Towards Controllable Sampling and Diverse Score Distillation in Diffusion Models

Yanbo Xu

CMU-RI-TR-25-27

May 6

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Professor Shubham Tulsiani, *chair*
Professor Deva K. Ramanan
Professor Jun-Yan Zhu
Sheng-Yu Wang

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Robotics.

To my parents.

Abstract

Denoising diffusion models have emerged as a powerful paradigm for generative modeling, which has been widely used for perception, generation, and action. These models can be utilized through sampling or score distillation; however, existing methods lack controllability in sampling and suffer from limited diversity in score distillation. In this thesis, we propose two complementary mechanisms to enhance the controllability and diversity of diffusion-based generation. First, we introduce a score rescaling approach that enables users to steer the sampling diversity of diffusion models without requiring any modifications to training. This method, validated across diverse tasks—including pose estimation, depth prediction, image generation, and robotic manipulation—demonstrates that adjusting the sampling distribution can lead to significant performance improvements. Second, we address the inherent mode-seeking limitation in score distillation for 3D optimization. Inspired by the diffusion sampling process, we propose a novel formulation that encourages optimization to follow diverse generation paths, thereby improving sample diversity while maintaining fidelity. We further introduce an approximation to adapt this formulation to practical settings where generation trajectories cannot be strictly preserved. We empirically validate our approach across multiple applications, including 2D optimization, text-to-3D generation, and single-view reconstruction. Together, these contributions advance the flexibility and effectiveness of diffusion models, broadening their applicability in generative modeling and beyond.

Acknowledgments

These two years at CMU have not been easy, but they have been transformative. I have grown not only as a researcher, but also as a person. None of this would have been possible without the support and encouragement of my advisor, labmates, friends and families.

First and foremost, I would like to thank my advisor, Prof. Shubham Tulsiani, for giving me the freedom to pursue research that genuinely interests me, and for patiently guiding me throughout the process, and for all his kindness and help during my master journey. Beyond resarch problems, I deeply appreciate his mentorship in essential research skills such as presenting ideas clearly, formulating meaningful problems, and thinking critically. His insight and research taste have had a profound impact on me. The valuable lessons I've learned from him will continue to shape my development as a researcher throughout my career.

I am also grateful for the many discussions and collaborations with my labmates and peers in Smith Hall. I especially thank Bharath, Sheng-Yu, Yufei, Jeff, and Qitao for the stimulating conversations and support. Homonga and Yehonathan have broadened my perspective through their deep knowledge in their respective domains. I also sincerely thank Himangi for her thoughtful feedback on my Statement of Purpose. I have benefited greatly from my collaborations with Lucas, Sunjae, their input and energy made our work both productive and enjoyable.

Many of these people have been more than labmates—they are friends. I'm especially grateful to my two roommates, Wei and Zihan, for making life in Pittsburgh more comfortable and fun. I've cherished the hotpot dinners, karaoke nights, foosball, and billiards sessions with Lucas, Zihan, Qitao, and Sunjae, fun road trip with Z, Hanzhe, and the exciting Catan nights hosted by Jiashun. I'll always remember the guitar jam sessions with Sheng-Yu, and the badminton games with Mingxun, Qi, Weichen, Harry and Yufei. I also treasure the memorable trips and gatherings with my friends from HKUST (especially Benran, Junkai, Xiang, Ziruo, Suixin), and the wonderful New Year's party hosted by Yufei and Jialu (and of course, the cat Xinger). Smith Hall itself may not be the most lively building, but conversations with friends like Anish and Neham made it brighter. Thank you all for adding color and joy to my life at CMU.

Lastly, I want to thank my parents and family for their unwavering support, encouragement, and understanding. I am forever grateful for their belief in me, and for raising me to be the person I am today. I love you all.

Contents

When this dissertation is viewed as a PDF, the page header is a link to this Table of Contents.

List of Figures

List of Tables

Chapter 1

Parametric Score Rescaling for Steering Sampling Diversity

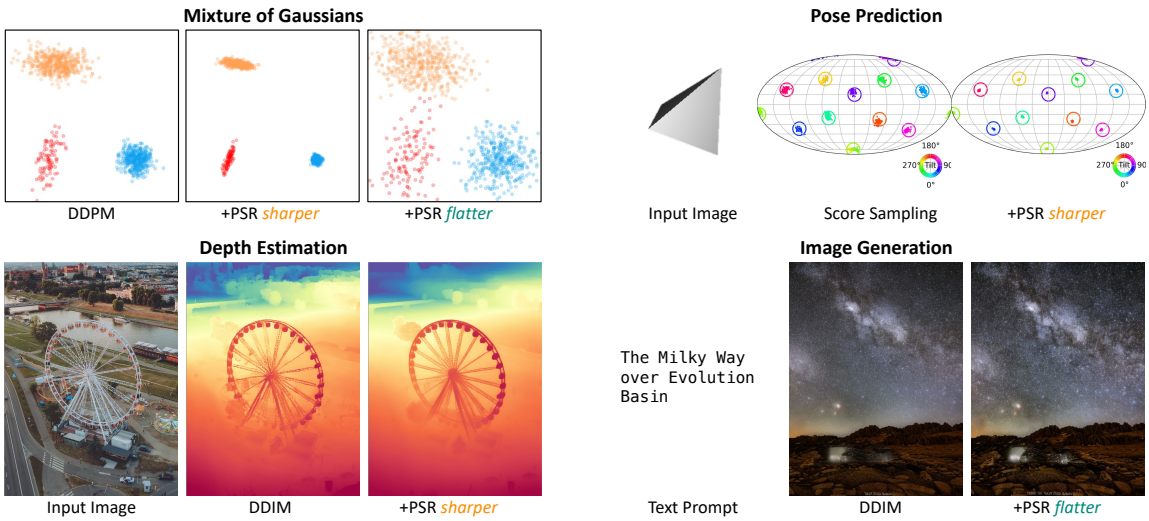

Figure 1.1: We present **Parametric Score Rescaling** (PSR), an approach to steer the sampling distribution in a diffusion model to be sharper or flatter than the training distribution (top-left). We use PSR to modify the sampling of pre-trained diffusion models across different tasks (pose prediction, depth estimation, and image generation) and find that it yields consistent improvements.

1.1 Introduction

Denoising diffusion models have become ubiquitous across computer vision, enabling applications such as generation, perception, and interaction. Given training data $\{\mathbf{x}^n\}$, they can model the underlying data distribution $p_\theta(\mathbf{x})$ (or $p_\theta(\mathbf{x}|\mathbf{c})$ from training tuples $\{(\mathbf{x}^n, \mathbf{c}^n)\}$). At inference, these models then allow drawing samples $\mathbf{x} \sim p_\theta(\mathbf{x})$, *e.g.,* to generate novel images.

However, in certain applications, we may not want to truly sample the modeled distribution. For example, when predicting depth from RGB input, we may want the more likely estimate(s) as output. In contrast, an artist exploring design choices may want the trained image generative model to yield more diverse samples even if they maybe somewhat less likely in the data. In this work, we ask whether we can 'steer' diffusion models to output more likely (or conversely, more diverse) samples. More specifically, *given any trained diffusion model, can we adapt its sampling process to trade-off sample diversity and likelihood at inference?*

Towards answering this question, we note that a denoising diffusion model learns a family of score functions $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$ corresponding to the data distribution with varying levels of (time-dependent) noise. We study the case of isotropic gaussian data, and derive a 'score rescaling' function (determined by two parameters in addition to the diffusion schedule) that can blur/sharpen the sampling distribution by rescaling the predicted scores from a trained diffusion model. We then analyze the applicability of our 'parametric score rescaling' (`PSR`) formulation for broader settings, demonstrating that it can allow tuning the sampling diversity of generic diffusion models.

We perform experiments to highlight the broad applicability of `PSR`, studying four different applications of denoising diffusion models – pose estimation, depth prediction, image generation, and robot manipulation. Across these applications, we show that `PSR` can improve the performance of pre-trained diffusion models *e.g.,* allowing more precise depth and pose inference, or enabling image generation to better match real data distribution.

Prior Art. We are of course not the first to consider the likelihood-diversity trade-off in sampling generative models. For example, the technique of 'temperature scaling' [6, 10, 39], initially used to calibrate classification networks, is often adopted for steering sampling from auto-regressive models by varying the temperature of the

predicted next token distribution. However, as Shih *et al.* [55] demonstrated, this 'greedy' approach does not represent a temperature scaling of the joint distribution, and presented a technique for fine-tuning autoregressive (and diffusion) models for temperature-scaled inference.

While PSR can be similarly thought of as a temperature-scaling approach for sampling the joint distribution in diffusion models, it *does not require any training/finetuning*, and to our knowledge, PSR is the first such training-free technique for tuning sampling from diffusion models. Although classifier-free guidance (CFG) [16] can have similar effects in certain scenarios, we note that there are some fundamental differences. First, CFG cannot be applied for unconditional diffusion. Even for conditional diffusion, it requires changes to the training procedure (condition dropout when training) and cannot be leveraged for models trained without this protocol. Finally, is not mathematically equivalent to altering the sharpness of the modeled distribution which. Another interesting alternative to CFG suggested by Karras *et al.* [26] is to use a 'bad version of the diffusion model for guidance, and while their formulation does improve generation fidelity, it is not a probabilistically grounded mechanism for steering diversity and also requires multiple copies of a diffusion model.

1.2 Formulation

In this section, we derive a mechanism to steer the sampling process in denoising diffusion, effectively allowing sampling from a broader/narrower version of the distribution learned by a given (pre-trained) diffusion model. We first introduce some notation and review preliminaries of denoising diffusion models. We then formalize the task of 'steerable' sampling and derive our parametric score rescaling formulation.

Notations and Conventions. We denote a diffusion model as ϵ_θ, and assume the model predicts noise *i.e.,* $\hat{\epsilon} = \epsilon_\theta(\mathbf{x}_t, t)$, but note that this is a matter of convenience as different predictions are interchangeable (see supplementary). Finally, while we only discuss unconditional distributions and diffusion models in the text below, our formulation is equally applicable for the conditional setting.

Preliminaries. A denoising diffusion generates samples by reversing a forward

process that adds noise to data $\mathbf{x}_0$:

$$\mathbf{x}_t = \alpha_t \mathbf{x}_0 + \sigma_t \epsilon;\ \epsilon \sim \mathcal{N}(\mathbf{0}, I) \tag{1.1}$$

A diffusion model ϵ_θ can be trained by learning to predict the added noise *i.e.,* minimizing $\mathbb{E}_{\mathbf{x}_0,\epsilon}\|\epsilon_\theta(\mathbf{x}_t, t) - \epsilon\|^2$. Under this training objective, a diffusion model learns to approximate the score function of the (noisy) data:

$$\nabla_{\mathbf{x}_t} \log\ p(\mathbf{x}_t) \approx -\frac{\epsilon_\theta(\mathbf{x}_t, t)}{\sigma_t} \tag{1.2}$$

Given a trained model ϵ_θ, one can obtain samples from the (approximated) data distribution $p(\mathbf{x}_0)$ by following a reverse process that begins with $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, I)$ and iteratively denoises it, for example via DDIM [56] inference:

$$\mathbf{x}_{t-1} = \alpha_{t-1}\frac{\mathbf{x}_t - \sigma_t \epsilon_\theta(\mathbf{x}_t, t)}{\alpha_t} + \sigma_{t-1}\epsilon_\theta(\mathbf{x}_t, t) \tag{1.3}$$

1.2.1 Problem Statement

Given a training dataset $\{\mathbf{x}^n\}$, a denoising diffusion model ϵ_θ can approximate the underlying (unknown) data distribution $p(\mathbf{x}_0)$ and allow generating novel samples. In this work, we ask whether we can alter the sampling process such that the generated samples are not from $p(\mathbf{x}_0)$, but from a 'sharper' or 'flatter' version of it. To formalize this, we assume that the data distribution $p(\mathbf{x}_0)$ can be considered as a mixture of (*an unknown set of*) gaussians (while this is a strong assumption, we show empirically that the resulting approach is broadly applicable across tasks):

$$p(\mathbf{x}_0) \equiv \sum_m w_m \mathcal{N}(\mathbf{x}_0; \mu_m, \Sigma_m)$$

We can then define a family of corresponding 'sharper' or 'flatter' distributions (parametrized by k):

$$\bar{p}_k(\mathbf{x}_0) \equiv \sum_m w_m \mathcal{N}(\mathbf{x}_0; \mu_m, \frac{1}{k}\Sigma_m)$$

Intuitively, $\bar{p}_k(\mathbf{x}_0)$ represents a distribution where the variance near each local mode in the data distribution is scaled by $\frac{1}{k}$, with $k > 1$ leading to a 'sharper' distribution and $k < 1$ a 'flatter' one compared to the original. Using the above definition, we can formalize our problem statement as follows:

> Given a diffusion model ϵ_θ trained to approximate a (unknown) data distribution $p(\mathbf{x}_0)$, can we construct a diffusion model $\bar{\epsilon}_\theta$ that would produce samples from $\bar{p}_k(\mathbf{x}_0)$?

We note that this task formulation is different from a temperature scaling of the joint distribution as the mixture weights are unchanged *i.e.,* we can view our formulation as seeking a mechanism to flatten/sharpen the samples around 'local' modes in a data distribution while preserving the 'global' distribution of the samples.

1.2.2 Parametric Score Rescaling

Toward answering the above question, we observe that just as ϵ_θ approximates the score of the noisy data *i.e.,* $\epsilon_\theta(\mathbf{x}_t, t) \approx -\sigma_t \nabla_{\mathbf{x}_t} \log\ p(\mathbf{x}_t)$ (see Appendix), $\bar{\epsilon}_\theta$ should similarly approximate the score for the (noisy versions of) data sampled from the target distribution $\bar{p}_k(\mathbf{x}_0)$. Specifically, if we consider $\bar{p}_k(\mathbf{x}_t|\mathbf{x}_0) \equiv \mathcal{N}(\alpha_t \mathbf{x}_0, \sigma_t^2 I)$ to represent the forward diffusion process on samples from $\bar{p}_k(\mathbf{x}_0)$, we would expect $\bar{\epsilon}_\theta(\mathbf{x}_t, t) \approx -\sigma_t \nabla_{\mathbf{x}_t} \log\ \bar{p}_k(\mathbf{x}_t)$. This leads us to the key insight that relating the score $\nabla_{\mathbf{x}_t} \log\ p(\mathbf{x}_t)$ to $\nabla_{\mathbf{x}_t} \log\ \bar{p}_k(\mathbf{x}_t)$ would allow deriving $\bar{\epsilon}_\theta(\mathbf{x}_t, t)$ from $\epsilon_\theta(\mathbf{x}_t, t)$. More specifically:

> If there is a linear transformation $\mathcal{T}$ s.t. $\nabla_{\mathbf{x}_t} \log\ \bar{p}_k(\mathbf{x}_t) \equiv \mathcal{T}(\nabla_{\mathbf{x}_t} \log\ p(\mathbf{x}_t))$, then $\bar{\epsilon}_\theta(\mathbf{x}_t, t) \equiv \mathcal{T}(\epsilon_\theta(\mathbf{x}_t, t))$

We show that such a $\mathcal{T}$ (which refer to as a 'score rescaling function') can indeed be derived for a simple scenario, and then analyze the formulation in more generic settings.

Score Rescaling for Isotropic Gaussian Data. We consider data drawn from an isotropic gaussian distribution $\mathbf{x_0} \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2 I)$. Under the forward diffusion process,

the noisy data distribution $p(\mathbf{x_t})$ can also be shown to be a gaussian:

$$p(\mathbf{x_t}) \equiv \mathcal{N}(\alpha_t \boldsymbol{\mu}, (\alpha_t^2 \sigma^2 + \sigma_t^2) I)$$

We can also derive the score under the above distribution:

$$\nabla_{\mathbf{x}_t} \log\ p(\mathbf{x}_t) = -\frac{\mathbf{x}_t - \alpha_t \boldsymbol{\mu}}{\alpha_t^2 \sigma^2 + \sigma_t^2} \tag{1.4}$$

Similarly, if we define a corresponding 'sharper' or 'flatter' data distribution $\bar{p}_k(\mathbf{x}_0) \equiv \mathcal{N}(\boldsymbol{\mu}, \frac{1}{k}\sigma^2 I)$, we can compute its score as follows:

$$\nabla_{\mathbf{x}_t} \log\ \bar{p}_k(\mathbf{x}_t)) = -\frac{\mathbf{x}_t - \alpha_t \boldsymbol{\mu}}{\frac{\alpha_t^2}{k} \sigma^2 + \sigma_t^2} \tag{1.5}$$

Denoting by s_t the signal-to-noise ratio $\frac{\alpha_t^2}{\sigma_t^2}$ in denoising diffusion, we can combine eq. (1.4) and eq. (1.5) to derive $\mathcal{T}$ for isotropic gaussian data $\mathcal{N}(\boldsymbol{\mu}, \sigma^2 I)$:

$$\nabla_{\mathbf{x}_t} \log\ \bar{p}_k(\mathbf{x}_t) = \frac{s_t \sigma^2 + 1}{s_t \frac{\sigma^2}{k} + 1}\ \nabla_{\mathbf{x}_t} \log\ p(\mathbf{x}_t) \tag{1.6}$$

Note that $k = 1.0$ recovers the original diffusion model.

Mixture of Gaussians. While we derived the 'score rescaling' function for a single (isotropic) gaussian, we can show that is also a valid approximation if the data distribution is a mixture of *well-separated* isotropic gaussians *i.e.,* for most points in space, a single mixture component dominates. We analyze this further in the supplementary, but intuitively, the score function in such a scenario can be approximated as the (weighted) score from the closest gaussian, and the above derivation still (approximately) holds.

Although this is not valid for anisotropic gaussians or mixtures of 'nearby' gaussians, we empirically show (in section 1.3) that such a score rescaling still yields the desired flattening/sharpening of the sampling distribution.

Steering Diffusion Inference. We can operationalize the above derivation (equation 1.6) into a simple algorithm for steering the sampling from a pre-trained diffusion model ϵ_θ. Assuming the diffusion forward process has signal-to-noise ratio s_t, we can define 'score rescaling' function via two user-defined parameters k and σ:

Sampling with Parametric Score Rescaling (k, σ)

Given a diffusion model ϵ_θ, substitute its noise prediction with:

$$\bar{\epsilon}_\theta(\mathbf{x}_t, t) = r_t(k, \sigma)\ \epsilon_\theta(\mathbf{x}_t, t),$$

where rescaling factor $r_t(k, \sigma) := \frac{s_t\sigma^2+1}{s_t\frac{\sigma^2}{k}+1}$

We note that this is a one-line change that can be easily integrated into the sampling process in any off-the-shelf diffusion model! We first analyze its effects in sampling simple 2D data and then apply it to diffusion models trained across different tasks.

1.3 Analysis

In this section, we seek to gain insight into the empirical behavior of `PSR` as well as compare it to alternate strategies for tuning sampling sharpness. We do so by analyzing the generated samples when applying `PSR` and other techniques for 'toy' 2D distribution $p(\mathbf{x}_0)$, where we can compare the generated samples to those under the (analytically computed) 'ground-truth' target distribution $\bar{p}_k(\mathbf{x}_0)$.

