Mean-Shift Distillation for Diffusion Mode Seeking

Vikas Thamizharasan
UMass Amherst
vthamizharas@umass.edu

Nikitas Chatzis
NTUA
nchatzis@tuc.gr

Iliyan Georgiev
Adobe Research
igeorgiev@adobe.com

Matthew Fisher
Adobe Research
matfishe@adobe.com

Evangelos Kalogerakis
TU Crete
UMass Amherst
kalogerakis@tuc.gr

Difan Liu
Adobe Research
diliu@adobe.com

Nanxuan Zhao
Adobe Research
nanxuanz@adobe.com

Michal Lukáč
Adobe Research
mike.k.lukac@gmail.com

Abstract

We present mean-shift distillation, *a novel diffusion distillation technique that provides a provably good proxy for the gradient of the diffusion output distribution. This is derived directly from mean-shift mode seeking on the distribution, and we show that its extrema are aligned with the modes. We further derive an efficient product distribution sampling procedure to evaluate the gradient. Our method is formulated as a drop-in replacement for score distillation sampling (SDS), requiring neither model retraining nor extensive modification of the sampling procedure. We show that it exhibits superior mode alignment as well as improved convergence in both synthetic and practical setups, yielding higher-fidelity results when applied to both text-to-image and text-to-3D applications with Stable Diffusion.*

1. Introduction

Soon after image diffusion [9] models exploded in popularity, DreamFusion [26] and SJC [33] concurrently introduced the idea of using them for image optimization. Intuitively, this can be expressed as the notion that images more likely to be generated by a diffusion model are "better" in the sense of being more faithful to the data distribution the diffusion model was trained on.

Formally, diffusion models provide a mechanism to sample images $x \in \mathcal{I}$ from a learned distribution $p(x)$. We then have a parameter vector $\vartheta \in \mathcal{P}$, along with an image-generating model $g : \mathcal{P} \to \mathcal{I}$. Given an initialization ϑ^0, we seek to optimize a ϑ^k such that $p(g(\vartheta^k)) > p(g(\vartheta^0))$. We expect this to yield an image $g(\vartheta^k)$ of higher quality, under the metric the diffusion model is trained for.

We could imagine optimizing ϑ by determining the gradient $\nabla p(g(\vartheta))$ and ascending along it. However, while we may use our diffusion model to sample from $p(x)$, we can neither easily evaluate $p(x)$ nor determine its gradient $\nabla_x p(x)$. Even though we can formally express $p(x)$ in terms of the score function $\epsilon(x, t)$ through the instantaneous change of variable formula [10], evaluating this formula requires calculating the divergence of the score function along the entire ODE path, making this of only theoretical interest. Evaluating the gradient of this quantity is even less practical.

Score distillation sampling (SDS) [26, 33] attempts to address this problem by offering proxies for the density gradient that are easier to estimate. However, their theoretical properties are not rigorously established, and SDS suffers from significant bias as well as variance, yielding inaccurate gradients. Examining the loss landscape of SDS in Fig. 1, we indeed see that not only are the maxima of this function not collocated with the modes of $p(x)$, but even in the simplest cases the loss creates "phantom modes" that are well out of distribution. Our method offers both better alignment with the distribution and lower variance of the gradient estimate.

Contributions. In this paper, we propose *mean-shift distillation*, a distribution-gradient proxy based on a well-known mode-seeking technique. Furthermore, we show that:

- This proxy can be implemented easily, with minimal changes to the diffusion sampling procedure;
- It evaluates with less variance than SDS with improved mode alignment;
- It has superior behavior, converging to modes of the trained distribution with a clear termination criterion.

2. Related Work

Denoising diffusion. In our work we rely most directly on the mean-shift method of mode seeking [7, 8], but our ability to apply it to diffusion rests on a body of theoretical analysis of this process.

Mathematically, denoising diffusion consists of solving an initial value problem (IVP) on a random variable from a simple, typically standard normal distribution, where the time-dependent gradient is learned by reversing the process of adding noise to the distribution being modeled [29, 30]. Already in these works authors suggest ways in which the output distribution may be manipulated by adding terms to the differential equation underlying the initial value problem, a property we will rely on to manipulate the output to our method's advantage.

