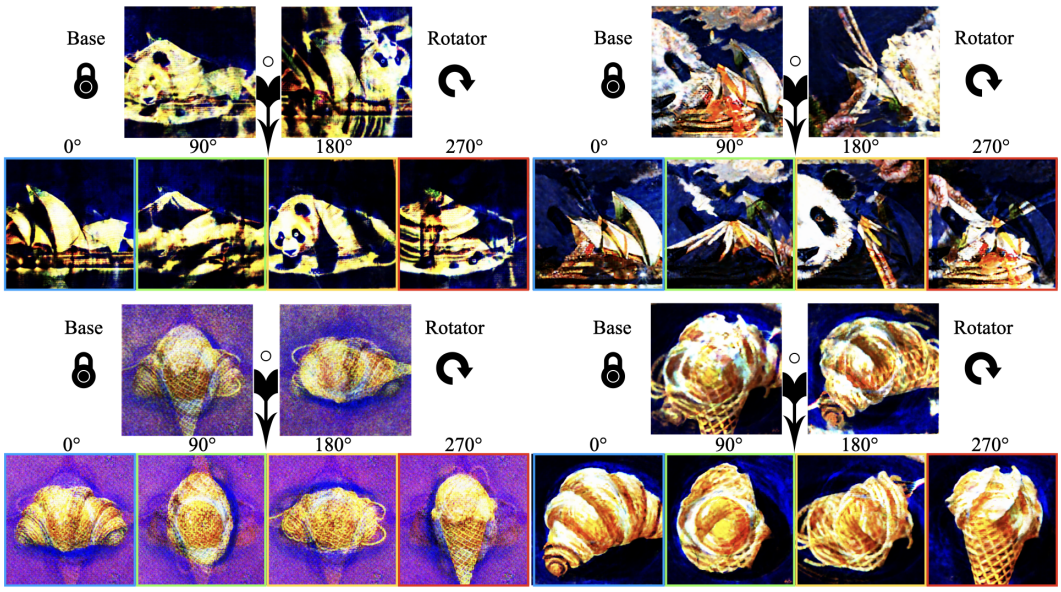
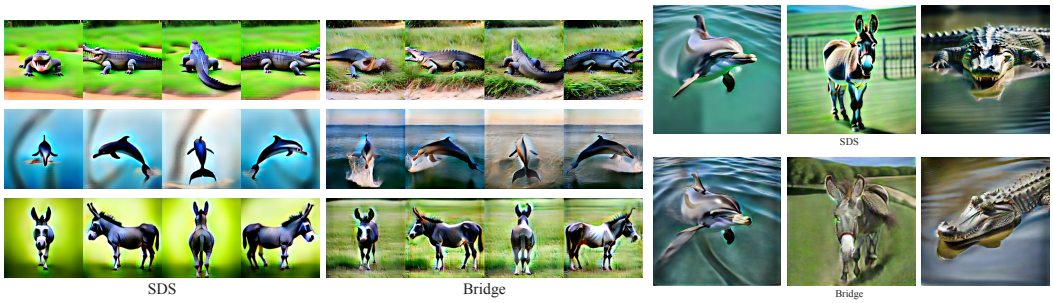


Figure A2: **Diffusion illusions.** We generate overlaid optic illusions with SDS and our method. While SDS suffers from color artifacts, our methods produce more details and proper color.



(a) MVDream [57] (b) SDXL [47]

Figure A3: **Comparison of SDS and ours with MVDream [57] and SDXL [47].** We compare SDS with our two-stage process in two new settings (MVDream and SDXL). The two-stage process produces more natural colors and realistic details.

Illusion Generation. Prior works have shown that diffusion models can be leveraged to generate optical illusions [17, 5]. In these settings, the same image looks semantically different when transformed. To use the diffusion model sampling process, a previous study shows that the transformation has to be orthogonal [17]. However, there remain interesting illusions that are not formed by orthogonal transformation. One such is the rotation overlays. Given a base and a rotator image, by composing the base image with the rotator image at different angles, rotation overlays use two images to display four images. As such composition is not defined by an orthogonal matrix, the existing method [5] employs SDS to optimize the base and rotator images. Such a method suffers from the over-saturation problem, as shown in Figure A2. We show that our method can generate such optical illusions with better visual quality.