1.3.1 Setup

We use a mixture of gaussians for 2D distribution $p(\mathbf{x}_0)$. Formally, $p(\mathbf{x}_0) \sim \sum_i \pi_i N(\mu_i, \Sigma_i)$, where π_i represents mixture weight, and μ_i, Σ_i describe each gaussian parameters. Given a mixture of gaussians, we can analytically derive its score function, which we directly use for diffusion steps instead of training a neural network. We compare `PSR` with two other alternatives: (1) temperature scaling [6, 10, 39] in an autoregressive model and (2) naive noise scaling in diffusion [55], explained below.

Temperature Scaling in Autoregressive Model (AR). Autoregressive models the 2D data distribution as $p(x, y) \sim p(x)p(y|x)$, where $\mathbf{x}_0 = (x, y)$. $p(x)$ and $p(y|x)$ are also a gaussian mixture and can be computed analytically. Temperature scaling changes the sampling distribution of autogressive model as $x \sim p^\tau(x)$ and $y \sim p^\tau(y|x)$.

We use ground truth analytical solutions for $p(x)$ and $p(y|x)$ during sampling. The limitation of this approach is that it does not apply temperature scaling on the "joint" distribution but only on the marginal distribution.

Naive Noise Scaling (NNS). DDPM[18] samples data using

$$\mathbf{x}_{t-1} = \alpha_t \frac{\mathbf{x}_t - \sigma_t \epsilon_\theta(\mathbf{x}_t, t)}{\alpha_t} + \sqrt{\sigma_{t-1}^2 - v_t^2}\epsilon_\theta(\mathbf{x}_t, t) + v_t \epsilon$$

, where $v_t = \sigma_{t-1}/\sigma_t \sqrt{1 - \alpha_t^2/\alpha_{t-1}^2}$, and $\epsilon \sim \mathcal{N}(\mathbf{0}, I)$. Naive noise scaling [55] tunes the sampling sharpness by increasing/reducing the variance of noise added at each denoising step of DDPM, i.e. $\hat{\epsilon} = \epsilon/k'$ at each step. Intuitively, by increasing/reducing the variance of noise, the variance of DDPM samples increases/decreases, respectively. We denote the factor of variance change as k'. However, k' has no clear probabilistic interpretation of its effects and is mode seeking.

1.3.2 Results

Isotropic Gaussian Mixture. We first test `PSR` on the simplest setup, where $p(\mathbf{x}_0)$ is a mixture of equally weighted isotropic gaussian with equal variance σ^2. We compare `PSR` with other sampling approaches previously explained. Fig.1.3 presents the results, where the first two columns display $p(\mathbf{x}_0)$ and $\bar{p}_k(\mathbf{x}_0)$, respectively, while the subsequent columns correspond to different sampling methods. When $k = 10.0$, `PSR` successfully reduces the variance of each mode while preserving their weights. In contrast, even in this simple setting, other methods sample non-uniformly from each mode. As previously explained, temperature scaling changes the marginal distribution, causing the weight of the Gaussian with x coordinate not overlapping with others to reduce. The mode-seeking behavior of naive noise scaling also leads to an uneven distribution across modes. Similarly, when $k = 0.5$, `PSR` successfully increases the variance across modes, and although the other methods also similarly increase variance, we do find that the weights from `PSR`are more uniform.

Anisotropic Gaussian Mixture. Under the more generic distribution where $p(\mathbf{x}_0)$ is an uneven mixture of anisotropic Gaussians, σ only approximates the variance of data. Nevertheless, as shown in Fig.1.4, `PSR` consistently creates a sharper distribution when $k > 1$ and a flatter distribution when $k < 1$ across varying values of σ, following

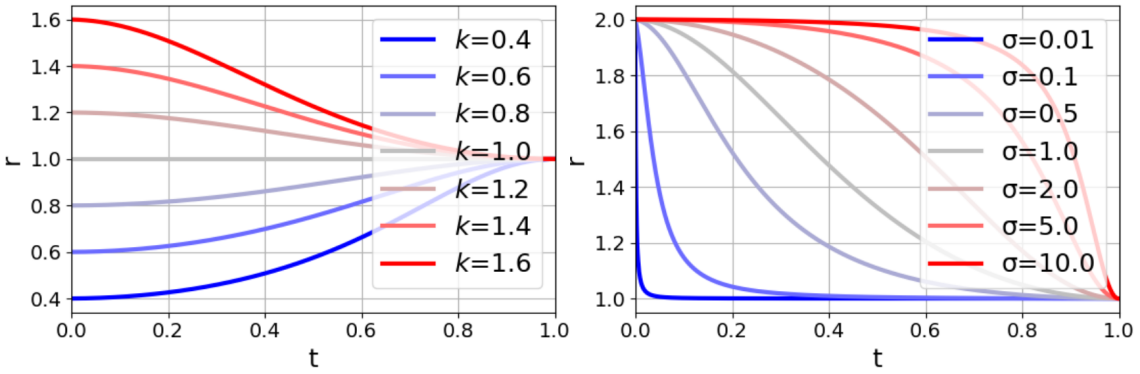

Figure 1.2: **Effect of σ and k on rescaling factor r_t.** Fixed $\sigma = 1.0$, varying k(left). Fixed $k = 2.0$, varying σ(right).

our intention. This shows that even though we derived PSR under isotropic Gaussian data of variance σ, PSR can be leveraged in more diverse scenarios.

1.3.3 Interpreting Rescaling Hyperparameters

Beyond interpreting σ as the variance of the data, and k as a control factor for adjusting variance at each local mode, we provide an alternative interpretation of their roles using the rescaling factor r_t. This becomes particularly crucial when extending PSR to more complex domains, where multiple modes often exhibit varying and anisotropic variance, as demonstrated in the previous experiment. In fig. 1.2, we plot the r_t against k and σ. Note that k now indicates the max/min of the rescaling factor r_t. And as $t \to 0$, signal-to-noise ratio $s_t \to \infty$, $r_t \to k$. Meanwhile, σ indicates how early we want to steer the sampling process. The larger σ, the earlier the sampling is steered. A very small σ let us use the original diffusion sampling ($r_t \approx 1.0$) and only steer the last few denoising steps.

1.4 Applications

We demonstrate the broad applicability and effectiveness of PSR by applying it to a diverse set of real-world applications, spanning perception (pose estimation and depth estimation in section 1.4.1 and 1.4.2), generation (image synthesis in section 1.4.3), and interaction (robotic manipulation in section 1.4.4).

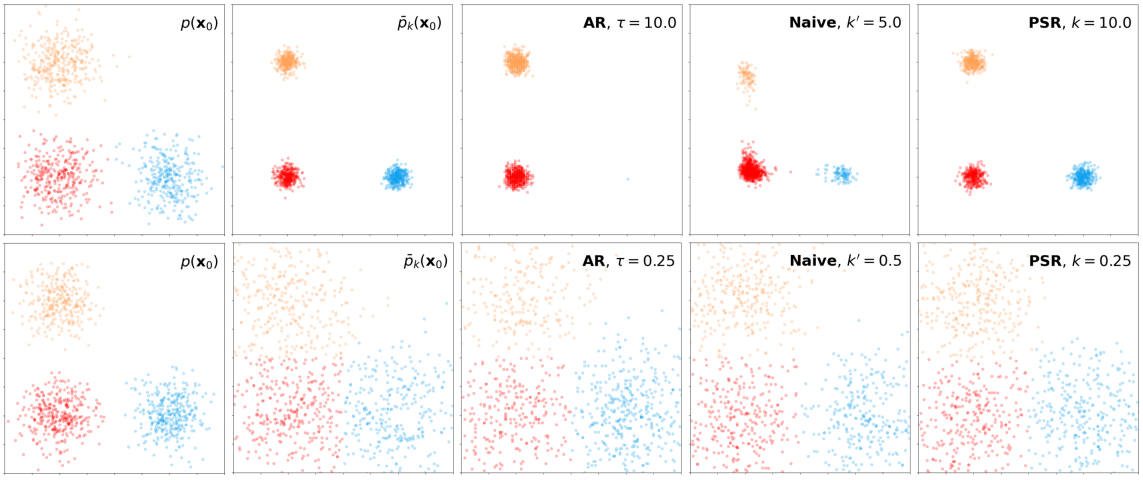

Figure 1.3: **Comparison on uniform mixture of 2D isotropic gaussians. AR** - temperature scaling, **Naive** - naive noise scaling (NNS), **PSR** - ours. When $k = 10.0$ (top), AR and NNS tend to converge to a subset of modes (orange, red) and (red), respectively. When $k = 0.25$(bottom), the trend becomes opposite. AR biases the samples more on the mode it lost at $k = 10.0$(blue), and NNS also focuses on the modes it was less focusing at $k = 10.0$(orange, blue). **PSR** preserves weights uniformly at both $k = 10.0, 0.25$.

1.4.1 Pose Prediction

We first evaluate `PSR` on object pose prediction from a single image. We focus on predicting $SO(3)$ rotations of objects using the SYMSOL dataset [37], which contains geometric shapes with a high order of symmetries. The inherent ambiguities arising from object symmetries necessitate modeling a multi-modal distribution. Previous work [20, 22, 30, 61, 68] show that diffusion models can effectively model and sample from such multi-modal distributions and predict accurate poses. Since better pose predictions require samples close to the ground truth modes, we apply `PSR` to the $SO(3)$ diffusion model following the setup of [20] to sample a sharper distribution using k values greater than 1.

We visualize the effect of `PSR` in fig. 1.5 where we show the sampled poses on an example image from SYMSOL. `PSR` samples poses more concentrated around ground truth modes (the circle centers) than score sampling used in [20]. Moreover, we evaluate `PSR` quantitatively in table 1.1. `PSR` predictions have lower average error and higher accuracy under a range of accuracy thresholds compared to score sampling,

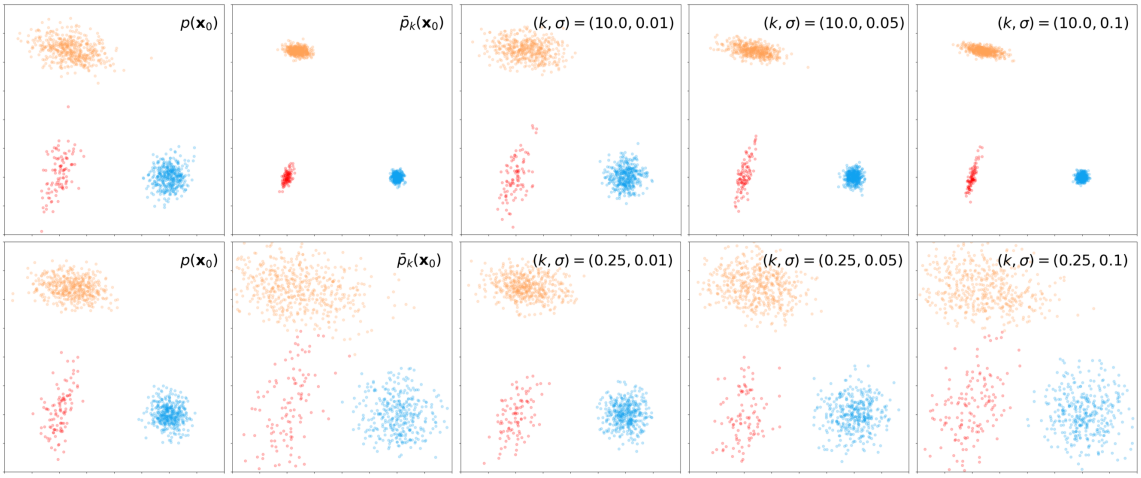

Figure 1.4: **PSR on nonuniform mixture of 2D anisotropic gaussians.** The mixture weight of $p(\mathbf{x}_0)$ is 0.1(red), 0.5(orange), and 0.4(blue). When $k = 10.0$(top), **PSR** consistently makes the distribution sharper, while making the distribution flatter at $k = 0.25$(bottom), across all σ values.

highlighting the benefits of predicting close to modes. Furthermore, we find that naive noise scaling also reduces pose error, achieving a performance slightly better than `PSR` on SYMSOL. This is a strong sign that steering the sampling distribution during test time can, in fact, improve model performance. However, we note that `PSR` remain robust and applicable over many tasks and sampling methods where naive noise scaling is not possible.

Finally, we ablate the mean prediction error over σ and k (see supplementary figure and details). `PSR` consistently reduces the prediction error across a wide range of $k \in (1, 40]$ compared to the baseline ($k = 1$) and reaches the optimal at $k \approx 7$.

1.4.2 Depth Estimation

The task of monocular depth estimation is inherently challenging due to its uncertainty—an object could appear large but be far away, or small but close. Several methods [7, 28, 51] addresses this by using diffusion models. We chose Marigold [28] in our experiment, which fine-tuned a pre-trained text-to-image diffusion model for depth estimation, achieving impressive results. However, individual samples may be suboptimal due to the stochastic nature of diffusion-based sampling and the ambiguity

Table 1.1: **Quantitative results for pose estimation.** We show the mean error of predictions in degrees, and percentage accuracy of prediction within a threshold of 0.1, 0.2, 0.5, and 1 degree. We use $(k, \sigma) = (7.0, 0.5)$ for PSR and $k' = 40$ for naive noise scaling.

	Error (deg) ↓	Acc@ (deg) ↑			
		0.1	0.2	0.5	1.0
Score Sampling	0.444	1.37	9.44	68.33	97.91
+ Naive Noise Scaling(40)	**0.350**	**3.43**	**20.02**	**84.98**	**99.10**
+ Naive Epsilon Scaling(7)	0.357	2.94	18.76	84.56	99.07
+ PSR (7.0, 0.5)	0.356	3.02	18.52	84.05	99.00

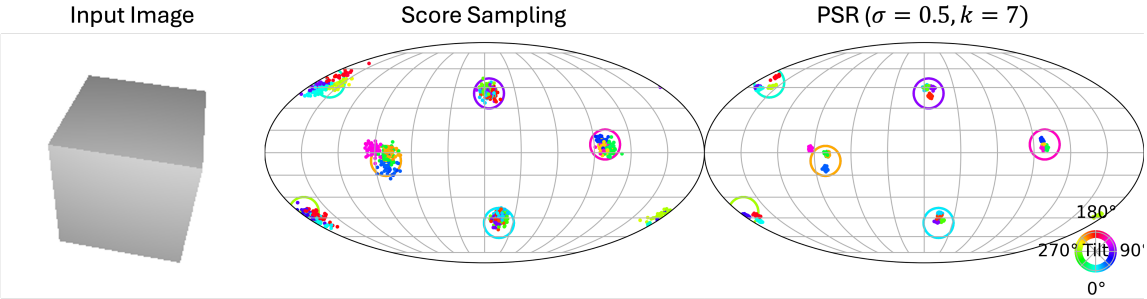

Figure 1.5: **Predicted poses on SYMSOL**. We show predicted poses (right) on an input image (left) and notice PSR reduces prediction error. Each sample is a dot on the sphere, positioned by its first canonical axis, with color indicating rotation. Circles denote ground truth poses. We modify the location of samples to exaggerate error by a factor of 15 to show visual difference given plotting constraints.

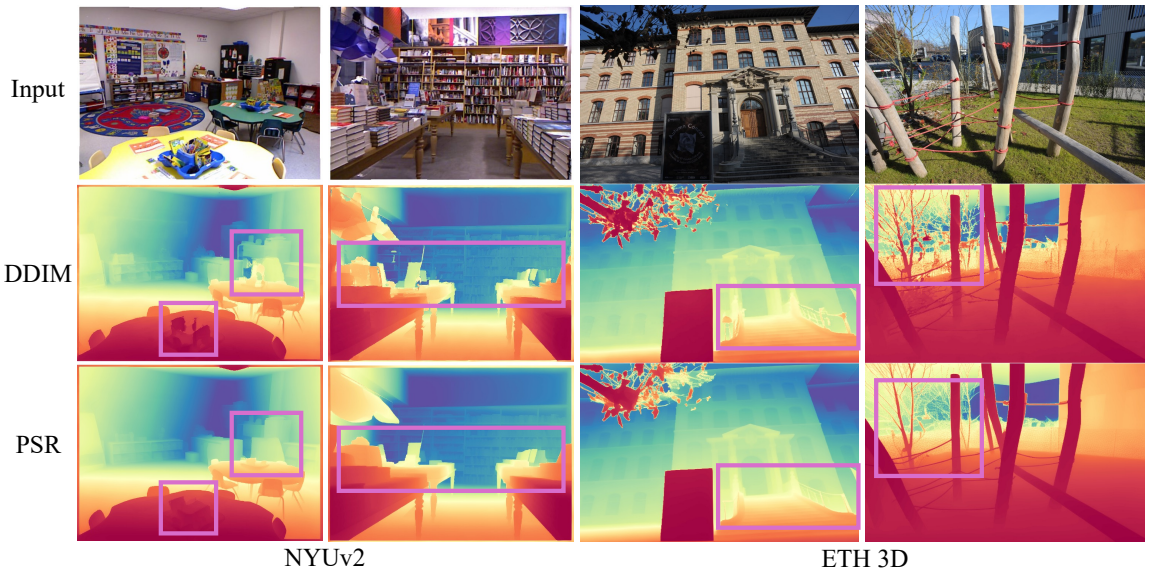

Figure 1.6: **Qualitative Depth Estimation Comparisons.** Compared to DDIM, PSR with $k > 1$ predicts cleaner depth in the regions with high uncertainty (highlighted by pink boxes).

of depth estimation. To mitigate this, Marigold introduces a test-time ensembling method that aggregates multiple samples using a pixel-wise median, significantly improving final performance at the cost of increased computational expense.

As shown in table 1.2, PSR enhances prediction accuracy by sampling from a sharper distribution, ensuring more probable samples given the input image. Notably, PSR is complementary to the ensembling—we achieve similar performance on ETH3D [52] when ensembling with 2 samples compared to the DDIM result with an ensemble size of 10. We also include some qualitative comparisons in fig. 1.6. Note that PSR predicts cleaner depth compared to DDIM, especially in regions with high uncertainty (as highlighted in the figure). We also show the effect of σ and k on the AbsRel metric in supplementary. Compared with the DDIM sample ($k = 1$), PSR demonstrates consistent performance gain in various (k, σ) configurations.

1.4.3 Image Generation

In addition to pose and depth prediction, diffusion models [18, 42, 43, 48, 57]—are among the most powerful and widely used approaches for image generation. We examine the effect of steering the sampling distribution of diffusion models for

Table 1.2: **Quantitative Evaluation of Depth Estimation.** PSR improves the results without additional computation and outperforms the naive baseline. With the ensemble technique, PSR further improves the performance. With an ensemble size of 2, PSR and get compatible results to the optimal performance (DDIM w/ensemble=10). We use $(k, \sigma) = (1.3, 10.0)$ for PSR.