A surprising connection between mean shift and diffusion emerged from the analysis of the optimal denoising model [6, 16, 17]. Since the forward (noising) process can be expressed as successive convolutions with a Gaussian kernel, the intermediate distributions are in fact Gaussian-kernel density estimates of the data distributions, with kernel bandwidth proportional to the time parameter. Therefore in the ODE of the reverse (inference) process, the gradient of the denoiser is theoretically equal to the mean-shift vector with appropriate kernel and bandwidth. Mean-shifting on the IVP time domain does not in fact seek modes of the output distribution, but we take advantage of this knowledge to implement mean shift on that domain.

Further related to the analysis of modes in particular, [5, 18] suggest that applying classifier-free guidance (CFG) [14] to diffusion has the effect of sharpening the modes of the output distribution. This guidance does not explicitly *seek* modes, but we have found that using CFG synergizes well with both SDS and our method.

Distilling diffusion priors. Score distillation sampling (SDS) [26, 33] has emerged as a useful technique for leveraging the priors learned by large-scale image models beyond 2D raster images. SDS provides an optimization procedure to estimate the parameters of a differentiable image generator, such that the rendered image is pushed towards a higher-probability region of a pre-trained prompt-conditioned image diffusion model. Originally proposed to optimize volumetric representations like NeRFs, it has been extended to other non-pixel-based representations [3, 15, 31, 40].

The tendency of SDS to produce over-smoothened results due to high variance is well documented. A plethora of works have been proposed to mitigate this behavior, e.g. to factorize the gradient to reduce the bias [2, 12, 19, 42], or to replace the uniform noise sampling in SDS with noise obtained by running DDIM inversion [22, 23]. SteinDreamer [34] propose a control variate for SDS, [38, 39] improve diversity of generations, and [35] alleviate the multi-view inconsistency problem.

VSD [36] tackle the low-fidelity problem by treating the target parameters as a random variable and estimate the variational distribution to produce diverse, high-fidelity results. SDI [23], on the other hand and unlike previous variance-reduction methods, find a better approximation of the added noise term, eliminating one of the root causes of the excessive variance. These methods attribute SDS/SJC to be mode-seeking; we show it is not and introduce mode-seeking behavior to address the excessive variance and low-fidelity results.

We draw the distinction between the above methods to knowledge distillation works designed for one-step inference [24, 37, 41]. Diff-Instruct [24] show that SDS is a special case of their distillation formulation when the generator's output is a Dirac's Delta distribution and the marginal of the variational score is Gaussian. While the derived gradients resemble SDS and VSD, these methods require training an auxiliary score network to estimate the variational score— challenging to generalize beyond text-to-image generation— and have been targeted for different use cases, namely faster inference, model compression, and dataset privacy-preserving.

3. Mean-Shift Distillation

In this section we derive the mean-shift vector for the diffusion output distribution, and show how it approximates the gradient thereof. We further show how an efficient estimate of this vector may be obtained with a minimal modification of diffusion sampling. We begin with a motivation of our development by illustrating the pitfalls of SDS.

3.1. Motivation

Given a pre-trained diffusion model ϵ_ϕ, the SDS loss penalizes the KL-divergence of a unimodal Gaussian distribution centered around x and the data distribution $p_\phi(z_t; c, t)$ captured by the frozen diffusion model conditioned on text embeddings c. With $x = g(\vartheta)$, an image rendered by ϑ via a differentiable renderer g, [26] derive the gradient of the loss $\mathcal{L}_{\text{SDS}}$ with respect to ϑ:

$$\nabla_\vartheta \mathcal{L}_{\text{SDS}} = \mathbb{E}_{t,\epsilon}\Big[\alpha(t)\left(\epsilon_\phi(\alpha(t)x + \epsilon; t) - \epsilon\right)\Big]\frac{\partial x}{\partial \vartheta},$$
$$\text{with } t \sim U(0,T), \epsilon \sim \mathcal{N}(0, \sigma(t)I). \quad (1)$$

To illustrate the pitfalls of SDS, we simulate it in 2D using a small denoising diffusion network (Fig. 1). This allows us to set $\vartheta = x \in \mathbb{R}^2$ (where g becomes an identity map). We construct a fractal-like dataset as shown by [18], with analytic ground-truth probability density and score. This data distribution is a mixture of highly anisotropic Gaussians, where most of the probability mass resides in narrow regions, emulating the low intrinsic dimensionality of natural images [4, 27]. For a baseline, we compare it with DDIM [29], a popular first-order sampling algorithm, with classifier-free guidance (CFG) [14]. More details can be found in Sec. 4.2.