B.2 Additional Qualitative Results

Additional text-to-image results. We explore our proposed method across different base models in text-to-image experiments, including MVDream [57] and SDXL [47]. Since MVDream denoises

https://doi.org/10.52202/079017-1064

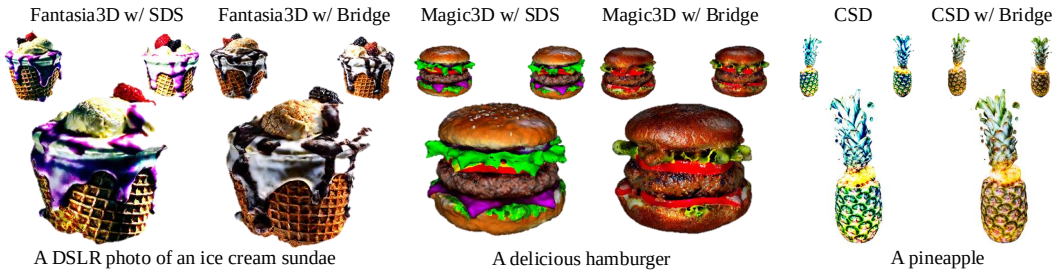


Figure A4: **Comparison with more text-to-3D baselines.** We apply our two-stage optimization as a drop-in replacement of SDS in Fantasia3D [9], Magic3D [35] and CSD [76] for texture refinement. We notice that this change greatly improves details and visual quality and reduces SDS artifacts.

Figure A5: **Ablation study of negative prompts.** We compare SDS results with those from the two-stage optimization (bridge) using our negative prompts and five sets of negative prompts generated by GPT (GPT 1 is the first set, GPT 2 second, GPT 3 third, etc.). All negative prompts produce similar results and outperform the SDS baseline.

four camera-conditioned images jointly, we treat the canvas of four images as a single optimization variable for the SDS gradient. In Figure A3, we compare the SDS baseline to the proposed two-stage optimization, in which we generate more natural colors and detail. This is especially noticeable in the background around the crocodile and donkey.

Additional text-guided NeRF optimization results. For text-guided NeRF optimization comparison against baselines, we show more results in Fig. A7. We test on the prompts used in the original paper [70] and additional prompts [71] that we find to be challenging. We notice that SDS often suffers from over-saturation problems. Our method does not require training a LoRA while it can still improve SDS by getting rid of the color artifacts and generating more details.

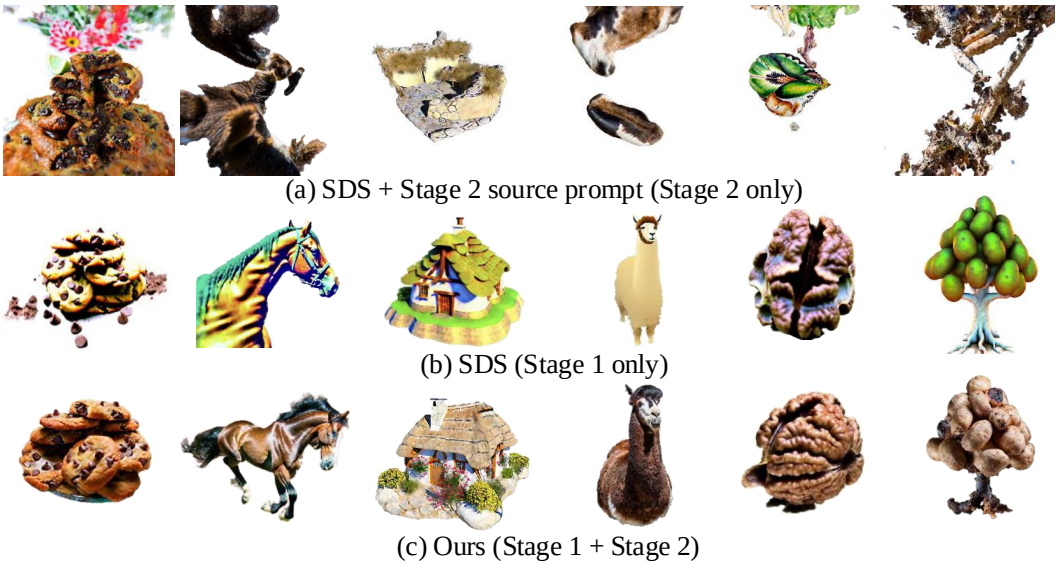
We also perform comparisons with more competitive baselines. We test with Fantasia3D [9], Magic3D [35], and CSD [76] through a drop-in replacement of SDS with our method. Specifically, all three methods optimize a textured DMTet, which is initialized from an SDS-optimized NeRF, using SDS or CSD for 5k or 10k iterations. We replace the SDS or CSD stage of these approaches with the two-stage optimization motivated by our framework. Just like our text-to-3D NeRF experiment, we perform the first stage for 60% of iterations and the second stage for 40% of iterations. Note that we keep all the other hyperparameters the same, which were tuned for the baselines, not our method. This replacement leads to the same optimization time as the original methods. For Fantaisia3D and Magic3D, we use threestudio for fair comparison (Magic3D does not have code available) and the default prompts, which are generally believed to work the best with this reimplementation. For CSD, we use the official implementation. As shown in Figure A4, our method improves the visual quality of all the methods by reducing the oversaturated artifacts of SDS and improving the details.