	NYUv2 [38]		ETH3D[52]	
	AbsRel ↓	δ_1 ↑	AbsRel ↓	δ_1 ↑
DDIM	6.0	95.9	7.1	90.4
+ Naive Noise Scaling	5.85	**96.0**	6.82	95.6
+ PSR	**5.84**	**96.0**	**6.68**	**95.7**
+ PSR (w/ ensemble = 2)	5.67	96.2	6.50	95.8
DDIM (w/ ensemble=10)	5.5	96.4	6.5	96.0

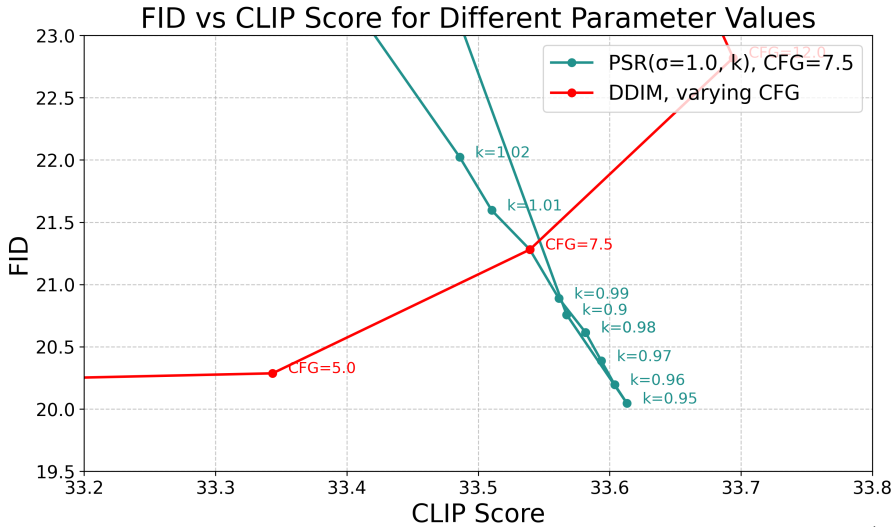

Figure 1.7: **FID vs CLIP Score.** PSR achieves better text-alignment (CLIP) and image fidelity (FID), improving upon the Pareto frontier of CFG tuning, which trades off between FID and CLIP.

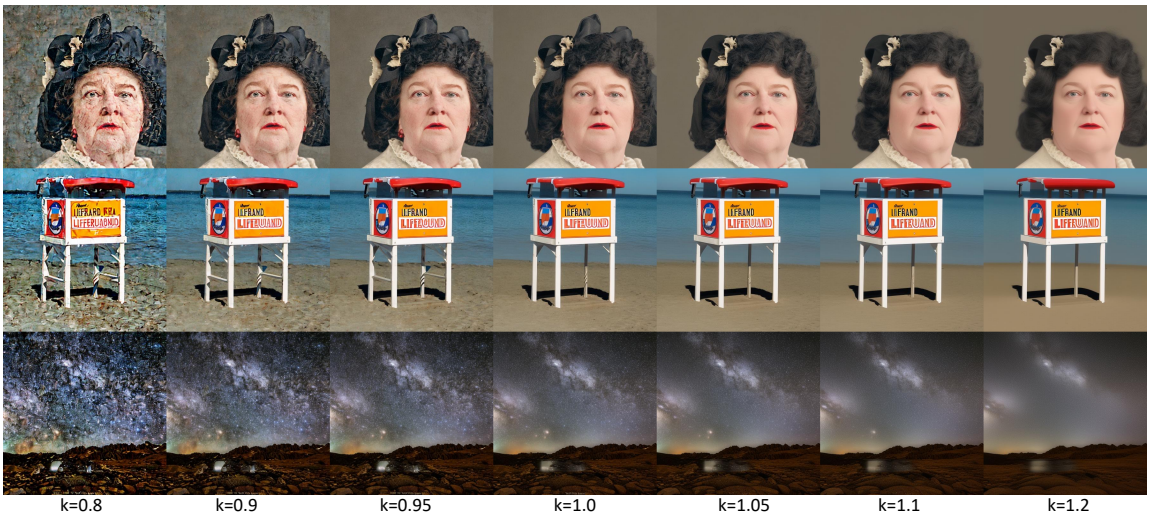

Figure 1.8: **Qualitative Results for Varying K.** PSR allows for tuning the generated outputs to be more diverse and detailed (lower k) or more smooth and likely (higher k). While neither extreme is desirable, we notice a slightly smaller of $k = 0.95$ gives pleasing images with more details.

diversity versus likelihood with PSR in image generation. Unlike pose and depth estimation, which requires high precision, image generation is a more creative task where sampling from flatter distributions helps to recover more pleasing images with more high frequency details. We use Stable Diffusion v2 [48] and evaluate FID [13, 41] and CLIP [46] against a 5k image subset from LAION Aesthetics [53] across different configurations of CFG, PSR parameter k, and sampling methods. We fixed the number of sampling steps to 50. In Fig. 1.7, we see adjusting CFG makes a trade-off between text-alignment and image fidelity—higher CFG increases CLIP score at the cost of worse FID. Meanwhile, PSR allows for additional tuning beyond the Pareto frontier of CFG. Fixing CFG at 7.5, $k = 0.95$ achieves better FID and CLIP score compared to DDIM sampling.

Moreover, we show that PSR is capable of improving both FID and CLIP scores across different diffusion sampling methods in Tab. 1.3. While naive noise scaling with $k' < 1$ also improves upon the DDPM baseline, PSR remains better while being applicable to sampling methods that do not support naive noise scaling, such as DDIM.

Qualitatively, we observe in Fig. 1.8 that lower ks lead to images with more

	FID ↓	CLIP ↑
DDIM	21.28	33.54
+ PSR (0.95, 1.0)	**20.05**	**33.61**
DDPM	22.81	33.66
+ Naive Noise Scaling (0.96)	19.87	33.68
+ Naive Epsilon Scaling (1.01)	21.38	33.53
+ PSR (0.9, 1.0)	**19.57**	**33.77**
EulerDiscrete	22.11	33.54
+ PSR (0.95, 1.0)	**19.95**	**33.61**

Table 1.3: **Results for Image Generation.** PSR improves image fidelity (FID) and text-alignment (CLIP) consistently across multiple sampling methods—DDIM, DDPM, and EulerDiscrete.

high-frequency detail (in the extreme case more noisy), and higher ks lead to a smoother image. We infer that using a smaller k flattens the modeled distribution and allows better coverage of the desirable image space. Overall, our results highlight that the control over the likelihood-diversity trade-off enabled by ours is beneficial in image generation.

1.4.4 Robotic Manipulation

Lastly, we examine the applicability of PSR on predicting robot actions, with a focus on robotic manipulation. In particular, we use the diffusion policy [4, 29, 66, 67] as the diffusion backbone, where a policy is trained for each task. We use checkpoints provided by Chi *et al.* [4], containing both transformer (Diffusion Policy-T) and U-Net (Diffusion Policy-C). It is worth mentioning that the problem is formulated as sequential decision-making, which differs from previous domains. In particular, the policy only models a short-horizon or future actions rather than the full length of the task episode. Hence, we are steering the sampling distribution of the short-horizon actions, rather than a distribution of the long-horizon actions [55], which may cause undesirable effects across the episode.

We choose two complex manipulation tasks (Transport, Tool-Hang) [35, 72] and one simple 2D manipulation task (Push-T) in simulation with state observation, with multi-human data (mh) for transport and proficient human data (ph) for the remaining two. Results are in Table. 1.4. Without any further training, PSR consistently

improves performance, especially when the base policy performance is low. Compared to other domains, optimal k is smaller than 1.0 for some tasks (Transport, Tool Hang) and larger than 1.0 for others (Push T). This is because each base policy has a different level of proficiency. For a nonproficient base policy, sampling diverse actions with $k < 1.0$ tends to help escape the local minimum, while for a proficient base policy, $k > 1.0$ tends to help. Moreover, for certain tasks and models, both $k > 1$ and $k < 1$ show performance gain. As $k > 1$ and $k < 1$ are both steering the sampling process, changing the sampling distribution may help to escape the local minimum, even for $k > 1$. See supplementary for details.

Table 1.4: **Results for Robotic Manipulation.** Success rates are computed across 3 seeds, 3 checkpoints, and 150 different environment initial conditions (1350 episodes in total), where we compute the mean and std over seeds. The results are computed with best (k, σ) and k' for both `PSR` and Naive Noise Scaling.

	Transport	ToolHang	Push-T
Model	mh	ph	ph
DiffusionPolicy-C	63.2 ± 0.3	57.0 ± 0.4	90.1 ± 0.5
+ `PSR`	**66.9 ± 1.4**	63.3 ± 1.3	**90.6 ± 0.004**
+ Naive Noise Scaling	64.6 ± 0.7	58.4 ± 1.1	90.2 ± 0.6
+ Naive Epsilon Scaling	64.3 ± 1.0	**65.9 ± 0.7**	90.2 ± 0.1
DiffusionPolicy-T	41.4 ± 2.7	87.7 ± 1.7	90.2 ± 0.5
+ `PSR`	**45.9 ± 2.4**	**89.3 ± 0.9**	**90.7 ± 0.5**
+ Naive Noise Scaling	45.3 ± 1.9	88.7 ± 0.9	90.3 ± 0.3
+ Naive Epsilon Scaling	43.6 ± 2.2	87.9 ± 1.5	90.3 ± 0.6

1.5 Discussion

We presented `PSR`, an approach to alter the sampling distribution for a pre-trained diffusion model. While we demonstrated its efficacy across several (toy and real) tasks, there are fundamental limitations worth highlighting. First, unlike temperature scaling, `PSR` can only alter the 'local' sampling *e.g.,* `PSR` does not change the weights of the components in a gaussian mixture, only the variance. Moreover, while `PSR` does empirically steer the sampling diversity in generic scenarios, the theoretical guarantees are limited to simpler settings and one may be able to derive a better algorithm for different distributions. Nevertheless, as `PSR` can be readily applied to any off-the-shelf

diffusion model, we believe it would be a generally useful technique for the community to explore and also hope this work inspires further investigations on how one can controllably vary the sampling distribution of a pre-trained diffusion model.

Chapter 2

Diverse Score Distillation

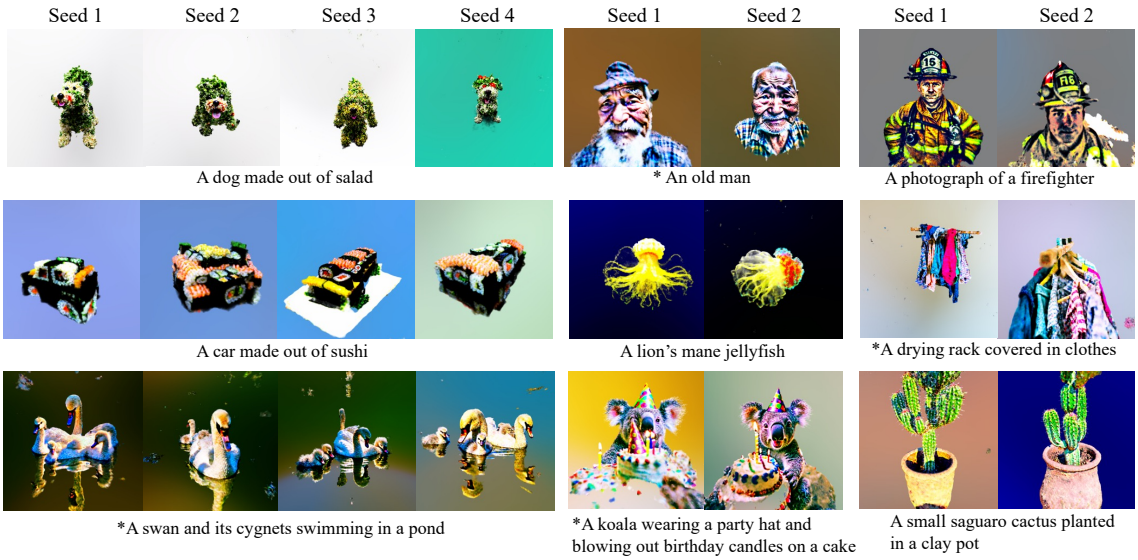

Figure 2.1: **Diverse Score Distillation.** We present a sampling-inspired score distillation formulation that allows obtaining diverse (3D) outputs via different initial optimization seeds. * "A DSLR photo of".

2.1 Introduction

The impressive progress in generative AI has helped democratize the ability to create visual content. In particular, recent image [49] or video [19] generation models allow

end-users to easily create (diverse and photorealistic) visual output from just a text prompt, for example enabling one to synthesize images of an avocado chair or a teddybear skating in Times Square. This success of 2D generative models, however, has not yet been matched by their 3D counterparts, and the goal of inferring diverse and high-fidelity 3D outputs from just a text prompt or an image remains an elusive goal. One key bottleneck towards this is the availability of 3D data. Although we have witnessed promising advances from generative 3D [3, 40] or multi-view [24, 54] approaches that leverage synthetic 3D [5] or real-world multi-view datasets [47], their diversity and scale is still short of the 2D datasets that empower complex and photorealistic generation.

As an alternate paradigm for 3D generation without learning generative 3D (or multi-view) models, several approaches have explored mechanisms to 'distill' pre-trained large-scale 2D generative models for 3D inference. Following DreamFusion [44], which introduced a 'score distillation sampling' (SDS) formulation to approximate log-likelihood gradients from a diffusion model, these approaches cast 3D inference as a (2D diffusion-guided) optimization task and leverage 2D diffusion models to obtain gradients for renderings of 3D representations being optimized. While follow-up methods in this paradigm have since improved various aspects of this pipeline, these methods are all fundamentally *'mode seeking'*, and thus (unlike diffusion-based generation) exhibit limited diversity in their inferred 3D representations (see fig. 2.5).

In this work, we present an alternate formulation for distilling diffusion models that overcomes this limitation, and allows diverse (3D) generation. Our approach is inspired by the ODE perspective on sampling from diffusion models [58] which highlights that a trained model can be viewed as inducing a learned ODE that maps noise samples to data, and different starting points for the ODEs (*i.e.,* different initial noise samples) yield diverse data samples. Building on this insight, we derive a gradient that allows an optimization to 'follow' an ODE, and enables diverse optimization outputs by simply specifying different ODE (noise) initializations. We first validate our approach for 2D image generation via optimization and show that, akin to DDIM sampling, it allows diverse and high-fidelity generation.

However, when naively applying this ODE-based formulation for 3D optimization, we find that it does not result in plausible 3D outputs. We show that this is because while a 2D optimization process can perfectly follow a specified (2D) ODE, the 3D

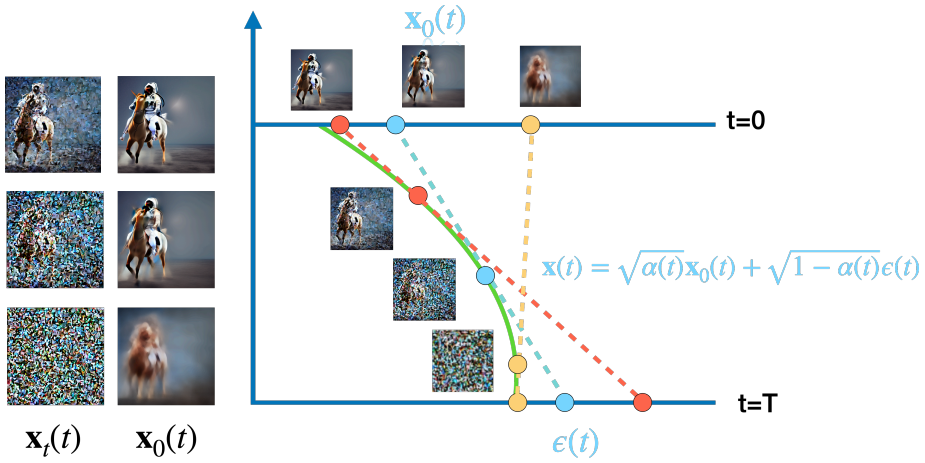

Figure 2.2: **DDIM ODE Trajectory.** When noisy image $\mathbf{x}(t)$ is sampled along a DDIM ODE trajectory, there is an induced process in the one-step prediction space $\mathbf{x}_0(t)$.

optimization process seeks to infer a 3D representation that follows such ODEs across multiple views, and inherently cannot do so perfectly. We then generalize our diffusion distillation formulation to allow for such drift, and show that it allows high-fidelity and diversity for 3D (and 2D) inference. To better highlight the connections (and key differences) of our formulation with prior score distillation objectives, we also present a unifying perspective across existing methods.

We validate our approach for text-based-3D optimization using a pre-trained 2D diffusion model and show that our formulation yields similar (or better) generations in terms of quality compared to the state-of-the-art, while allowing significantly more diverse outputs. In addition, we also showcase our approach's ability to guide single-view 3D inference via distilling novel-view diffusion priors, and in particular, find that it allows multi-modal 3D outputs.

2.2 Related Works

Feed-forward 3D Generation. The success of image generative models [49] has spurred advancements in 3D generation. GAN-based approaches [2, 9] can be trained for 3D generation from images by incorporating rendering inductive bias into the generator. With 3D or multi-view data, diffusion-based 3D or multi-view

generators [3, 24, 40, 54] can be trained to produce high-quality results. However, the scale of 3D datasets [5, 47] remains limited compared to 2D image collections [53], which significantly constrains the performance of the trained models.

Diffusion-guided Optimization via Score Distillation. To circumvent the data limitation, other lines of work seek to optimize a 3D shape by 'distilling' prior from a trained 2D diffusion model. The idea is referred to as Score Distillation Sampling (SDS), which is introduced by [44, 60]. However, these methods suffer from over-smoothing and lack of detail. A large Classfier Free Guidance (CFG) [15] is also needed for effective generation, leading to over-saturation. More importantly, due to the 'mode-seeking' behavior of SDS, the 3D shapes are very similar – an undesirable property for generation tasks.

Several approaches aim to reduce gradient variance and mitigate over-smoothing, including the use of negative prompts and Classifier-Free Guidance [27, 36, 65], prediction differences between adjacent timesteps [34], time annealing [71], and DDIM Inversion [31]. Although these improve fidelity, the lack of diversity in generation persists. To encourage diversity, ProlificDreamer (VSD) [63] proposes optimizing for a 3D distribution given the prompt by using an additional model to capture the distribution of the current rendering. However, this dual training can be unstable and limits the quality.

Perhaps most closely related to ours, some recent approaches have sought inspiration from ODE-based sampling [33, 64] of diffusion models, but these also do not ensure diversity in their formulation.

2.3 Method

Our goal is to develop a diffusion-guided optimization framework that ensures diversity in the optimized outputs *i.e.,* multiple instantiations of the optimization process (using different seeds) should result in different output samples, thus allowing, for example, multiple plausible 3D representations given text conditioning. To develop such a formulation, we take inspiration from the sampling process in denoising diffusion models which inherently yields such diverse samples. We present an initial formulation (section 2.3.2) that seeks to enforce the optimization to track the evolution

of a diffusion sampling process. In section 2.3.3, we then extend this to incorporate scenarios where this might not be exactly feasible (*e.g.,* the renderings of a 3D representation being optimized cannot perfectly match target 2D diffusion processes). Before decribing our formulation, we first review some preliminaries about denoising diffusion, score distillation, and DDIM sampling, all of which play a central role in our approach. After outlining our approach, we place our formulation in context of prior score distillation methods, in particular focusing on the diversity of sampling and relation to ODEs (section 2.3.4).