It is immediately apparent how even in this simple setting, the optima to which SDS converges do not model the output distribution well. Furthermore, the convergence itself is problematic due to very high variance of SDS, which we will address later.

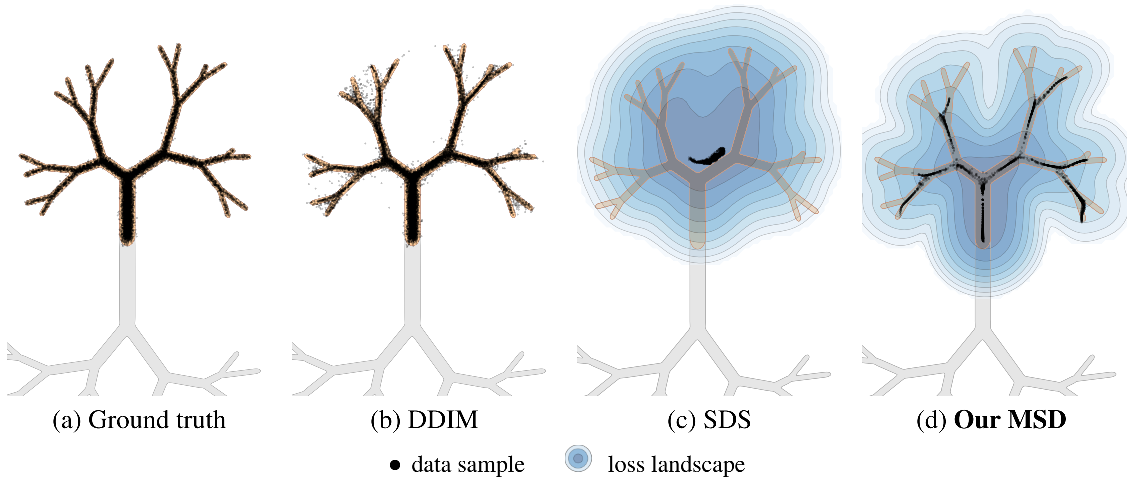

Figure 1. Mode-seeking simulated in a fractal-like 2D distribution with two (orange, gray) classes, adapted from [18]. We compare the behavior of diffusion sampling (DDIM) to optimization-based diffusion distillation, in a class-conditional setting. With class=orange, **(a)** Ground truth distribution, **(b)** DDIM sampling , **(c)** SDS, and **(d)** our MSD. All methods are run without guidance.

3.2. Mean-shift Gradient Approximation

We start by convolving the data density p with a radial Gaussian kernel $G_\lambda(x) = c_\lambda e^{-x^2/\lambda^2}$ with bandwidth λ, normalized by a constant c_λ. This convolution yields a smoothed density $p^*_\lambda(x)$:

$$p^*_\lambda(x) = p * G_\lambda(x) = \int G_\lambda(x-y)p(y)dy. \tag{2}$$

We now take the gradient $\nabla_x p^*_\lambda(x)$ of the smoothed density and substitute the Gaussian kernel's gradient:

$$\begin{aligned}\nabla_x p^*_\lambda(x) &= \int \nabla_x G_\lambda(x-y)p(y)dy \qquad (3)\\ &= \int c_\lambda(x-y)G_\lambda(x-y)p(y)dy. \qquad (4)\end{aligned}$$

We then take the stationary-point equation and reorganize it as a fixed-point iteration:

$$\nabla_x p^*_\lambda(x) = 0 \implies x' = \frac{\int yG_\lambda(x-y)p(y)dy}{\int G_\lambda(x-y)p(y)dy}, \tag{5}$$

where the constant c_λ cancels out. We will discuss the practical estimation of the integrals in Sec. 3.3 below.

The iterative process in Eq. (5) is a continuous version of mean shift [8]. We may turn this into gradient proxy with several desirable properties. Defining the mean-shift vector $\vec{m}(x) = x' - x$, it follows from Eq. (3) that

$$\vec{m}(x) \propto \nabla_x p^*_\lambda(x)/p^*_\lambda(x). \tag{6}$$

Since the smoothed density $p^*_\lambda(x)$ is always non-negative, $\vec{m}(x)$ is always aligned with its gradient $\nabla p^*_\lambda(x)$. It is also aligned with the gradient of the true density p as $\lambda \to 0$ (when such gradient exists). This means that a differential step along the vector $\vec{m}(x)$ will improve the likelihood of x, making this a good proxy for the kernel density estimation gradient. Furthermore, it implies that $\vec{m}(x)$ will be zero at the modes of $p^*_\lambda(x)$, giving us a convergence criterion.