B.3 Ablation Study

Ablation study of the negative prompts. We explore how the choice of negative prompts in our proposed methods affects the optimization. We prompted GPT-4 through ChatGPT a single time to generate alternative negative prompts using the following:

Here's a set of "negative prompts" to append to a text-to-image prompt that describe undesirable image characteristics: ", oversaturated, smooth, pixelated, cartoon, foggy, hazy, blurry, bad structure, noisy, malformed" I want to try a variety of them, please brainstorm many of roughly the same length.

We produce five variants through these methods as the alternative negative prompts:



(a) SDS + Stage 2 source prompt (Stage 2 only)

(b) SDS (Stage 1 only)

(c) Ours (Stage 1 + Stage 2)

Figure A6: **Ablation study of our method without stage 1.** We show directly optimizing with y_{src} from the start can undermine the quality of the geometry and produce unnecessary content.

1. ", washed out, grainy, distorted, flat, smeared, overexposed, undefined, choppy, glitchy, dull"
2. ", low contrast, jumbled, faint, abstract, over-sharpened, muddy, cluttered, vague, jagged, poor detail"
3. ", soft focus, muffled, streaky, patchy, ghosted, murky, unbalanced, skewed, mismatched, overcrowded"
4. ", overbright, scrambled, bleary, blocky, misshapen, uneven, fragmented, obscured, chaotic, messy"
5. ", dull tones, compressed, smeary, out of focus, unrefined, lopsided, erratic, irregular, spotty, stark"

We keep other hyperparameters identical and only ablate the negative prompts with the variations. As shown in Figure A5, we do not see obvious differences between our prompts and the variants.

Ablation study of stage 2. Instead of switching to stage 2 during the optimization process, we ablate with starting without any SDS optimization from the beginning. That is, we always use the y_{src} with the descriptors ", `oversaturated, smooth, pixelated, cartoon, foggy, hazy, blurry, bad structure, noisy, malformed`". As shown in Figure A6, this makes it hard to generate the proper geometry even though the local texture looks reasonable and is inclined to produce excessive details that are not described by the texts. We suspect that this is because using y_{src} increases the mismatching error at the beginning of the optimization process when the initialization does not resemble the target prompt at all.

C Potential Social Impact

We analyze how to use a pre-trained image diffusion as a prior in an optimization setup, necessary for domains such as 3D. On the positive side, these models can empower individuals to make 3D content creation more accessibly without requiring specialized skills. Additionally, professional artists and designers could rapidly prototype and visualize their ideas, accelerating the creative process. On the negative side, the ease of generating visual content could facilitate the spread of misinformation, proliferate biases in the training set and enable the usage of generated content for malicious purposes. In addition, there are ethical concerns regarding the potential for job displacement in industries reliant on traditional art-making skills and the copyright issues appeared in the training dataset.

https://doi.org/10.52202/079017-1064

Figure A7: **Additional comparison of text-guided NeRF optimization.** We show more examples to compare with different distillation methods, SDS and VSD.

https://doi.org/10.52202/079017-1064

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes, and we also summarized the contributions of the paper at the end of the introduction, which is supported by our analysis in the method section and results in the experiment section.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations in the conclusion section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

https://doi.org/10.52202/079017-1064

Answer: [NA]

Justification: Our theoretical result of score distillation as an optimization that approximates a Schrödinger Bridge path is obvious once we connect the score distillation formula with the Dual Diffusion Implicit Bridge. We also provide an intuitive error analysis of existing SDS variants.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide the details of experiments and will release the code to reproduce the results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: We will release the code with an opensource license when the paper is published.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We mention these details in the experiment section of the main paper and the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Most of our experiments are too expensive for us to run multiple rounds. For example, each run of our text-to-image generation with baseline VSD takes 1.3K GPU hours.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

https://doi.org/10.52202/079017-1064

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide these details in the experiment section as well as Table 1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We briefly discuss this in the conclusion section.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

https://doi.org/10.52202/079017-1064

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: We use human evaluation for our text-to-3D experiment, and we include the details about how the experimentation was done.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.