2.3.1 Background

Denoising Diffusion Models (DMs). To allow modeling a data distribution $p_0(\mathbf{x})$, diffusion models [17, 25, 56] are trained to reverse a forward process that gradually adds noise to data to obtain a prior distribution $p_T = \mathcal{N}(0, I)$. In particular, DMs adopt a forward process defined by coefficients $s(t)$ and $\sigma(t)$, where $p(\mathbf{x}(t)|\mathbf{x}(0)) = \mathcal{N}(s(t)\,\mathbf{x}(0), (s(t)\,\sigma(t))^2\,\mathbf{I})$, and train a noise estimator $\epsilon_\theta^t(\mathbf{x}_t)$ that can allow sampling from the reverse process *i.e.,* mapping noise to data samples. A common (variance preserving) instantiation of diffusion models that our framework adopts is to set $\sigma(t) = \frac{\sqrt{1-\alpha(t)}}{\sqrt{\alpha(t)}}$ and $s(t) = \sqrt{\alpha(t)}$.

DDIM Sampling and ODEs. We build on the DDIM [56] sampling process that can (deterministically) sample from the distribution modeled by a trained diffusion model ϵ_θ via the following iterative procedure:

$$\frac{\mathbf{x}(t-\delta)}{\sqrt{\alpha(t-\delta)}} = \frac{\mathbf{x}(t)}{\sqrt{\alpha(t)}} + [\sigma(t-\delta) - \sigma(t)]\epsilon_\theta^t(\mathbf{x}(t), \mathbf{y}) \tag{2.1}$$

, where $\mathbf{y}$ is a conditioning variable (CLIP text embeddings for the case of text-to-image generation). This update actually corresponds to a discretized approximation for the 'Probability Flow Ordinary Differential Equation' (PF-ODE) [58] for the diffusion model:

$$\frac{d\bar{\mathbf{x}}(t)}{dt} = \epsilon_\theta^t(\sqrt{\alpha(t)}\bar{\mathbf{x}}(t), \mathbf{y})\frac{d\sigma(t)}{dt} \tag{2.2}$$

where $\bar{\mathbf{x}}(t) = \frac{\mathbf{x}(t)}{\sqrt{\alpha(t)}}$. In particular, the distribution of generated samples $\mathbf{x}(0)$ obtained by following paths from random starting points $\mathbf{x}(T) \sim \mathcal{N}(0, I)$ is equivalent to the

distribution captured by (stochastic) generation from the diffusion model.

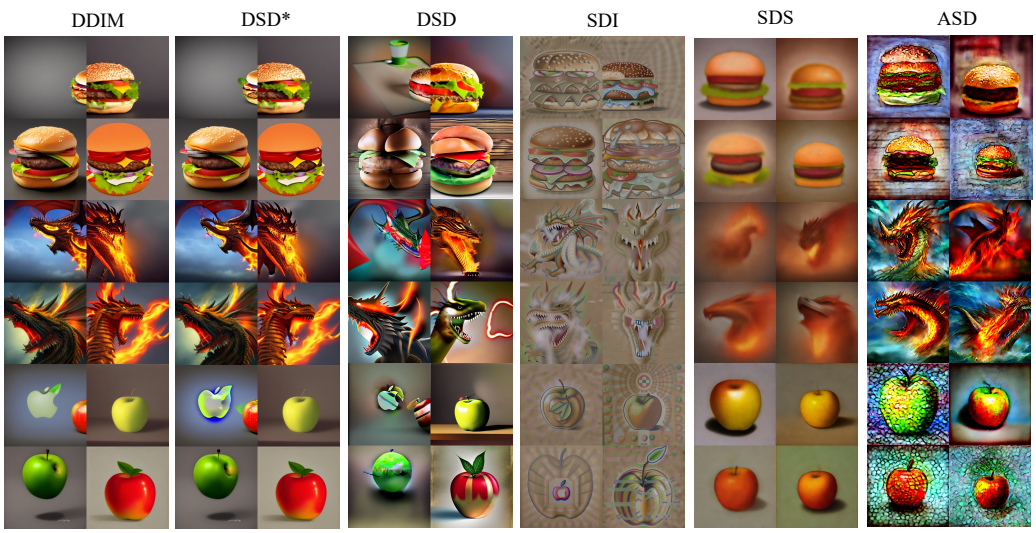

Figure 2.3: 2D Distillation Results using DDIM as reference. The prompts are "A hamburger", "A dragon with flames coming out of its mouth" and "An apple". We assign a fixed initial noise for each grid. When the number of DDIM steps equals the optimization step, our method (DSD*) resembles the DDIM sample. When different, the optimized image will deviate from the original ODE by a small margin. We observe that PSR yields more diverse and plausible generations compared to alternates.

Data and Noise Trajectories Induced by DDIM. We can define a 'one-step data prediction' variable as:

$$\mathbf{x}_0(t) = \bar{\mathbf{x}}(t) - \sigma(t)\epsilon(t) \tag{2.3}$$

, where $\epsilon(t) = \epsilon_\theta^t(\mathbf{x}(t), \mathbf{y})$. Combining eq. (2.2) and eq. (2.3), we can define an induced ODE in $\mathbf{x}_0(t)$ space:

$$\frac{d\mathbf{x}_0(t)}{dt} = -\sigma(t)\frac{d}{dt}\epsilon_\theta^t(\mathbf{x}(t), \mathbf{y}). \tag{2.4}$$

A discrete update for this one-step data prediction trajectory can also be derived as:

$$\mathbf{x}_0(t-\delta) = \mathbf{x}_0(t) - \sigma(t-\delta)[\epsilon_\theta^{t-\delta}(\mathbf{x}(t-\delta), \mathbf{y}) - \epsilon_\theta^t(\mathbf{x}(t), \mathbf{y})] \tag{2.5}$$

As noticed in [25, 33], the updates in $\mathbf{x}_0$ space have smaller variance compared to that of $\mathbf{x}_t$, which is desirable for an optimization process. We visualize the ODE

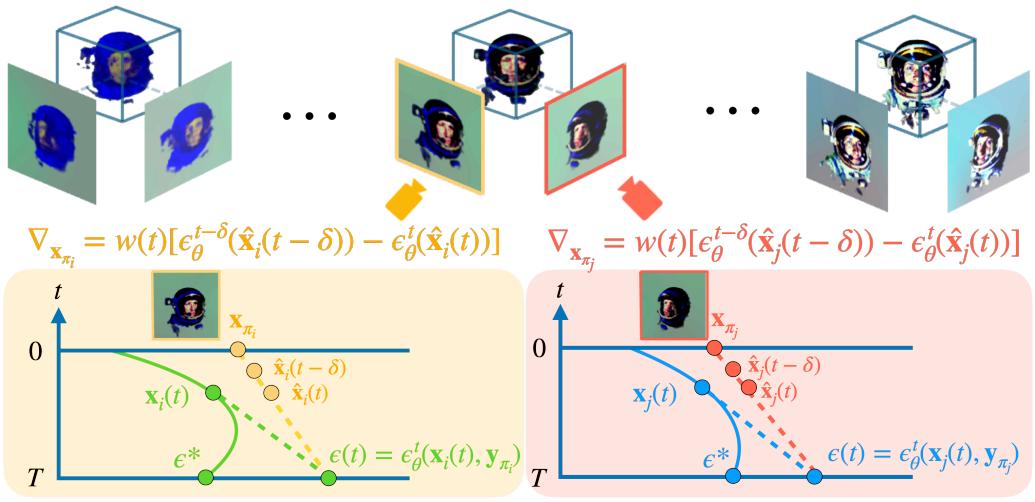

Figure 2.4: **Overview of DSD.** A unique ODE starting point ϵ^* is assigned to each 3D shape throughout the optimization process. Renderings from different views are assumed to be on the view-conditioned ODE, starting from ϵ^*. At each iteration, DSD simulates the corresponding ODE up to time t and obtains noise prediction $\epsilon(t)$ from the ODE. The rendered view is connected to the ODE by an interpolation approximation, which is then used to obtain the gradient.

trajectory for $\mathbf{x}(t)$ and the induced ones for $\epsilon(t)$ and $\mathbf{x}_0(t)$ in fig. 2.2.

Score Distillation. To leverage pre-trained diffusion models for guiding optimization, a score distillation-based approach relies on obtaining gradients for an image. We use $\mathbf{x}_\pi$ to represent the image whose gradient ($\nabla \mathbf{x}_\pi$) will be used to update the optimizable parameter ψ. For our 2D experiments, where we directly optimize an image, $x_\pi = \psi$ and $\partial g / \partial \psi = I$. For 3D generation, ψ denotes the parameter of the 3D shape, and x_π is the rendering of 2D images over camera poses π, i.e. $x_\pi = g(\psi, \pi)$. Here, g is the differential renderer. Using the the chain rule, one can get:

$$\nabla_\psi = \left(w(t) \nabla_{\mathbf{x}_\pi} \frac{\partial g}{\partial \psi} \right) \tag{2.6}$$

, where $w(t)$ is a weighting term to normalize gradients at various noise levels for stable optimization. The central design choice in this process is how one can obtain the gradient $\nabla \mathbf{x}_\pi$ (as well as how to sample the diffusion timestep t for computing the score).

2.3.2 Sampling-based Score Distillation

For simplicity, we first consider a 2D image generation scenario *i.e.,* using score distillation to optimize an image given a text description ($\psi \equiv \mathbf{x}_\pi$). We seek to define a gradient update that allows the optimized variable $\psi = \mathbf{x}_\pi$ to follow a $\mathbf{x}_0$ ODE trajectory (eq. (2.4)). Such an update would ensure generation fidelity, as the final image would be likely under the diffusion model (as it corresponds to on ODE output), while also allowing diversity by simply sampling different seeds $\epsilon^* \sim \mathcal{N}(0, \mathbf{I})$ as starting points for the ODE. We can define such a gradient by adapting the update rule for $\mathbf{x}_0(t)$ (eq. (2.5)):

$$\nabla^{\text{Sampling}}_{\mathbf{x}_\pi} := \sigma(t-\delta)[\epsilon_\theta^{t-\delta}(\mathbf{x}(t-\delta), \mathbf{y}) - \epsilon_\theta^t(\mathbf{x}(t), \mathbf{y})] \tag{2.7}$$

To ensure that $\nabla^{\text{Sampling}}_{\mathbf{x}_\pi}$ allows the update of $\mathbf{x}_\pi$ to follow one underlying ODE (specified by a random seed ϵ^*), both $\mathbf{x}(t-\delta)$ and $\mathbf{x}(t)$ should satisfy the DDIM PF-ODE. As shown in the appendix, we guarantee this by defining:

$$\begin{aligned}
&\mathbf{x}(t) = \texttt{DDIM-Forward}(\epsilon^*, \mathbf{y}, T \to t) \\
&\epsilon(t) = \epsilon_\theta^t(\mathbf{x}(t), \mathbf{y}) \\
&\mathbf{x}_0(t) = (\mathbf{x}(t) - \sqrt{1-\alpha(t)}\epsilon(t))/\sqrt{\alpha(t)} \\
&\mathbf{x}(t-\delta) = \sqrt{\alpha(t-\delta)}\mathbf{x}_0(t) + \sqrt{1-\alpha(t-\delta)}\epsilon(t)
\end{aligned} \tag{2.8}$$

, where DDIM-Forward$(\epsilon^*, \mathbf{y}, T \to t)$ is the numerical solver for eq. (2.2) from time T to t, starting from ϵ^*.

We refer to the formulation in eq. (2.7) and eq. (2.8) as 'Sampling-based Score Distillation' as the resulting optimization simulates the ODE sampling process from $\mathbf{x}(T) = \epsilon^*$. In fact, when the number of optimization steps are chosen to be exactly equal to DDIM sampling steps, outputs from this formulation are equivalent to DDIM samples, and we discuss in the appendix how minor modifications to the step size and δ can yield similar generations if they are not.

2.3.3 Distillation via Interpolation Approximation

While we motivated the formulation in section 2.3.2 via optimization of a 2D representation, we can use the derived score function to also optimize a 3D representation. As shown by Perp-Neg [1], images generated from the same seeds with view-conditioned prompts using the orthogonal update rule can be treated as the set of images from one object. Therefore, we use a *common* seed ϵ^* across all views of the 3D representation (see appendix for empirical justification) with separate view-conditioned prompt $\mathbf{y}_\pi$ [1] to instantiate the ODE that each view $\mathbf{x}_\pi$ should follow. To obtain different 3D shapes ψ_i and ψ_j, we can choose $\epsilon_i^* \neq \epsilon_j^*$.

However, naively applying this formulation does not produce impressive (or even valid) 3D generations. The key reason for this is that the 3D representation ψ cannot accurately follow the computed score for each viewpoint $\mathbf{x}_\pi$, thus leading the (3D) optimization process to deviate from the (2D) ODE paths. This is particularly challenging because the gradient formulation in section 2.3.2 only depends on the specified ODE, and not on the current estimate $\mathbf{x}_\pi$, thus providing no 'correction' mechanism for the 3D representation to generate valid renderings in case of a drift.

To address this issue, we observe that both $\mathbf{x}(t)$ and $\mathbf{x}(t-\delta)$ in eq. (2.8) can be expressed as linear combinations of $\mathbf{x}_0(t)$ and $\epsilon(t)$. Our key insight is that we can instead re-define them to be an interpolation of $\mathbf{x}_\pi$ and $\epsilon(t)$:

$$\begin{aligned}
\mathbf{x}(t) &= \texttt{DDIM-Forward}(\epsilon^*, \mathbf{y}, T \to t) \\
\epsilon(t) &= \epsilon_\theta^t(\mathbf{x}(t), \mathbf{y}) \\
\hat{\mathbf{x}}(t) &= \sqrt{\alpha(t)}\mathbf{x}_\pi + \sqrt{1-\alpha(t)}\epsilon(t) \\
\hat{\mathbf{x}}(t-\delta) &= \sqrt{\alpha(t-\delta)}\mathbf{x}_\pi + \sqrt{1-\alpha(t)}\epsilon(t)
\end{aligned} \tag{2.9}$$

In the ideal case, $\mathbf{x}_\pi$ evolves similarly to $\mathbf{x}_0(t)$, but when it does not, this alteration allows a correction mechanism as the gradients are dependent on the current optimization variable. We thus generalize our previous formulation for scenarios where the optimization may drift from the ODE (shown in fig. 2.4). This leads to our Diverse Score Distillation (DSD) gradient:

$$\nabla_{\mathbf{x}_\pi}^{DSD} = \sigma(t-\delta)[\epsilon_\theta^{t-\delta}(\hat{\mathbf{x}}(t-\delta), \mathbf{y}) - \epsilon_\theta^t(\hat{\mathbf{x}}(t), \mathbf{y})] \tag{2.10}$$

Algorithm 1 Diverse Score Distillation in 3D

Initialization: 3D parameter ψ, trained text-to-image diffusion model ϵ_θ^t, prompt $\mathbf{y}$, set of cameras around 3D shape Π, differentiable renderer g
$\epsilon^* \sim \mathcal{N}(0, I)$
for $i = 1$ to N **do**
    $t \leftarrow T(1 - i/N)$
    $\pi \leftarrow \text{Uniform}(\Pi)$
    $\mathbf{x}_\pi \leftarrow g(\psi, \pi)$
    Compute $\nabla^{\text{DSD}}_{\mathbf{x}_\pi}$ (eq. (2.10))
    $\nabla_\psi = w(t)\nabla^{\text{DSD}}_{\mathbf{x}_\pi} \frac{\partial g}{\partial \psi}$
end for
return ψ

, where we obtain $\hat{\mathbf{x}}(t-\delta)$ and $\hat{\mathbf{x}}(t)$ from eq. (2.9). Using this formulation, we can instantiate our DSD-based 3D optimization procedure in algorithm 1.

2.3.4 Comparing Score Distillation Formulations

Our approach operationalized two key insights: a) if the optimization approach follows the evolution of an ODE, we can expect higher fidelity outputs as these are likely under the diffusion model, and b) the ability to control the generation process (via a random seed) can yield diverse generations. Below, we briefly review alternate score distillation frameworks, in particular highlighting whether they follow an ODE and/or can generate diverse output, and also summarize this in in table 2.1.

Score Distillation Sampling (SDS). SDS defines $\nabla^{\text{SDS}}_{\mathbf{x}_\pi} = \mathbb{E}_\epsilon[\epsilon_\theta^t(\mathbf{x}(t), \mathbf{y}) - \epsilon]$, where $\mathbf{x}(t)$ is defined by adding random noise to $\mathbf{x}_\pi$: $\mathbf{x}(t) = \sqrt{\alpha(t)}\mathbf{x}_\pi + \sqrt{1-\alpha(t)}\epsilon$. While this corresponds to the gradient of image log-likelihood under diffusion, the gradient update does not correspond to an ODE. Moreover, the randomly sampled t and ϵ make this a high-variance estimate.

Asynchronous Score Distillation(ADS) [34] also samples a random ϵ every iteration. Unlike SDS, the gradient is calculated from two adjacent timesteps. The gradient is the same as the DDIM update rule for the sampled ODE at each iteration. However, since the ODE varies with each iteration, this approach also cannot obtain diverse results owing to the high variance gradients.

Score Distillation via Inversion (SDI) [33] (similarly [31]) seeks to follow the

	SDS	ASD	SDI	Consistent 3D	VSD	DSD
Diversity	✗	✗	✗	✓	✓	✓
ODE-following	✗	✗	✓	✗	✗	✓

Table 2.1: **Conceptual Comparisons of Score Distillation Methods.** An optimal distillation method should satisfies both criteria.

ODE update rule by inverting the current image (assumed to be $\mathbf{x}_0(t)$) back to $x(t)$ using DDIM Inversion [56]. The inversion process is a good estimation of the current ODE and a low-variance gradient is applied at every iteration. However, the choice of the ODE cannot be specified and solely depends on the (3D) initialization, leading to lack of diversity in the results.

Consistent3D [64] attempts to follow the consistency ODE [59] by setting a constant ϵ^*. However, this does not define a valid ODE in the space of denoising diffusion models (including StableDiffusion), necessitating a large classifier-free guidance (CFG) [15] scale (50-100) for the method to be effective, which in turn reduces diversity. In contrast, our approach operates with a CFG of 7.5, aligning with the standard 2D sampling process.

2.4 Experiments

We seek to empirically validate the ability of `PSR` to guide diffusion-based optimization and generate diverse and high-fidelity outputs. We first analyze our formulation in the simple but informative task of image optimization and then examine text-to-3D generation. We further ablate some key decision decisions and also highlight the applicability for distillation-based single-view reconstruction.