3.3. Gradient Estimation via Product Sampling

The integrals in Eq. (5) can be both estimated using samples y from the density p; such estimation yields the classical mean-shift expression

$$x' = \frac{\sum_{y_i \sim p} G_\lambda(x-y_i)y_i}{\sum_{y_i \sim p} G_\lambda(x-y_i)}. \tag{7}$$

In our case, we do not have such samples readily available. We could in theory use images from the training dataset as these samples, or else use the diffusion model to generate them – either as a pre-process or on-the-fly during iteration. Unfortunately, that would be prohibitively costly as the datasets are typically quite large and accurate estimation would require a very large number of samples for practical (i.e. small) kernel bandwidths λ.

Our key insight is that the right-hand side of Eq. (5) can be viewed as an expectation with respect to a density $\dot{p}_\lambda$ that is the product of p and the kernel G_λ centered at x:

$$x' = \int y\, \dot{p}_\lambda(y|x)dy \;=\; \mathbb{E}_{y\sim\dot{p}_\lambda(y|x)}[y]. \tag{8}$$

To generate samples y from this product density, we exploit the fact that diffusion models employ score-based sampling [9, 30]. Instead of using the score $\nabla \log p$ of the density p in DDIM sampling, we use the score of our product density:

$$\begin{aligned}\nabla_{z_t}\log(\dot{p}_\lambda(y|x)) &= \nabla_{z_t}\log p(z_t) + \nabla_{z_t}\log G_\lambda(x-z_t)\\ &= \nabla_{z_t}\log p(z_t) - \frac{x-z_t}{\lambda^2}, \qquad (9)\end{aligned}$$

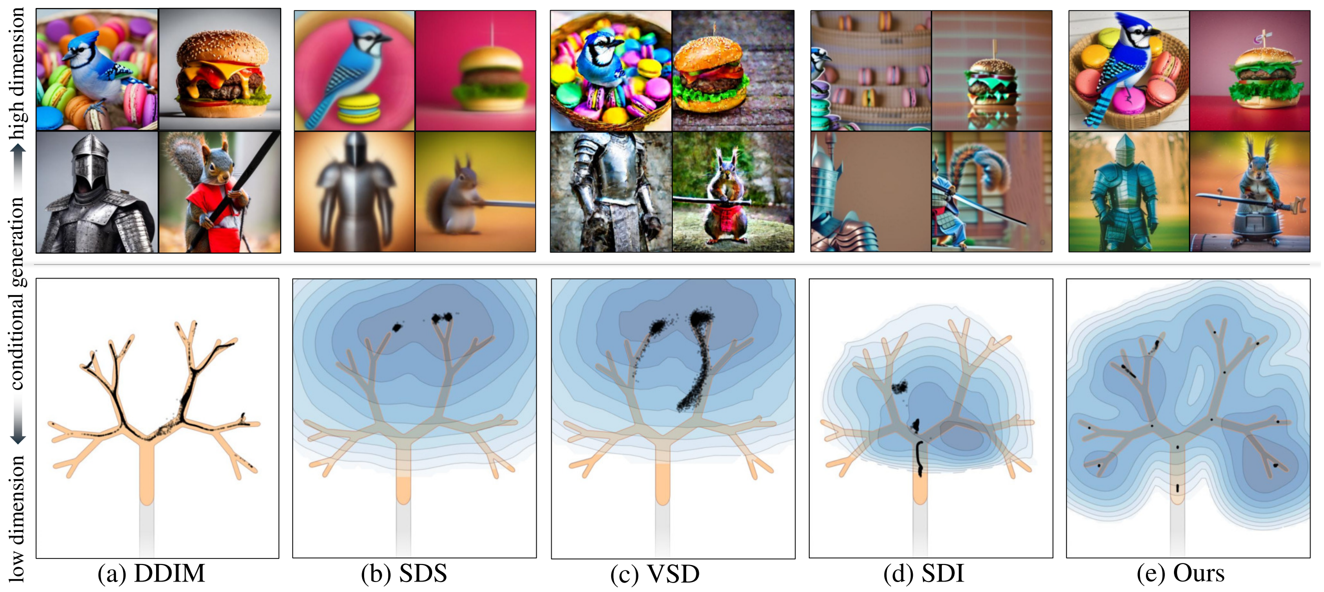

Figure 2. We juxtapose diffusion sampling vs diffusion distillation in low-dimensional ($\mathbb{R}^2$) and high-dimensional ($\mathbb{R}^{64\times64\times4}$) setting, using guidance via CFG [14]. **Top**: **(a)** text-conditioned generation of image via DDIM with 32 steps, **(b) - (e)** optimized coordinate-based neural implicit image for SDS, VSD, SDI, and our MSD respectively with StableDiffusion (CFG=7.5, § 4.4). **Bottom**: **(a)** class-conditioned generation of 2D points via DDIM with 32 steps, **(b) - (e)** optimized 2D points for SDS, VSD, SDI, and our MSD respectively (CFG=4, § 4.2). Text-prompts in clockwise order: *"A DSLR photo of a ... hamburger*, *squirrel dressed as a samurai weighing a katana*, *knight in silver armor*, and *bluejay on basket of macarons"*.

where $z_t = \alpha(t)x + \epsilon$. This is the sum of the density score (provided by the diffusion model) and the score of our Gaussian kernel. Having the ability to generate samples y_i from the product density, we can estimate the mean-shift iterate (8) as

$$x' \approx \frac{1}{N} \sum_{y_i \sim \hat{p}_\lambda(y|x)} y_i. \tag{10}$$

In practice we use a single sample y, which simplifies our mean-shift vector to

$$\vec{m}(x) = y - x. \tag{11}$$

We can thus step along $\vec{m}$ to seek the modes of the data density p. Substituting a learned score model into 9 gives us

$$\hat{\epsilon}_t = \epsilon_\theta(z_t; t) - \frac{x - z_t}{\lambda^2}. \tag{12}$$

3.4. Practical Considerations

As noted in SJC [33], distillation with unconditioned diffusion models is challenging in high-dimensional settings like images. While we show unconditioned diffusion distillation is practical in simple 2D toy datasets below, we operate in the conditional setting throughout.

Impact of guidance. Conditional score estimates from diffusion models, $\epsilon_\theta(z_t, c) \approx -\sigma_t \nabla_{z_t} \log p(z_t|c)$, are improved in practice with classifier-free guidance (CFG) [14], which sharpens the distribution around the modes:

$$\tilde{\epsilon}_\theta(z_t, c) = (1 + w)\epsilon_\theta(z_t, c) - w\epsilon_\theta(z_t). \tag{13}$$

We may directly substitute this for the denoiser term in Eq. (12). Despite its practical success, the denoising direction induced by CFG does not provide theoretical guarantees in producing samples from $p_{0,w}(z_t|c)$ [5]. Even in simple settings, as observed in Fig. 1(b), CFG can lead to mode drops. While alternative guidance strategy exists [18], we stick with the dominant practice of using CFG (Eq. (13)). We have found that this synergizes well with mode-seeking by mean-shift, and show the effects of this in evaluation below. See discussion in Supplemental §B.

Integrating kernel score. Because the magnitude of the kernel term in Eq. (12) can be quite high when $|y - z_t|$ is high relative to λ, directly implementing this can result in instability while denoising particularly with explicit integrators. Higher-order integrators are generally capable of dealing with this instability, but require many more score function evaluations.

To address this, we note that in isolation the kernel term has the form of a negative exponential centered on y, or explicitly:

$$z_{t+\Delta t} = y + (z_t - y)e^{\frac{\Delta t}{\lambda^2}}, \tag{14}$$

where Δt is negative. We take advantage of this to formulate a stable approximation that avoids the stability issues with