Baselines. We select SDS [44], HiFA [71] and ASD [34] to represent the methods that sample random noise, and select Consistent 3D [64] and SDI [33] as the approaches that perform score distillation from an ODE perspective. We also include VSD [63] as it minimizes the KL divergence between the rendering and image distribution, thus promising diverse generation. For fair comparison across methods, we adopt the codebase from SDI (which builds on a commonly adopted implementation [11]) and use a shared one-stage optimization scheme. We only vary

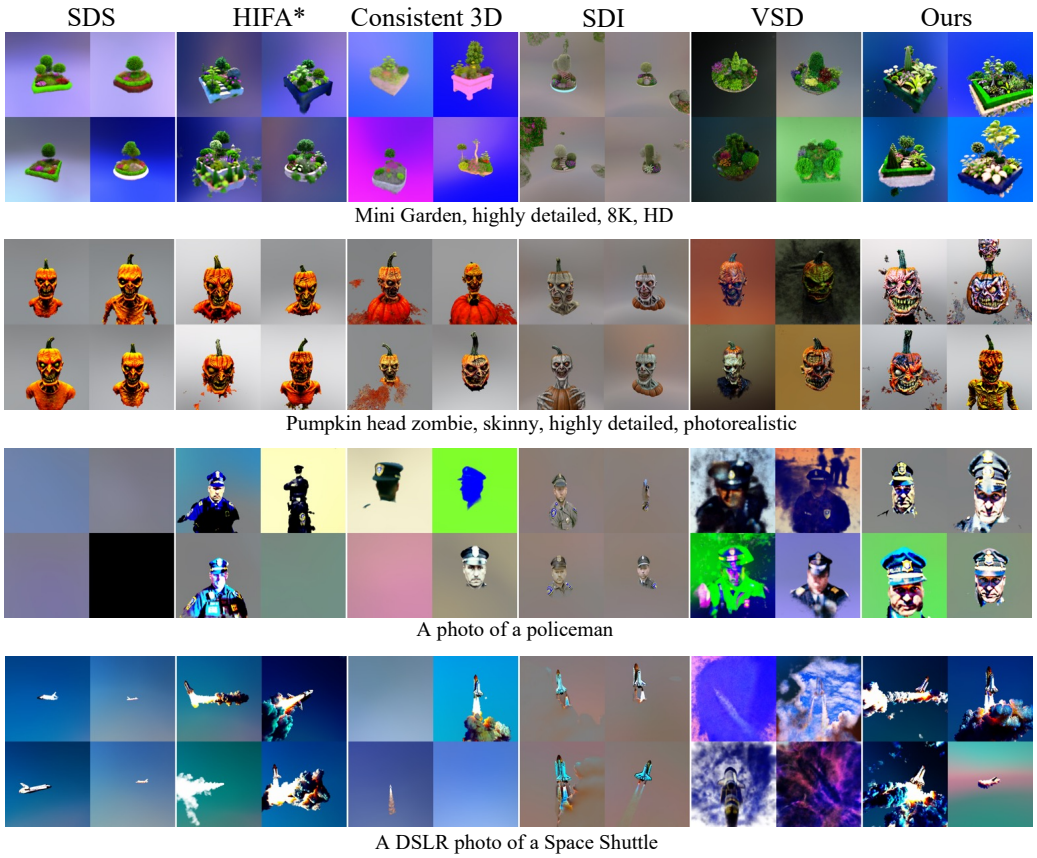

Figure 2.5: **Generation Comparison.** We visualize 4 text-to-3D generations from various score distillation methods. We find that DSD is capable of generating high-quality 3D shapes while being more diverse compared to prior methods. Please see supplementary for videos.

the score function across methods (except for VSD, where we use its released code for multi-particle optimization).

2.4.1 Image Generation via Score Distillation

In fig. 2.3, we visualize 4 samples generated by each method across different prompts. When our optimization steps are aligned with the DDIM sampling steps (`PSR`*), the results are equivalent to DDIM sampling. However, in the general case (`PSR`), where the optimization steps differ from the DDIM steps (see Appendix), slight variations do arise. Despite this, our method consistently produces generations that are more

diverse and of higher fidelity compared to baseline score distillation methods. We use the official code of SDI [33] for its 2D results.

2.4.2 Text to 3D Generation

We use various score distillation frameworks to obtain 4 different 3D generations from each text prompt and visualize these in fig. 2.5.

Consistent with our discussion in section 2.3.4, SDS [44] and ASD [34] suffer from mode-seeking behavior and produce less sharp results. Although VSD can theoretically generate diverse results, simultaneous training of a LoRA [21] network that models the rendering distribution in VSD limits its quality and is potentially unstable. Consistent 3D is more diverse than the aforementioned methods thanks to its ODE formulation. However, the large CFG is harmful to the result. While SDI [33] is capable of generating high-quality images, its diversity is limited due to the inversion-based ODE seeking.

We also quantitatively evaluate the quality and diversity of the distilled 3D shapes. For quality, we use FID [14] and CLIP-SIM [45]. For each prompt, FID is calculated across 100 generated samples from Stable Diffusion and the 100 rendered images from 10 3D shapes. For diversity, we propose to use LPIPS [69] to measure the difference between different 3D shapes generated using the same prompt. We use generated 3D shapes per-prompt and measure the pair-wise LPIPS for the same camera. table 2.2 demonstrates the calculated metrics, measured from 15 prompts. A smaller FID indicates that the generated 3D shapes are more similar to the 2D images produced by the DM, while a higher CLIP-SIM reflects better text-image alignment of the 3D shapes. Additionally, when the generated 3D shapes appear more distinct, the LPIPS score should be larger. As shown in table 2.2, and consistent with the observations in fig. 2.5, our method generates 3D shapes with comparable or superior quality while achieving greater diversity compared to baseline methods.

Ablation. We also investigate alternative ways of getting the $\mathbf{x}(t)$ and $\mathbf{x}(t-\delta)$ from eq. (2.9): random noise (similar to ASD [34]), linear approximation (similar to Consistent3D [64], and our sampling with interpolation approximation (section 2.3.3). fig. 2.6 and table 2.3 show the qualitative and quantitative results, respectively. It

	FID ↓	CLIP-SIM ↑	LPIPS ↑
SDS	270.80	27.69	0.2695
HiFA	258.99	28.29	0.3202
SDI	**247.98**	28.68	0.2407
Consistent 3D	270.58	28.16	0.2936
VSD	269.22	27.83	0.3336
Ours	251.40	**28.97**	**0.4013**

Table 2.2: **Quantitative Comparison.** We evaluate the fidelity (FID, CLIP-SIM) and diversity (LPIPS) of generations from different methods.

	FID ↓	CLIP-SIM ↑	LPIPS ↑
Random (ASD)	265.80	28.24	0.3333
Linear	274.46	28.17	0.3579
Ours	251.40	**28.97**	**0.4013**

Table 2.3: **Ablation.** We ablate three ways of getting $\mathbf{x}(t)$ and $\mathbf{x}(t-\delta)$: random noise at each iteration (ASD [34]), linear approximation and ODE sampling approximation. Simulating the given ODE improves the overall quality and diversity.

can be observed that simulating a predefined ODE trajectory improves the generation diversity, and that our formulation leads to better results compared to a linear interpolation.

2.4.3 Applying DSD to Other Tasks

Improving Single-view-to-3D Distillation. In addition to text-to-3D generation, score distillation has also been widely used for single-view-to-3D reconstruction [32, 70] by leveraging DMs that are trained with image and camera conditions. Most methods still utilize the SDS [44] gradient, which is prone to mode-seeking and often results in overly smooth outputs. Given an input image, the geometry of the underlying 3D shape is highly undetermined. As shown in fig. 2.7, with the same DM, our method can produce diverse 3D shapes with high-frequency details.

Diverse Generation with MVDream. As all other methods [33, 34, 44, 63, 64, 65, 71] that use stable diffusion [49] as the score model, our approach also encounters the Janus effect. One way to mitigate this issue is by employing multiview diffusion

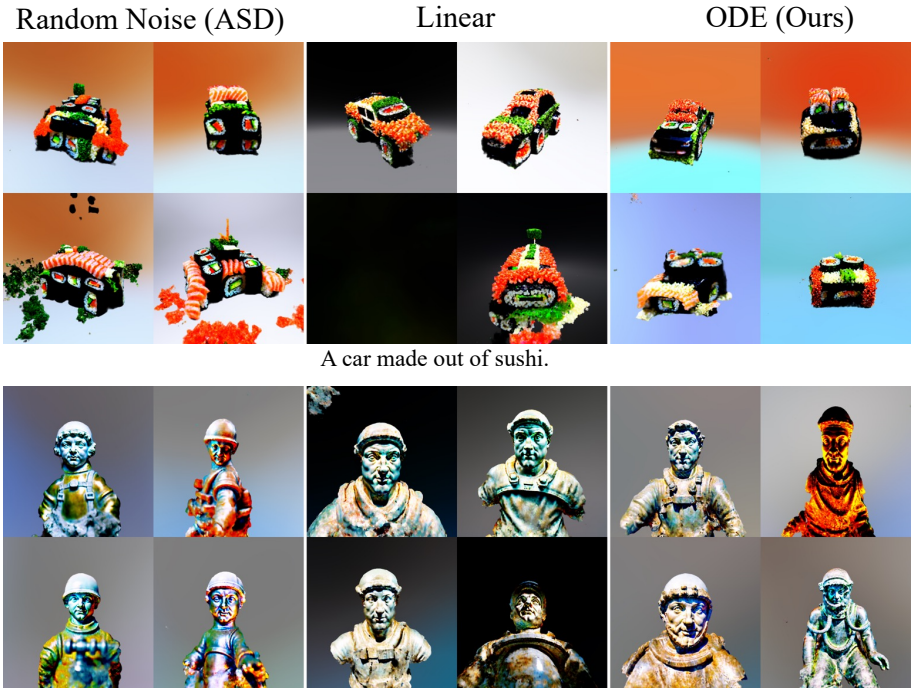

Figure 2.6: **Qualitative Ablation Results.** ODE sampling approximation improves diversity and is also more robust compared with Random noise.

models trained on multiview data with camera conditions. For instance, MVDream [54] is such a model, augmented with additional text conditioning. However, similar to DreamFusion [44], the original MVDream suffers from limited diversity due to its reliance on SDS. By integrating our method with MVDream, we achieve diverse 3D generation without the Janus effect. Examples of generated samples are shown in fig. 2.8.

2.5 Discussion

We presented DSD, a formulation for score distillation that akin to sampling from a diffusion model, can yield diverse samples via optimization, while matching/improving the fidelity compared to existing formulations. However, there are still several challenges and open questions. First, while sampling from diffusion models

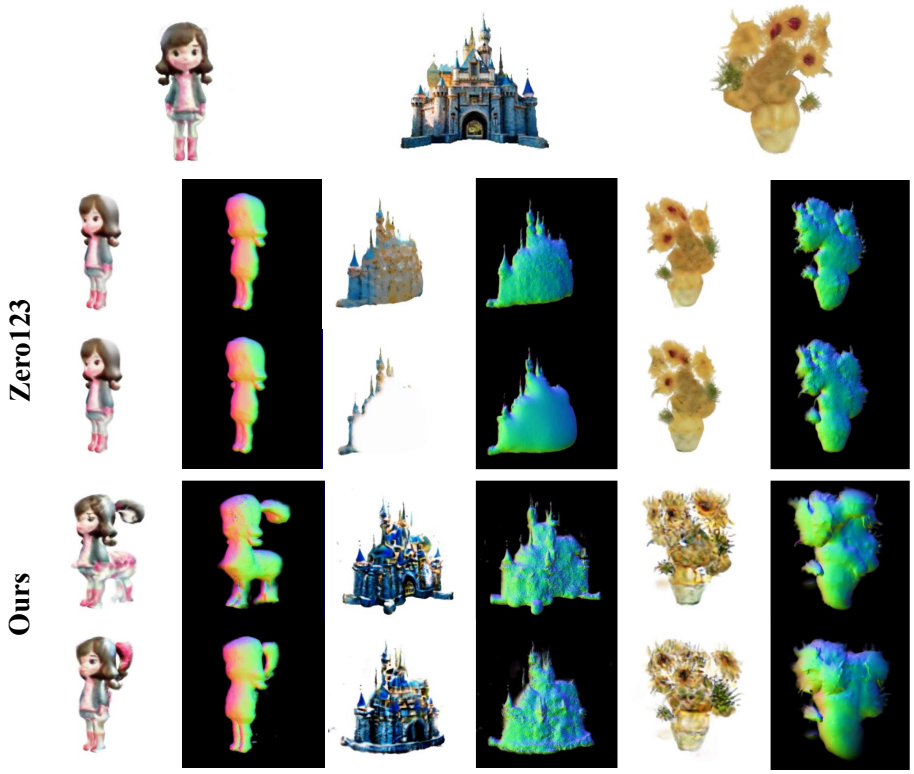

Figure 2.7: **Diverse Single-view-to-3D Distillation.** With an image and camera-conditioned DM, DSD reconstructs high-frequency details of the object with diverse interpretations of the underlying geometry. The default distillation adopted by Zero-1-to-3 [32] does not yield multi-modal generations and has limited details.

is (relatively) efficient, score distillation-based optimization is not and it would be interesting to explore formulations that can match the inference speed of denoising diffusion. Moreover, the photo-realism of optimized 3D representations still falls short of generations from the guiding 2D diffusion models, and it remains a challenge to bridge this gap. Finally, we primarily investigated the applications of DSD for text-to-3D and single-view reconstruction, but believe this can be more broadly applicable, for example in 3D editing [12, 62] or relighting [23].

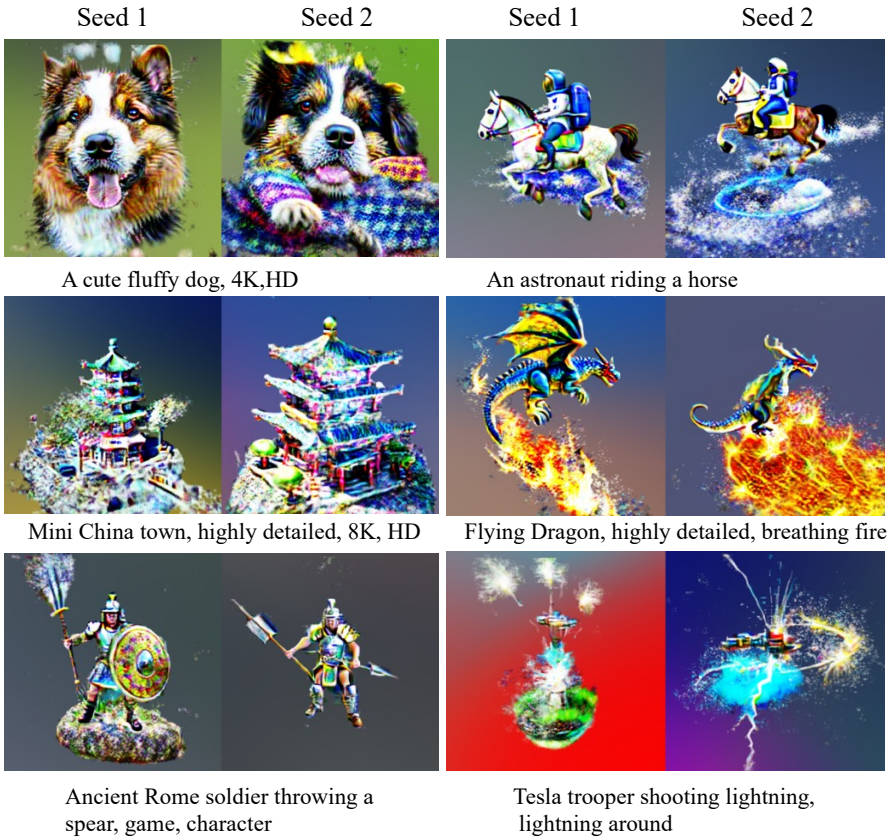

Figure 2.8: **Diverse Generation with MVDream.** By using a camera-aware diffusion model, such as MVDream, our method can generate diverse 3D shapes without Janus effects.

Appendix A

Parametric Score Rescaling for Steering Sampling Diversity Supplementary

A.1 Proofs and Discussions

Relating Score Function to Noise Prediction. In a diffusion model, we define a forward process s.t. $\mathbf{x}_t = \alpha_t \mathbf{x}_0 + \sigma_t \epsilon$, and trained the model ϵ_θ to predict the noise given $\mathbf{x}_t$. Defining $\bar{\mathbf{x}}_t \equiv \frac{\mathbf{x}_t}{\alpha_t}$, we get:

$$\bar{\mathbf{x}}_t = \mathbf{x}_0 + \frac{\sigma_t}{\alpha_t}\epsilon$$

Following Tweedie's formula [8], we can show that the noise prediction is an estimate of the score for the variable $\bar{\mathbf{x}}_t$:

$$\nabla_{\bar{\mathbf{x}}_t} \log\ p(\bar{\mathbf{x}}_t) = -\frac{\epsilon_\theta(\mathbf{x}_t, t)}{\frac{\sigma_t}{\alpha_t}}$$

We can also show that:

$$\nabla_{\bar{\mathbf{x}}_t} \log\ p(\bar{\mathbf{x}}_t) = \nabla_{\bar{\mathbf{x}}_t} \log\ p(\mathbf{x}_t) = \alpha_t \nabla_{\mathbf{x}_t} \log\ p(\mathbf{x}_t)$$

Combining the above, we obtain:

$$/labelscore \nabla_{\mathbf{x}_t} \log\ p(\mathbf{x}_t) = -\frac{\epsilon_\theta(\mathbf{x}_t, t)}{\sigma_t} \tag{A.1}$$

Application for $\mathbf{x}_0$ and v prediction diffusion models. Our method can be applied to all types of diffusion models, although we assumed a noise prediction network in the main paper. For several parameterizations for training a score-based model, e.g. denoising prediction ($\mathbf{x}_{0,\theta}$), noise prediction (ϵ_θ), and v-prediction (v_θ) [50], one can easily show their equivalence given $\mathbf{x}_t$:

$$\epsilon_t = \frac{\mathbf{x}_t - \alpha_t \mathbf{x}_{0,\theta}(\mathbf{x}_t, t)}{\sigma_t}$$
$$\epsilon_t = \alpha_t v_\theta(\mathbf{x}_t, t) + \sigma_t \mathbf{x}_t$$

. After obtaining $\bar{\epsilon_t} = r_t \epsilon_t$, one can have:

$$\bar{\mathbf{x}}_0 = (\mathbf{x}_t - \sigma_t \bar{\epsilon_t})/\alpha_t$$
$$\bar{v}_t = (\bar{\epsilon}_t - \sigma_t \mathbf{x}_t)/\alpha_t$$

Mixture of Isotropic Gaussians. The probability density function(PDF) of a mixture of isotropic gaussians is given by:

$$p(\mathbf{x}) \sim \sum_{i=1}^{K} \pi_i \mathcal{N}(\mathbf{x}; \mu_i, \sigma_i^2 I)$$

where I is the identity matrix, and $\mathcal{N}(\mathbf{x}; \mu_i, \sigma_i^2 I)$ is the PDF of a multivariate Gaussian distribution. From above, the score function can be derived as follows:

$$\nabla_{\mathbf{x}} \log p(\mathbf{x}) = \nabla_{\mathbf{x}} \log \left[\sum_{i=1}^{K} \pi_i \mathcal{N}(\mathbf{x}; \mu_i, \sigma_i^2 I) \right]$$

$$= \frac{\sum_{i=1}^{K} \pi_i \nabla_{\mathbf{x}} \mathcal{N}(\mathbf{x}; \mu_i, \sigma_i^2 I)}{\sum_{i=1}^{K} \pi_i \mathcal{N}(\mathbf{x}; \mu_i, \sigma_i^2 I)}$$

$$= \frac{\sum_{i=1}^{K} \pi_i \left(-\frac{1}{\sigma_i^2}(\mathbf{x}-\mu_i)\right) \mathcal{N}(\mathbf{x};\mu_i,\sigma_i^2 I)}{\sum_{i=1}^{K} \pi_i \mathcal{N}(\mathbf{x};\mu_i,\sigma_i^2 I)}$$

When each Gaussian is well separated, at each point $\mathbf{x}$, one gaussian dominates the density. Suppose the dominating gaussian to be $\mathcal{N}(\mathbf{x};\mu_j,\sigma^2 I)$. Then we can approximate the score function as below.

$$\mathcal{N}(\mathbf{x};\mu_j,\sigma_j^2 I) >> \mathcal{N}(\mathbf{x};\mu_i,\sigma_i^2 I) \ for\, j \neq i.$$

$$\nabla_{\mathbf{x}} \log p(\mathbf{x}) = \frac{\sum_{i=1}^{K} \pi_i \left(-\frac{1}{\sigma_i^2}(\mathbf{x}-\mu_i)\right) \frac{\mathcal{N}(\mathbf{x};\mu_i,\sigma_i^2 I)}{\mathcal{N}(\mathbf{x};\mu_j,\sigma_j^2 I)}}{\sum_{i=1}^{K} \pi_i \frac{\mathcal{N}(\mathbf{x};\mu_i,\sigma_i^2 I)}{\mathcal{N}(\mathbf{x};\mu_j,\sigma_j^2 I)}}$$

$$\simeq -\frac{1}{\sigma_i^2}(\mathbf{x}-\mu_i)$$

, which is a score function of $\mathcal{N}(\mathbf{x};\mu_j,\sigma_j^2 I)$.

In a diffusion model where $p(\mathbf{x})$ follows mixture of isotropic gaussians as $p(\mathbf{x}_0) \sim \sum_{i=1}^{K} \pi_i \mathcal{N}(\mathbf{x};\mu_i,\sigma_i^2 I)$, the noisy distribution $p(\mathbf{x}_t)$ also follows a mixture of isotropic gaussians, as below.

$$p(\mathbf{x}_t) \sim \sum_{i=1}^{K} \pi_i \mathcal{N}(\mathbf{x};\alpha_t\mu_i,(\alpha_t^2\sigma_i^2+\sigma_t^2)I)$$

, and $\bar{p}_k(\mathbf{x}_t)$ can be expressed as below.

$$\bar{p}_k(\mathbf{x}_t) \sim \sum_{i=1}^{K} \pi_i \mathcal{N}(\mathbf{x};\alpha_t\mu_i,\frac{(\alpha_t^2\sigma_i^2+\sigma_t^2)I}{k})$$

Assuming $\mathcal{N}(\mathbf{x};\mu_j,\sigma^2 I)$ is the dominating gaussian at x_t, we obtain:

$$\nabla_{\mathbf{x}_t} \log\ p(\mathbf{x}_t)) \simeq -\frac{\mathbf{x}_t - \alpha_t\mu_j}{\alpha_t^2\sigma_j^2+\sigma_t^2}$$

$$\nabla_{\mathbf{x}_t} \log\ \bar{p}_k(\mathbf{x}_t)) \simeq -\frac{\mathbf{x}_t - \alpha_t\mu_j}{\frac{\alpha_t^2}{k}\sigma_j^2+\sigma_t^2}$$

$$\nabla_{\mathbf{x}_t} \log\ \bar{p}_k(\mathbf{x}_t) = \frac{s_t\sigma_j^2 + 1}{s_t\frac{\sigma_j^2}{k} + 1}\ \nabla_{\mathbf{x}_t} \log\ p(\mathbf{x}_t) \tag{A.2}$$

As a result, if a mixture of isotropic gaussians is well-separated, we can derive a score rescaling actor analogous to r_t. If all gaussians have a same variance σ^2, we get same formulation to PSR.

A.2 More Results and Analysis

2D Toy Data. We experiment on more diverse 2D gaussian mixtures, which manifests the benefit of PSR over temperature scaling on autoregressive model and naive noise scaling. Results are in Fig.A.3 and Fig.A.4.

Pose Estimation. We show more pose prediction results in fig. A.5. PSR predicts tighter samples around the ground truth mode, which can be observed by the low spread of sampled poses compared to score sampling. We also include more depth samples and comparisons fig. A.1, and see that increasing k, across different σ, yields consistent improvement in pose accuracy.

Depth Estimation. We also show the effect of σ and k on the AbsRel metric in fig. A.2. Compared with the DDIM sample ($k = 1$), PSR demonstrates consistent performance gain in various (k, σ) configurations. We also include more depth samples and comparisons fig. A.6. A consistent improvement of PSR result can be observed, compared to the DDIM samples.

Image Generation. We show additional qualitative results for image generation in fig. A.7. We show PSR with varying k and highlight the control over the expression of high frequency details. Additionally, we show examples with DDIM and varying CFG. The significant change in image composition across CFG highlights that CFG changes the global sampling distribution as it injects negative or null prompts, leading to large changes in image semantics, which is orthogonal to PSR which steers the sharpness and flatness of the sampling distribution. Additionally, we show the effect of (k, σ) in Fig. A.12 on FID and CLIP scores. PSR achieves better scores over various configurations of parameters. Interestingly, while DDPM performs worse than DDIM without any scaling, DDPM with PSR achieves better scores than DDIM with PSR.

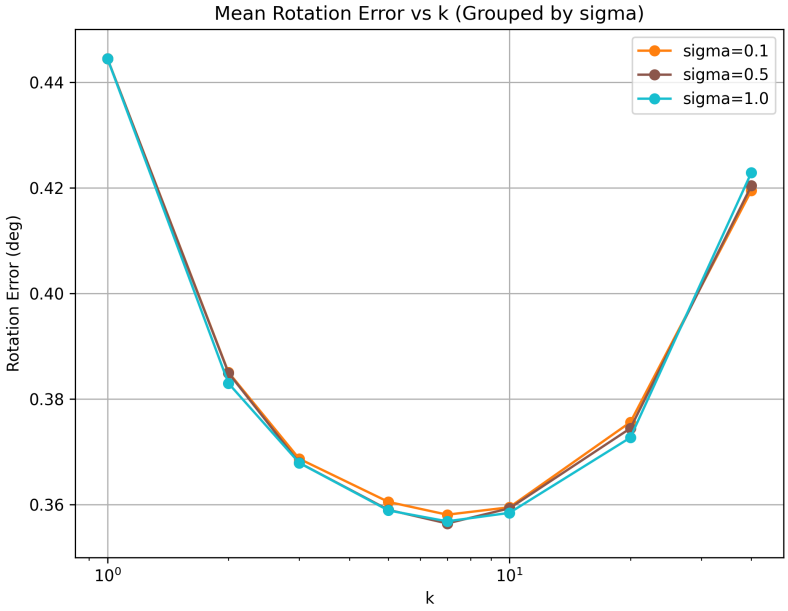

Figure A.1: **Effects of (k, σ) on pose estimation.** PSR with various (k, σ) configurations effectively outperforms the baseline sampling method $k = 1$. While PSR is not sensitive to σ in pose estimation, PSR reaches optimal performance with $k \approx 7$.

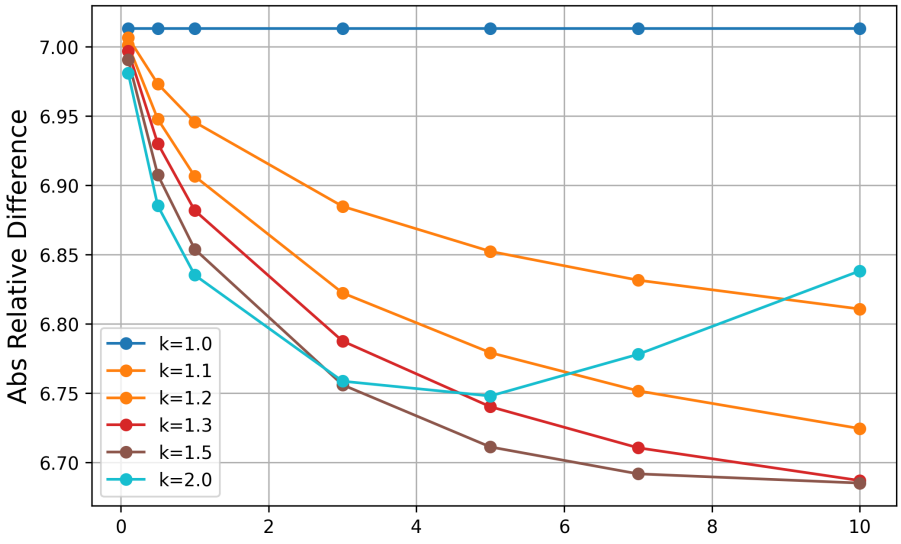

Figure A.2: **Effects of (k, σ) on depth estimation.** Comparing with DDIM sample $(k = 1)$, PSR demonstrates consistent performance gains in various (k, σ) configurations.

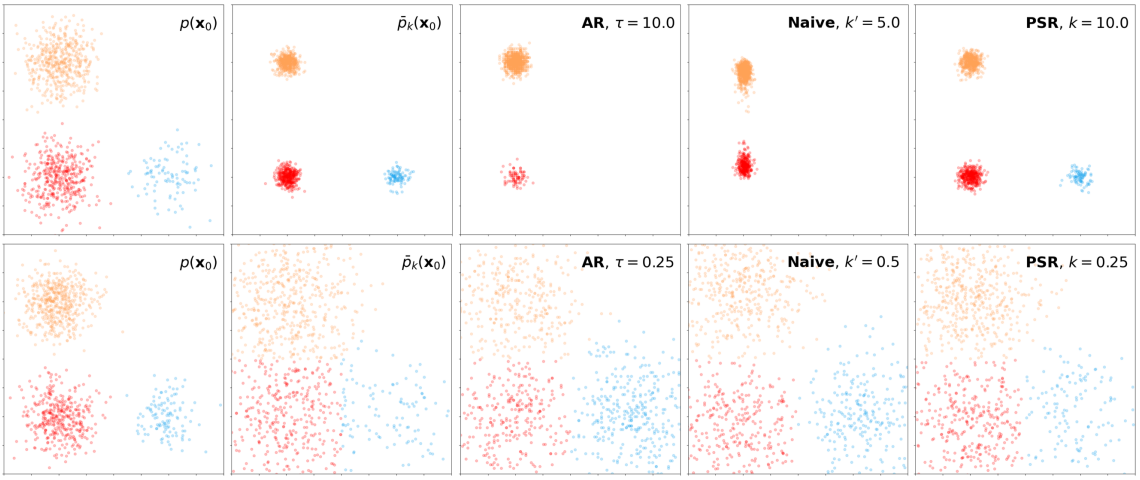

Figure A.3: **Comparison on nonuniform mixture of 2D isotropic gaussians. AR** - temperature scaling, **Naive** - naive noise scaling(NSS), **PSR** - ours. The mixture weight of $p(\mathbf{x}_0)$ is 0.4(red), 0.5(orange), and 0.1(blue). When $k = 10.0$ (top), AR and NNS tend to converge to a subset of modes (orange, red). When $k = 0.25$(bottom), the trend becomes the opposite. AR and NNS biases the samples more on the mode it lost at $k = 10.0$(blue), while the desired weight for blue is 0.1. **PSR** preserves weights at both $k = 10.0, 0.25$.

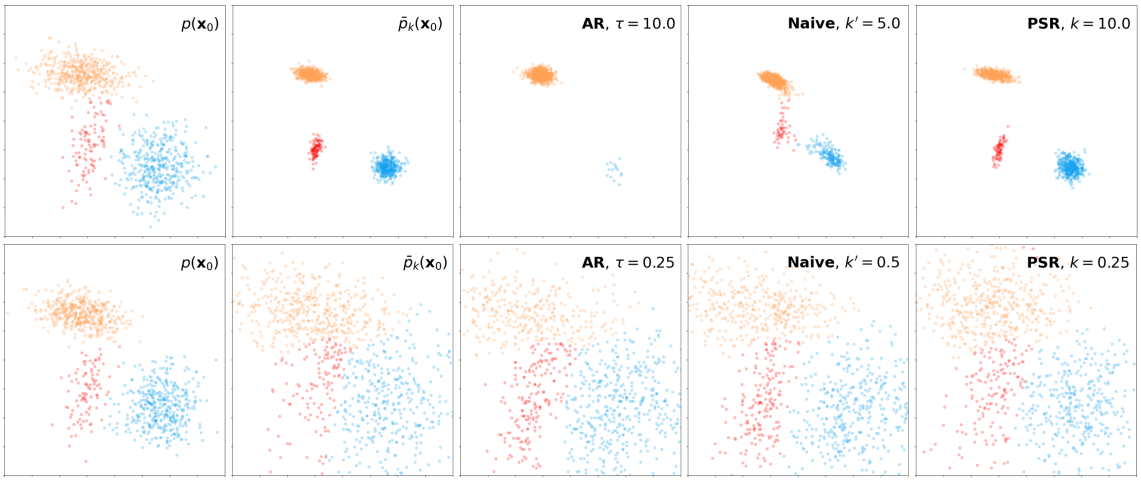

Figure A.4: **Comparison on nonuniform mixture of weakly-separated 2D anisotropic gaussians.** Notation is analogous to the above. The mixture weight of $p(\mathbf{x}_0)$ is 0.1(red), 0.5(orange), and 0.4(blue). Note that the Gaussians are not separated well in this case, making the setup challenging. When $k = 10.0$ (top), AR and NNS tend to converge to a subset of modes. Additionally, NNS tends to generate samples that are close to the mean of each mode, which is undesirable. When $k = 0.25$(bottom), AR biases the samples more on the red mode, while the desired weight for red is 0.1. **PSR** preserves weights and generated samples similar to $\bar{p}_k(\mathbf{x})$ at both k. We use $\sigma = \sqrt{0.025}$ for both k in `PSR`.

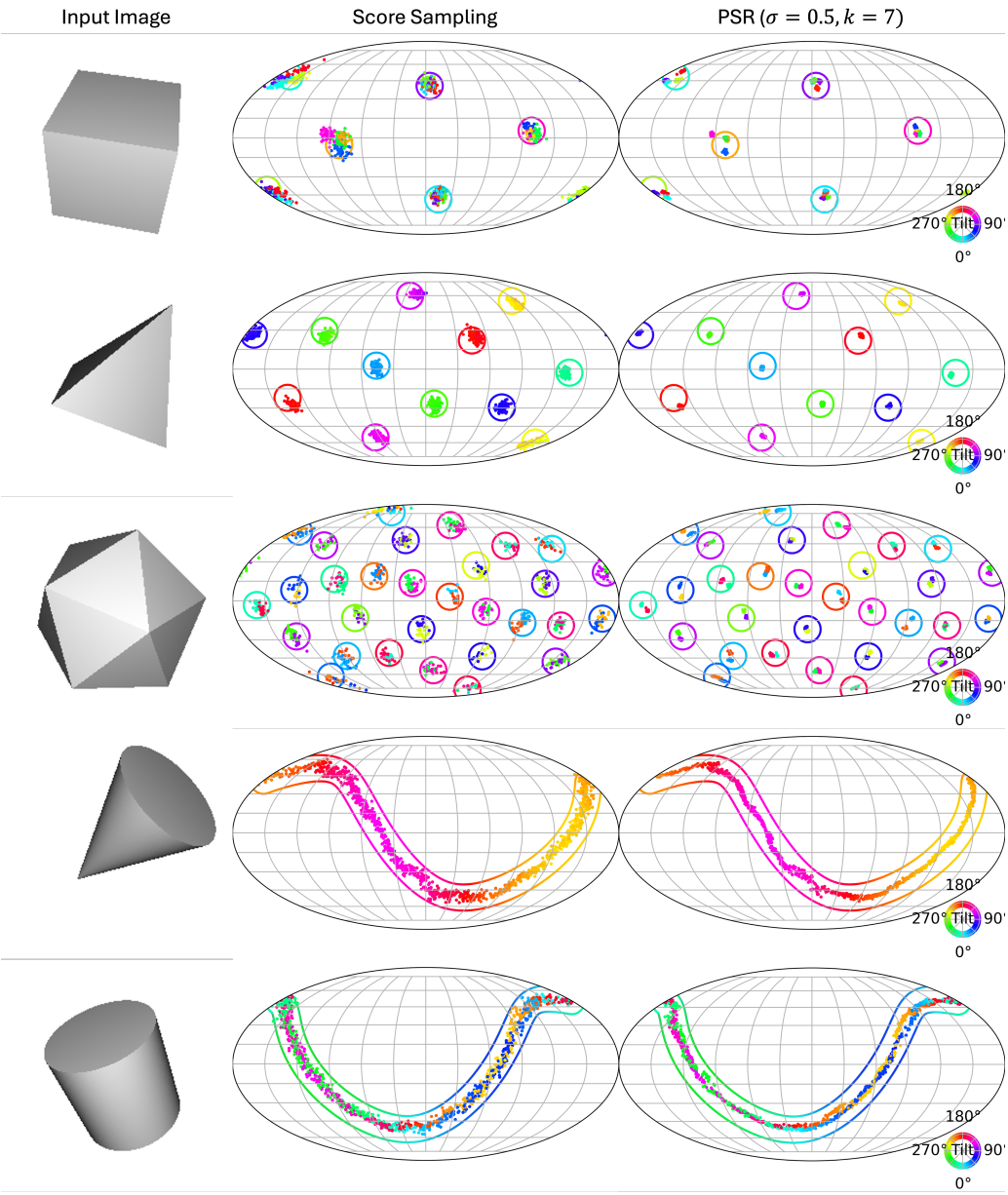

Figure A.5: **More predicted poses on SYMSOL**. We show all 5 classes of shapes in SYMSOL. We use $\sigma = 1, k = 7$ for these visualizations. PSR consistently reduces prediction error across all classes compared to score sampling. We modify the location of samples to exaggerate error by a factor of 15 to show visual difference given plotting constraints.