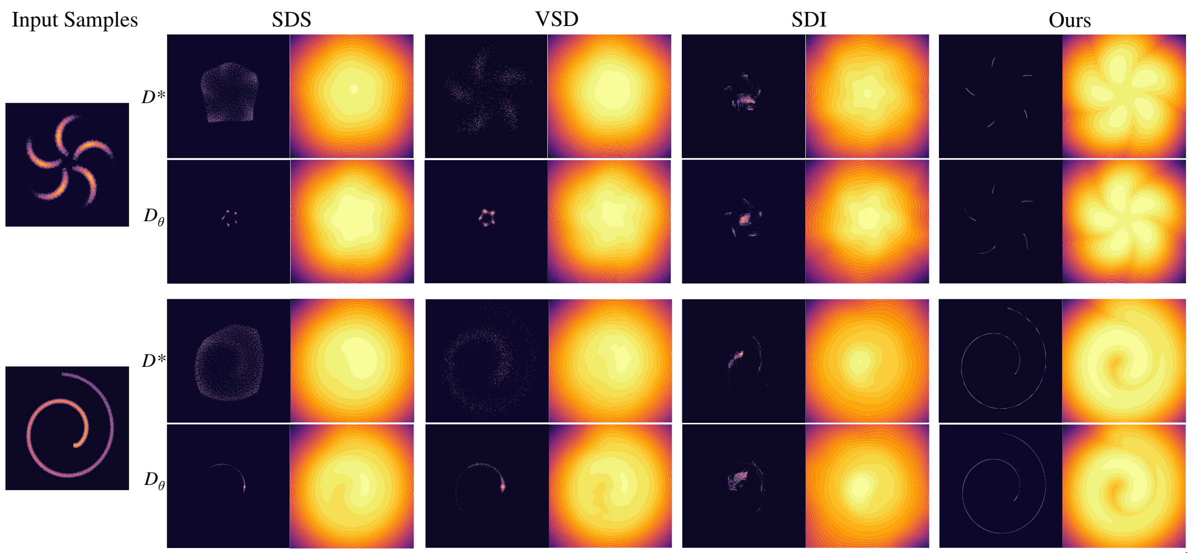

Figure 3. Unconditional distillation on two toy density datasets, *Pinwheel* (top) and *Spiral* (bottom), given an ideal denoiser (D^*) and a learned denoiser (D_θ). For each method and denoiser, we show the optimized samples (left) and the loss landscape (right). *Zoom in for clarity*.

a minimal change to the integration process. Instead of feeding the full composite score function to the integrator, in each time step we first integrate only the score function with the existing integrator to get $z'_{t+\Delta t}$. Immediately after, we separately account for the kernel term by computing the final output as

$$z_{t+\Delta t} = y + (z'_{t+\Delta t} - y)e^{\frac{\Delta t}{\lambda^2}}. \tag{15}$$

A full derivation is provided in the supplemental. We note such numerically instability has been observed when using high CFG values. A remedy is to apply guidance in a limited interval [20]. We leverage similar ad-hoc tricks by applying the kernel term in limited interval through the sampling chain.

4. Practical Implementation and Evaluation

In this section, we construct synthetic examples on which we demonstrate that our proposed method behaves as theory predicts, alleviating the issues SDS exhibits even in these simple scenarios. We further explain the issues encountered when translating this theory into practice, and describe adaptations we designed to make our method work with real-world diffusion models, retaining desirable properties. We make comparisons with two strong baselines that improve the convergence and performance of SDS; SDI [23], who propose a better noise term to reduce late-stage stochasticity, yet, retain the same gradient computation of SDS, and VSD [36], who propose to learn the variational score as opposed to assuming it to be a known analytic score like in SDS.

4.1. Idealized Setting

In order to manage large data dimensionality as well as massive training datasets, diffusion in practice employs a trained neural network to represent the denoiser D. However, [16, 17] have identified an analytical solution to minimizing the denoiser error, the *ideal denoiser* $D^*(x; t)$:

$$D^*(x;t) = \frac{\sum_i u_i \mathcal{N}(x; u_i, \sigma(t))}{\sum_i \mathcal{N}(x; u_i, \sigma(t))}, \tag{16}$$

where $u_0 \dots u_n$ are samples in our training set. Attentive readers will notice that this is in fact the discrete mean shift formula [8], with training samples taking the place of data samples and noise magnitude $\sigma(t)$ taking place of the kernel bandwidth λ. This expression is feasible to compute in practice for small datasets, and by setting

$$\epsilon^*_\phi(z_t; t) = -\frac{D^*(z_t; t) - z_t}{\sigma_t},$$

we may substitute it into the SDS formula (1) to get an explicit solution for the SDS gradient

$$\nabla_x \mathcal{L}_{\text{SDS}} = \mathbb{E}_{t, z_t \sim \mathcal{N}(\alpha_t x, \sigma_t^2 \mathbf{I})}\left[w(t) \frac{z_t - D^*(z_t; \sigma_t)}{\sigma_t} \frac{\partial x}{\partial \theta} \right].$$

We can brute force numerically evaluate this integral. We can compare both methods on synthetic datasets, eliminating any error introduced by training and evaluating a neural model to show that the theoretical properties hold.