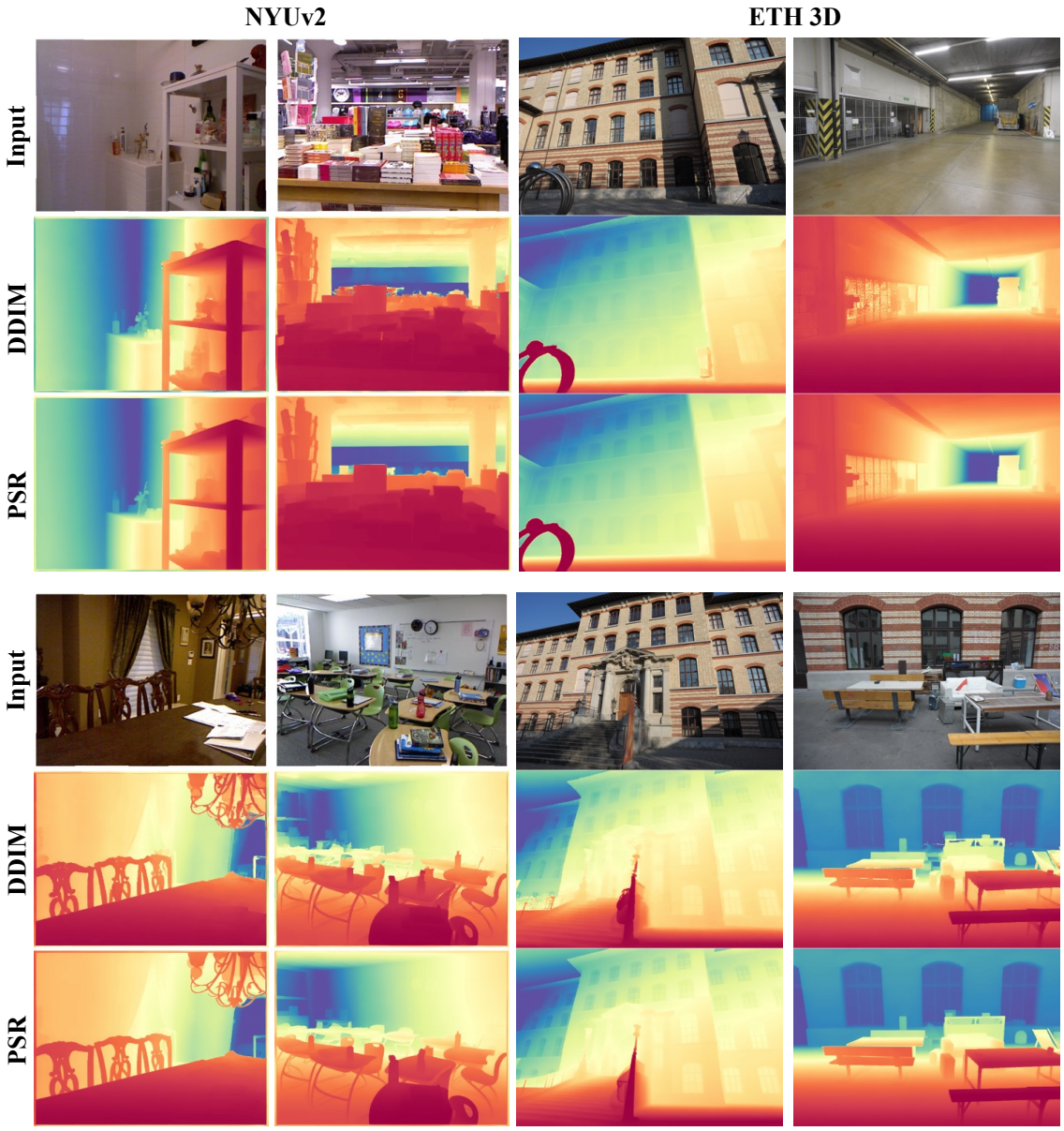

Figure A.6: **More Depth Prediction Comparison.** We include more samples from NYUv2 and ETH3D. PSR demonstrates consistent improvement compared to the DDIM samples.

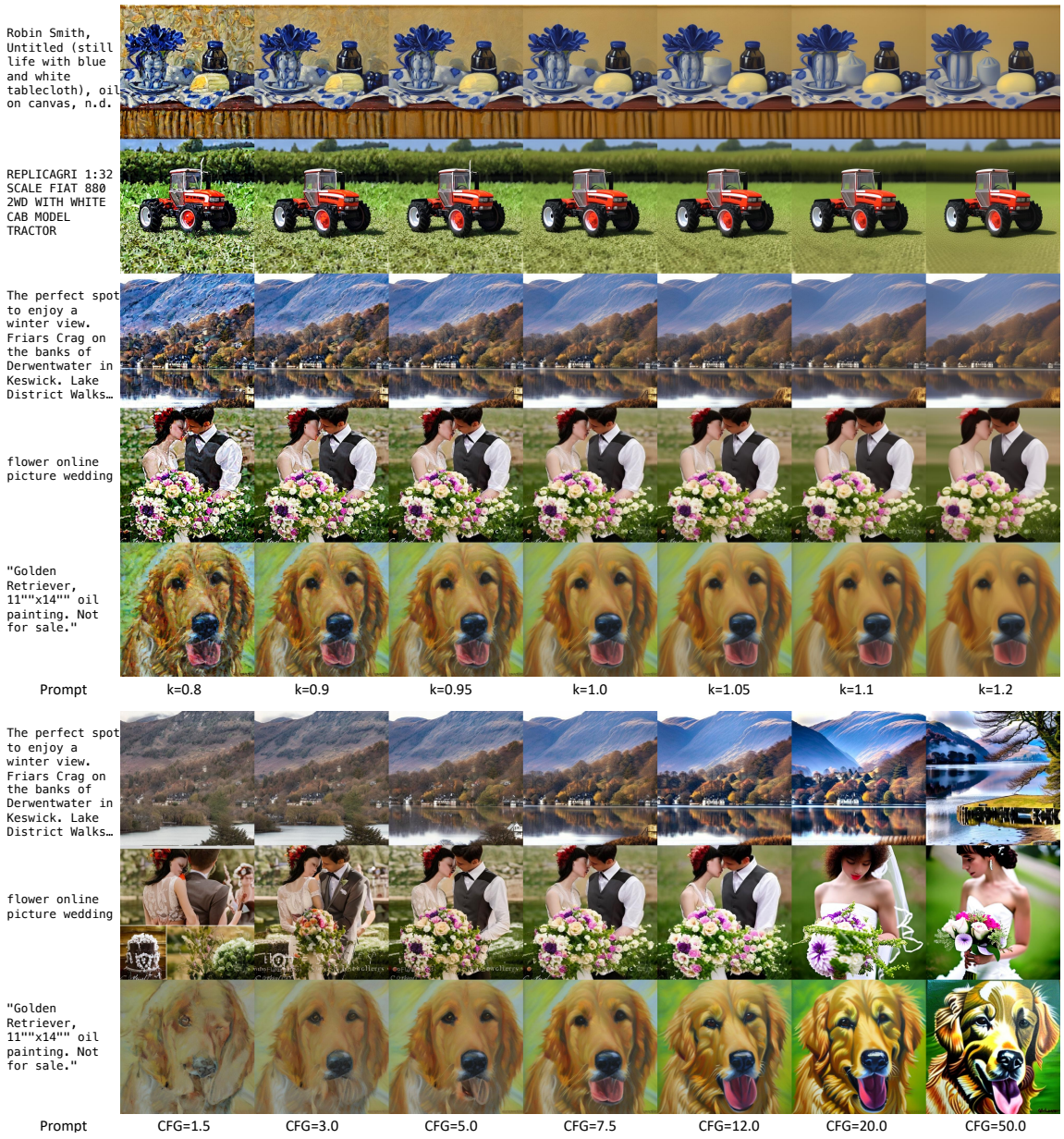

Figure A.7: **More Image Generation Results.** We further show the effect of PSR in steering the sampling distribution of Stable Diffusion to be sharper or flatter. A larger k forces a sharper samplings distribution and creates a smoother image while a smaller k allows for more high frequency details. All PSR (k, 1.0) comparisons (TOP) use $\sigma = 1.0$ and CFG=7.5. When $k = 1.0$, the results are the same as DDIM sampling at CFG=7.5. We also show results for DDIM with varying CFG on (BOTTOM) and observe the CFG making trade-offs between text-alignment and image fidelity by changing the underlying distribution.

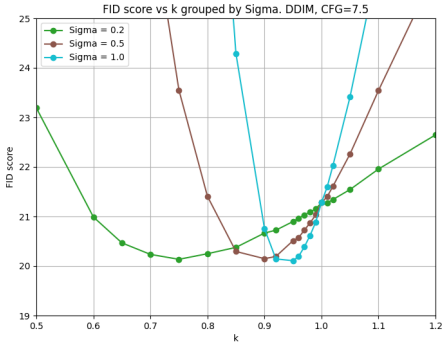

Figure A.8: FID vs. k for DDIM

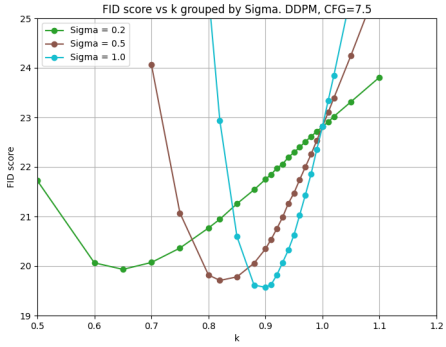

Figure A.9: FID vs. k for DDPM

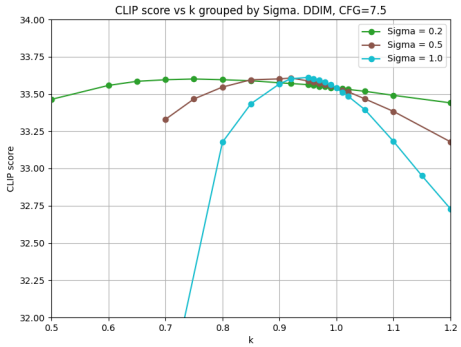

Figure A.10: CLIP vs. k for DDIM

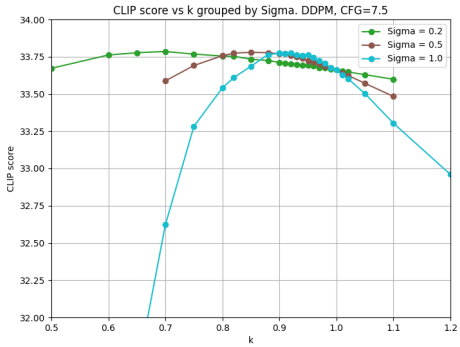

Figure A.11: CLIP vs. k for DDPM

Figure A.12: **Effects of (k, σ) on image generation.** The left column compares FID scores, while the right column compares CLIP scores. The top row shows results with DDIM +PSR, while the bottom row shows results with DDPM+PSR. Our method consistently achieves better FID and CLIP scores over various (k, σ) settings.

Appendix B

Diverse Score Distillation Supplementary

B.1 Additional Discussion

Equivalence of DDIM Sample and Sampling-based Score Distillation. Here we show the Sampling-based gradient in eq. (2.8) is equivalent to the induced DDIM Sampling process in the single-step prediction $\mathbf{x}_0(t)$ space eq. (2.5) if the number of optimization steps is equal to DDIM sampling steps.

We prove this by induction. Assume the equivalence holds true for $t+\delta$, i.e.:

$$\begin{aligned} x(t+\delta) &= \texttt{DDIM-Forward}(\epsilon*, T \to t+\delta) \\ \epsilon(t+\delta) &= \epsilon_\theta^{t+\delta}(x(t+\delta)) \\ x_0(t+\delta) &= \frac{x(t+\delta) - \sqrt{1-\alpha(t+\delta)}\epsilon(t+\delta)}{\sqrt{\alpha(t+\delta)}} \end{aligned} \tag{B.1}$$

By following update rule for eq. (2.8) with the assumption from eq. (B.1), we get:

$$x(t) = \sqrt{\alpha(t)}x_0(t+\delta) + \sqrt{1-\alpha(t)}\epsilon(t+\delta) \tag{B.2}$$

$$x_0(t) = x_0(t+\delta) - \alpha(t)[\epsilon_\theta^t(x(t)) - \epsilon_\theta^{t+\delta}(x(t+\delta))] \tag{B.3}$$

From the definition of a single DDIM step, we get

$$x^{\texttt{DDIM}}(t) = \sqrt{\alpha(t)}x_0(t+\delta) + \sqrt{1-\alpha(t)}\epsilon(t+\delta).$$

Therefore, we have

$$x(t) = x^{\texttt{DDIM}}(t) = \texttt{DDIM-Forward}(\epsilon*, T \to t) \tag{B.4}$$

By plugging eq. (B.2) into eq. (B.3), we get:

$$x_0(t) = \frac{x(t) - \sqrt{1-\alpha(t)}\epsilon_\theta^t(x(t))}{\sqrt{\alpha(t)}} \tag{B.5}$$

As eq. (B.4) and eq. (B.5) resemble our assumption at time t (eq. (B.1)), the induction holds. Here we also include more 2D samples, shown in fig. B.3. It justifies the proof above empirically (see the first (DDIM) and second (DSD*) column).

Justification of Using One ϵ^* for a 3D Shape. In 2D, it is trivial that each image corresponds to one ODE starting from ϵ^*. Intuitively, each rendering of a 3D shape should have one unique ODE trajectory, thus more than one ϵ should be required for one 3D shape. However, finding the set of $\epsilon*$ is highly non-trivial and the set size is infinite.

Empirically however, as shown in fig. B.1, we observe that the ϵ^* for each object is actually close to each other. This enables us to approximate the set of ϵ of each object with one ODE starting point. Additionally, our interpolation approximation eq. (2.8) allows slight drift from the ODE, which makes the optimization of 3D feasible. To further reduce the approximation error, we utilize view-conditioned text prompts and the sampling strategy from prep-neg [1]. Specifically, for different views, the ODEs will start from the same point but are conditioned with corresponding views.

B.2 Implementation Details

Diverse Score Distillation in 2D. In 2D, when the number of DDIM sampling steps is not equal to the optimization steps, we can make slight modifications to eq. (2.8) and yield similar generation results (see the third column in fig. B.3). The

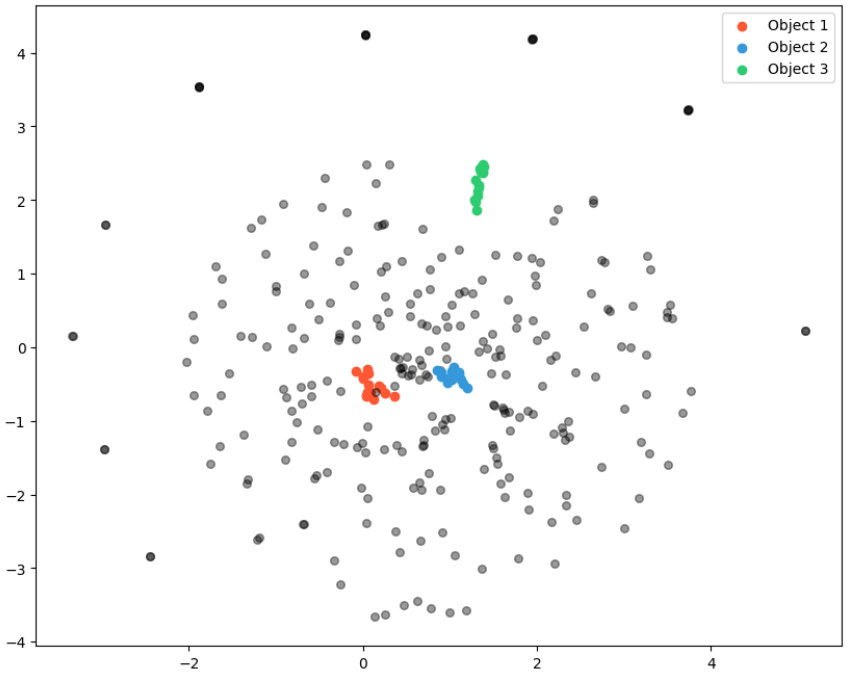

Figure B.1: **DDIM Inverted $\epsilon*$ for different objects, visualized using t-SNE.** Here we use DDIM inversion [56] to obtain the ODE starting points for different objects. Twenty objects are randomly sampled from Objavser [5] dataset, and we render 16 views for each object. It can be observed that the ODE starting points for the same object are closer to each other.

algorithm is shown in algorithm 2. The core idea is to simulate the discrete DDIM solver as closely as possible using the fixed DDIM sample timesteps t_{ddim}. For example, we will use the set $\{1000, 900, \ldots, 100\}$ when $N_{ddim} = 10$.

Diverse Score Distillation in 3D. algorithm 3 shows our method for 3D generation side by side with SDS (algorithm 4). We implement our code on Threestudio [11], a framework for score distillation. We use a maximum of 10 DDIM sampling steps during the optimization process. Our choice of δ is similar to [34], where $\delta = 0.1(t - t_{\texttt{MIN}})$. We use a cfg of 7.5 for $\epsilon_\theta^t(\mathbf{x}(t), y)$ and 1.0 for $\epsilon_\theta^{t+\delta}(\mathbf{x}(t+\delta), y)$, as we find this provides a better guidance in 3D. During the ODE-solving process, we use a constant cfg of 7.5. Our choice of 3D representation and geometrical regularization is identical to that of SDI [33] across experiments.

Algorithm 2 Diverse Score Distillation in 2D

Initialization: 2D image $\mathbf{x}_\pi$, trained text-to-image diffusion model ϵ_θ^t, prompt $\mathbf{y}$, ddim sampling steps N_{ddim}, optimization steps N, $\delta = T/N_{ddim}$, learning rate $lr = N_{ddim}/N$
$\epsilon^* \sim \mathcal{N}(0, I)$
for $i = 1$ to N **do**
  $t \leftarrow T(1 - i/N)$
  $t_\delta \leftarrow \min(t + \delta, T)$
  $t_{ddim} \leftarrow T(1 - \frac{\lceil t_\delta / N \rceil}{N_{ddim}})$
  $\mathbf{x}(t_{ddim}) \leftarrow \texttt{DDIM-Forward}(\epsilon^*, \mathbf{y}, T \rightarrow t_{ddim})$
  $\epsilon(t_{ddim}) \leftarrow \epsilon_\theta^{t_{ddim}}(\mathbf{x}(t_{ddim}))$
  $\mathbf{x}(t) \leftarrow \sqrt{\alpha(t)}\mathbf{x}_\pi + \sqrt{1 - \alpha(t)}\epsilon(t_{ddim})$
  $\mathbf{x}(t_\delta) \leftarrow \sqrt{\alpha(t_\delta)}\mathbf{x}_\pi + \sqrt{1 - \alpha(t_\delta)}\epsilon(t_{ddim})$
  $\nabla_{\mathbf{x}_\pi} \leftarrow lr \cdot \sigma(t)[\epsilon_\theta^t(\mathbf{x}(t), \mathbf{y}) - \epsilon_\theta^{t_\delta}(\mathbf{x}(t_\delta), \mathbf{y})]$
end for
return ψ

B.3 More 2D Results

More 2D Results. Additional 2D distillation results are presented in fig. B.3. Our method achieves greater diversity in the generated outputs by simulating the underlying ODE, whereas the baselines provide no explicit guarantee of diverse generation.

More 3D Results. We include more diverse 3D generation results in fig. B.2. Our method is capable of generating high-quality 3D shapes while maintaining diversity.

Algorithm 3 DSD

Initialization: 3D parameter ψ, trained text-to-image diffusion model ϵ_θ^t, prompt $\mathbf{y}$, set of cameras around 3D shape C, differential renderer g
$\epsilon^* \sim \mathcal{N}(0, I)$
for $i = 1$ to N **do**
  $t \leftarrow T(1 - i/N)$
  $\pi \leftarrow \text{Uniform}(\Pi)$
  $\mathbf{x}_\pi \leftarrow g(\psi, \pi)$
  $\mathbf{x}(t + \delta) \leftarrow$ `DDIM-Forward`$(\epsilon^*, \mathbf{y}, T \rightarrow t + \delta)$
  $\epsilon(t + \delta) \leftarrow \epsilon_\theta^{t+\delta}(\mathbf{x}(t + \delta), \mathbf{y})$
  $\hat{\mathbf{x}}(t + \delta) \leftarrow \sqrt{\alpha(t + \delta)}\mathbf{x}_\pi + \sqrt{1 - \alpha(t + \delta)}\epsilon(t + \delta)$
  $\hat{\mathbf{x}}(t) \leftarrow \sqrt{\alpha(t)}\mathbf{x}_\pi + \sqrt{1 - \alpha(t)}\epsilon(t + \delta)$
  $\nabla_\psi \leftarrow w(t)[\epsilon_\theta^t(\hat{\mathbf{x}}(t), \mathbf{y}) - \epsilon_\theta^{t+\delta}(\hat{\mathbf{x}}(t + \delta), \mathbf{y})]\frac{\partial g}{\partial \psi}$
end for
return ψ

Algorithm 4 SDS

Initialization: 3D parameter ψ, trained text-to-image diffusion model ϵ_θ^t, prompt $\mathbf{y}$, set of cameras around 3D shape C, differential renderer g
No fixed sample of ϵ
for $i = 1$ to N **do**
  $t \leftarrow \text{Uniform}(1, T)$
  $\pi \leftarrow \text{Uniform}(\Pi)$
  $\mathbf{x}_\pi \leftarrow g(\psi, \pi)$
  $\epsilon \sim \mathcal{N}(0, I)$
  $\mathbf{x}(t) \leftarrow \sqrt{\alpha(t)}\mathbf{x}_\pi + \sqrt{1 - \alpha(t)}\epsilon$
  $\nabla_\psi \leftarrow w(t)[\epsilon_\theta^t(\mathbf{x}(t), \mathbf{y}) - \epsilon]\frac{\partial g}{\partial \psi}$
end for
return ψ

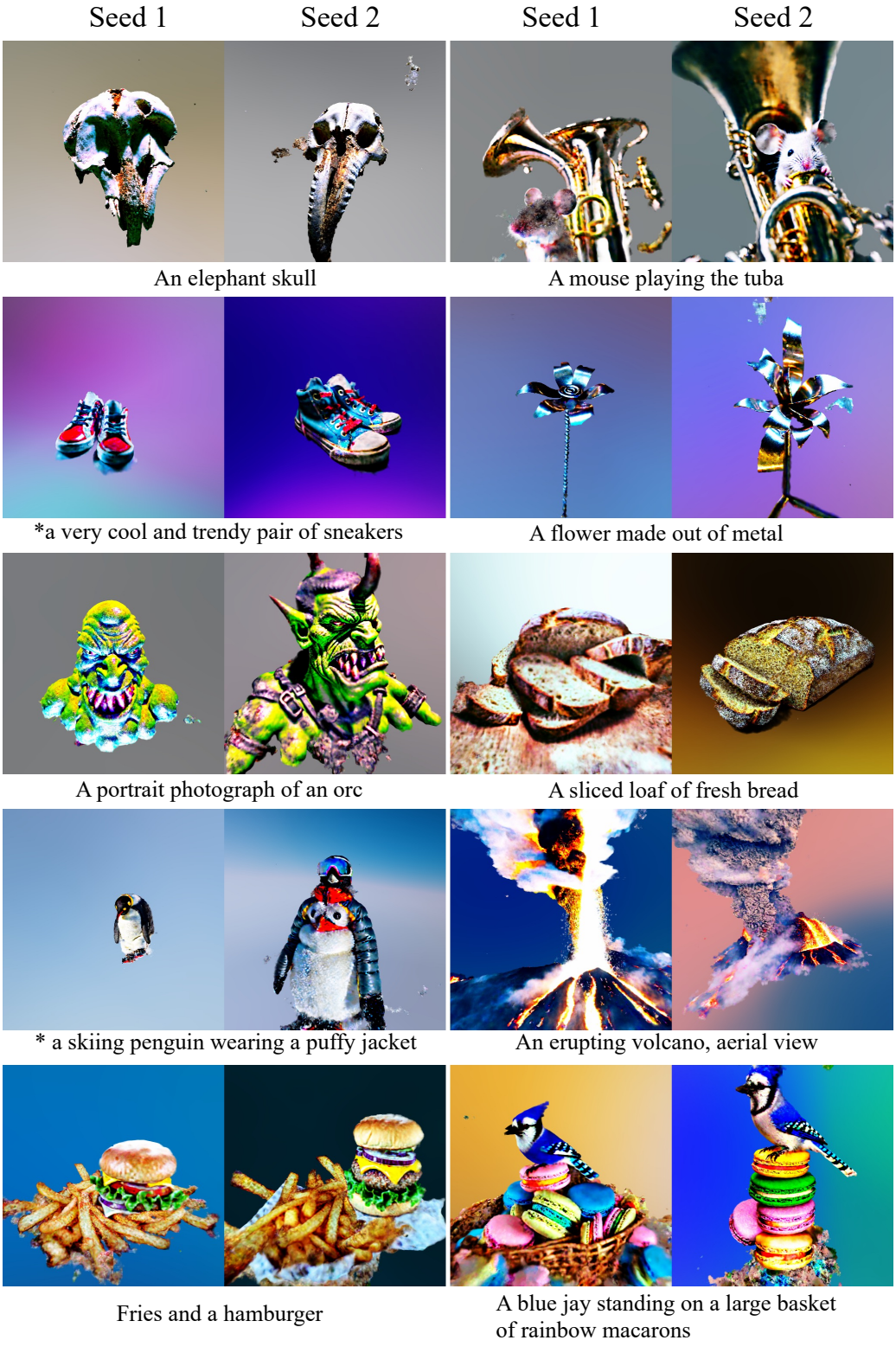

Figure B.2: **More 3D generation Results.** As different 3D shapes are generated with different ODEs, the results are diverse. * "A DSLR photo of".

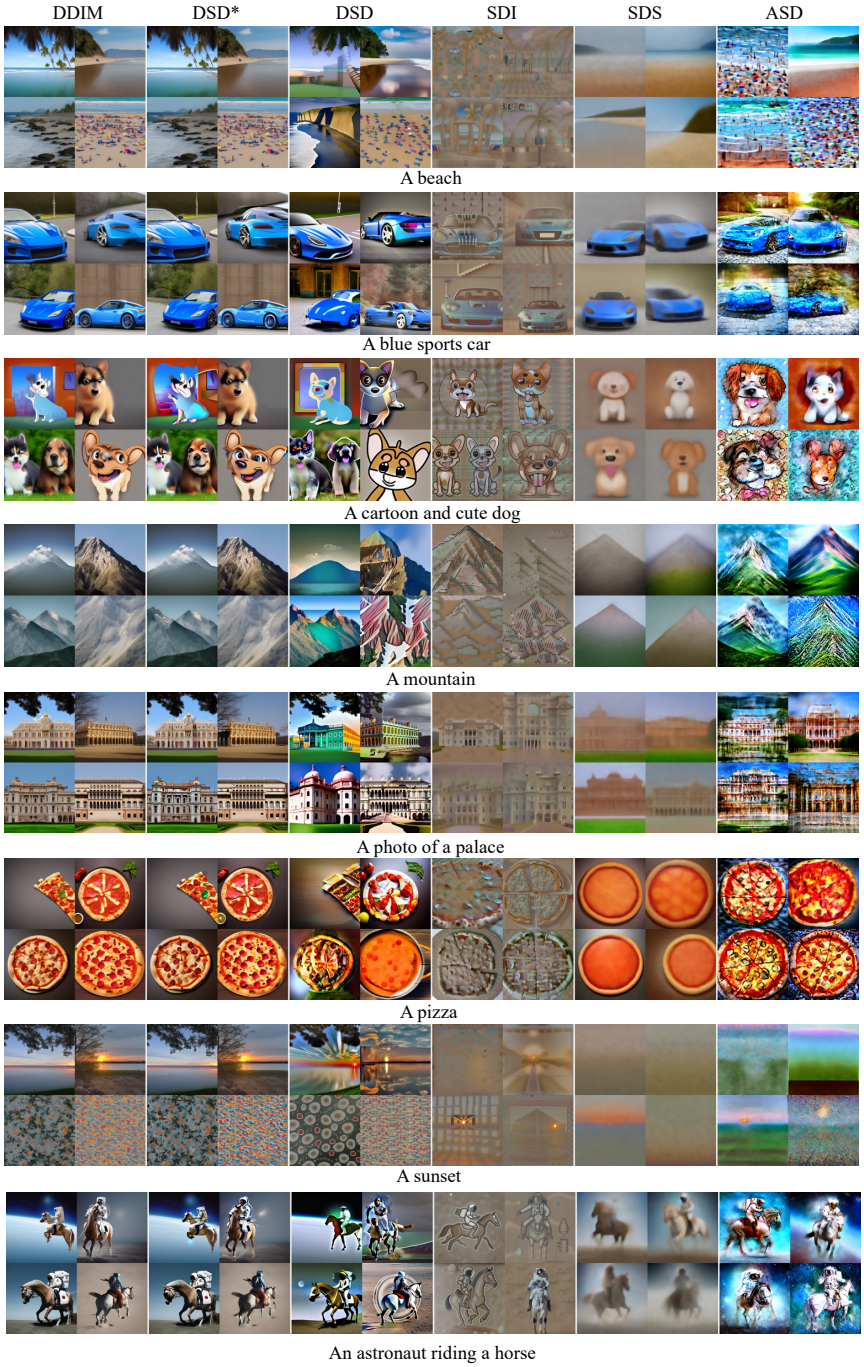

Figure B.3: **More 2D Distillation Results.** Our method can simulate the original ODE perfectly (DSD*) when the number of DDIM steps and optimization steps are equal. Ours (DSD) also generate diverse results by stimulating the underlying ODEs, while other baselines [33, 34, 44] have no explicit guarantee of diverse generation.

Bibliography

[1] Mohammadreza Armandpour, Ali Sadeghian, Huangjie Zheng, Amir Sadeghian, and Mingyuan Zhou. Re-imagine the negative prompt algorithm: Transform 2d diffusion into 3d, alleviate janus problem and beyond. In *arXiv preprint arXiv:2304.04968*, 2023. 2.3.3, B.1

[2] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *CVPR*, 2022. 2.2

[3] Zilong Chen, Yikai Wang, Feng Wang, Zhengyi Wang, and Huaping Liu. V3d: Video diffusion models are effective 3d generators. *arXiv preprint arXiv:2403.06738*, 2024. 2.1, 2.2

[4] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *IJRR*, 2023. 1.4.4

[5] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *CVPR*, 2023. (document), 2.1, 2.2, B.1

[6] Shrey Desai and Greg Durrett. Calibration of pre-trained transformers. In *EMNLP*, 2020. 1.1, 1.3.1

[7] Yiquan Duan, Xianda Guo, and Zheng Zhu. Diffusiondepth: Diffusion denoising approach for monocular depth estimation. In *European Conference on Computer Vision*, pages 432–449. Springer, 2024. 1.4.2

[8] Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011. A.1

[9] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. In *ICLR*, 2022. 2.2

[10] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *ICML*, 2017. 1.1, 1.3.1

[11] 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. 2.4, B.2

[12] Ayaan Haque, Matthew Tancik, Alexei A Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *CVPR*, 2023. 2.5

[13] 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. *Advances in neural information processing systems*, 30, 2017. 1.4.3

[14] 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. 2.4.2

[15] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 2.2, 2.3.4

[16] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. 2022. 1.1

[17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 2.3.1

[18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020. 1.3.1, 1.4.3

[19] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. In *NeurIPS*, 2022. 2.1

[20] Tsu-Ching Hsiao, Hao-Wei Chen, Hsuan-Kung Yang, and Chun-Yi Lee. Confronting ambiguity in 6d object pose estimation via score-based diffusion on se(3). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 352–362, 2024. 1.4.1

[21] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022. 2.4.2

[22] Yesukhei Jagvaral, Francois Lanusse, and Rachel Mandelbaum. DIFFUSION GENERATIVE MODELS ON SO(3), 2023. URL https://openreview.net/forum?id=jHA-yCyBGb. 1.4.1

[23] Haian Jin, Yuan Li, Fujun Luan, Yuanbo Xiangli, Sai Bi, Kai Zhang, Zexiang Xu, Jin Sun, and Noah Snavely. Neural gaffer: Relighting any object via diffusion.

In *NeurIPS*, 2024. 2.5

[24] Yash Kant, Aliaksandr Siarohin, Ziyi Wu, Michael Vasilkovsky, Guocheng Qian, Jian Ren, Riza Alp Guler, Bernard Ghanem, Sergey Tulyakov, and Igor Gilitschenski. Spad: Spatially aware multi-view diffusers. In *CVPR*, 2024. 2.1, 2.2

[25] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *NeurIPS*, 2022. 2.3.1, 2.3.1

[26] Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. *Advances in Neural Information Processing Systems*, 37:52996–53021, 2024. 1.1

[27] Oren Katzir, Or Patashnik, Daniel Cohen-Or, and Dani Lischinski. Noise-free score distillation. In *ICLR*, 2024. 2.2

[28] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1.4.2

[29] Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. In *CoRL*, 2024. 1.4.4

[30] Adam Leach, Sebastian M Schmon, Matteo T. Degiacomi, and Chris G. Willcocks. Denoising diffusion probabilistic models on SO(3) for rotational alignment. In *ICLR 2022 Workshop on Geometrical and Topological Representation Learning*, 2022. URL https://openreview.net/forum?id=BY88eBbkpe5. 1.4.1

[31] 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*, 2024. 2.2, 2.3.4

[32] 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. (document), 2.4.3, 2.7

[33] Artem Lukoianov, Haitz Sáez de Ocáriz Borde, Kristjan Greenewald, Vitor Campagnolo Guizilini, Timur Bagautdinov, Vincent Sitzmann, and Justin Solomon. Score distillation via reparametrized ddim. In *NeurIPS*, 2024. (document), 2.2, 2.3.1, 2.3.4, 2.4, 2.4.1, 2.4.2, 2.4.3, B.2, B.3

[34] Zhiyuan Ma, Yuxiang Wei, Yabin Zhang, Xiangyu Zhu, Zhen Lei, and Lei Zhang. Scaledreamer: Scalable text-to-3d synthesis with asynchronous score distillation. In *ECCV*, 2025. (document), 2.2, 2.3.4, 2.4, 2.4.2, 2.4.2, 2.3, 2.4.3, B.2, B.3

[35] Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot

manipulation. In *CoRL*, 2021. 1.4.4

[36] David McAllister, Songwei Ge, Jia-Bin Huang, David W Jacobs, Alexei A Efros, Aleksander Holynski, and Angjoo Kanazawa. Rethinking score distillation as a bridge between image distributions. In *NeurIPS*, 2024. 2.2

[37] Kieran A Murphy, Carlos Esteves, Varun Jampani, Srikumar Ramalingam, and Ameesh Makadia. Implicit-pdf: Non-parametric representation of probability distributions on the rotation manifold. In *Proceedings of the 38th International Conference on Machine Learning*, pages 7882–7893, 2021. 1.4.1

[38] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 1.2

[39] Jeremy Nixon, Michael W Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. Measuring calibration in deep learning. In *CVPR workshops*, 2019. 1.1, 1.3.1

[40] Evangelos Ntavelis, Aliaksandr Siarohin, Kyle Olszewski, Chaoyang Wang, Luc V Gool, and Sergey Tulyakov. Autodecoding latent 3d diffusion models. In *NeurIPS*, 2023. 2.1, 2.2

[41] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11410–11420, 2022. 1.4.3

[42] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023. 1.4.3

[43] 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. *arXiv preprint arXiv:2307.01952*, 2023. 1.4.3

[44] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *ICLR*, 2022. (document), 2.1, 2.2, 2.4, 2.4.2, 2.4.3, 2.4.3, B.3

[45] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 2021. 2.4.2

[46] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

1.4.3

[47] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *ICCV*, 2021. 2.1, 2.2

[48] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 1.4.3

[49] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2.1, 2.2, 2.4.3

[50] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022. A.1

[51] Saurabh Saxena, Abhishek Kar, Mohammad Norouzi, and David J Fleet. Monocular depth estimation using diffusion models. *arXiv preprint arXiv:2302.14816*, 2023. 1.4.2

[52] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3260–3269, 2017. 1.4.2, 1.2

[53] 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. *Advances in neural information processing systems*, 35:25278–25294, 2022. 1.4.3, 2.2

[54] Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv:2308.16512*, 2023. 2.1, 2.2, 2.4.3

[55] Andy Shih, Dorsa Sadigh, and Stefano Ermon. Long horizon temperature scaling. In *ICML*, 2023. 1.1, 1.3.1, 1.4.4

[56] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. (document), 1.2, 2.3.1, 2.3.4, B.1

[57] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2020. 1.4.3

[58] 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. 2.1, 2.3.1

[59] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *ICML*, 2023. 2.3.4

[60] 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*, 2023. 2.2

[61] Jianyuan Wang, Christian Rupprecht, and David Novotny. PoseDiffusion: Solving pose estimation via diffusion-aided bundle adjustment. 2023. 1.4.1

[62] Junjie Wang, Jiemin Fang, Xiaopeng Zhang, Lingxi Xie, and Qi Tian. Gaussianeditor: Editing 3d gaussians delicately with text instructions. In *CVPR*, 2024. 2.5

[63] 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. In *NeurIPS*, 2023. 2.2, 2.4, 2.4.3

[64] Zike Wu, Pan Zhou, Xuanyu Yi, Xiaoding Yuan, and Hanwang Zhang. Consistent3d: Towards consistent high-fidelity text-to-3d generation with deterministic sampling prior. In *CVPR*, 2024. 2.2, 2.3.4, 2.4, 2.4.2, 2.4.3

[65] Xin Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Song-Hai Zhang, and Xiaojuan Qi. Text-to-3d with classifier score distillation. In *ICLR*, 2024. 2.2, 2.4.3

[66] Yanjie Ze, Zixuan Chen, Wenhao Wang, Tianyi Chen, Xialin He, Ying Yuan, Xue Bin Peng, and Jiajun Wu. Generalizable humanoid manipulation with improved 3d diffusion policies. *arXiv preprint arXiv:2410.10803*, 2024. 1.4.4

[67] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. In *RSS*, 2024. 1.4.4

[68] Jason Y Zhang, Amy Lin, Moneish Kumar, Tzu-Hsuan Yang, Deva Ramanan, and Shubham Tulsiani. Cameras as rays: Pose estimation via ray diffusion. In *International Conference on Learning Representations (ICLR)*, 2024. 1.4.1

[69] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 2.4.2

[70] Zhizhuo Zhou and Shubham Tulsiani. Sparsefusion: Distilling view-conditioned diffusion for 3d reconstruction. In *CVPR*, 2023. 2.4.3

[71] Junzhe Zhu, Peiye Zhuang, and Sanmi Koyejo. Hifa: High-fidelity text-to-3d generation with advanced diffusion guidance. In *ICLR*, 2024. 2.2, 2.4, 2.4.3

[72] Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek

Joshi, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*, 2020. 1.4.4