4.2. Toy Distributions in $\mathbb{R}^2$

In addition to the fractal dataset (Fig. 1), we extend our analysis to other 2D datasets, with ${u_i}_{i=1}^m \subset \mathcal{M} \in \mathbb{R}^2$ sampled from various challenging toy 2D densities [28, 32]. For each, we sample 10^4 points from the data distribution and initialize our target points densely across a grid $[-1.5, 1.5]^2$.

Table 1. Metrics for class-conditional distillation on 2D Fractal dataset. For each metric, *top* to *bottom*: ideal denoiser (D^*), learned denoiser without guidance (D_θ no CFG), learned denoiser with CFG (D_θ w/CFG) [14], and learned denoiser with Autoguidance (D_θ w/AG)[18]. 1st and 2nd best among distillation-based methods for each column, highlighted.

Two-class Fractal dataset

Method	**NLL** $\downarrow$	**MMD** $\downarrow$	**Precision** $\uparrow$	**Recall** $\uparrow$
DDIM-D^*	-1.85	0.860	0.97	0.93
SDS-D^*	36.15	328.0	0.08	0.03
VSD-D^*	9.97	230.9	0.05	0.02
SDI-D^*	24.28	29.93	0.27	0.01
MSD-D^*	-1.32	30.46	0.92	0.33
DDIM-D_θ no CFG	-1.51	0.007	0.95	0.96
SDS-D_θ no CFG	9.12	87.04	0.01	0.00
VSD-D_θ no CFG	9.88	94.68	0.10	0.05
SDI-D_θ no CFG	-2.87	459.9	0.97	0.12
MSD-D_θ no CFG	-2.02	12.79	0.97	0.42
DDIM-D_θ w/CFG	-1.59	257.43	0.97	0.44
SDS-D_θ w/CFG	15.96	3875.11	0.17	0.03
VSD-D_θ w/CFG	18.97	3845.41	0.21	0.05
SDI-D_θ w/CFG	27.37	69.23	0.30	0.48
MSD-D_θ w/CFG	-1.15	133.41	0.94	0.40
DDIM-D_θ w/AG	-1.67	0.25	0.96	0.79
SDS-D_θ w/AG	11.33	71.05	0.04	0.03
VSD-D_θ w/AG	11.52	70.25	0.03	0.03
SDI-D_θ w/AG	0.65	15089	0.51	0.51
MSD-D_θ w/AG	-1.99	122.94	0.97	0.43

With 10^3 Monte Carlo samples, we benchmark SDS, VSD, SDI, and our MSD using both an ideal denoiser (16) and a learned denoiser (13). For the fractal dataset the denoiser is class-conditioned; its score is either left unchanged (without guidance) or guided via CFG or Autoguidance. For the other datasets, the denoiser is unconditioned.

We visualize the generated samples produced by all methods after the optimization in Figs. 2 and 3. We also visualize the reconstructed loss functions. This makes the behavior of all methods particularly obvious; the peaks of this reconstructed function are out of distribution for all methods except our MSD. Numerical evaluations in Tabs. 1 and 2 show our method outperform baselines. We suspect that this bias persists in SDS in higher dimensional settings and is what causes SDS optimized results to be blurry and exhibit other artifacts (top row of Fig. 2). More experimental details can be found in Sec. 4.5 and the supplemental.

Beyond bias, we also track the gradient-estimate variance, a key driver of convergence. Optimizing with stochastic gradients resembles a random walk: large variance slows progress, can push steps opposite the true gradient, and, once near an optimum, keeps the iterate oscillating instead of settling. To quantify the variance of an

Table 2. Metrics for unconditional distillation on 2D toy datasets. For each metric, *top* to *bottom*: ideal denoiser (D^*) and learned denoiser (D_θ). MMD scaled by 10^{-4}.

Unlabeled Spiral dataset

Method	**NLL** $\downarrow$	**MMD** $\downarrow$	**Precision** $\uparrow$	**Recall** $\uparrow$
DDIM-D^*	-1.39	0.410	0.97	0.93
SDS-D^*	30.37	13.85	0.02	0.03
VSD-D^*	10.15	23.46	0.04	0.09
SDI-D^*	35.64	39.51	0.10	0.90
MSD-D^*	-1.28	4.490	0.99	0.18
DDIM-D_θ	-1.32	1.160	0.96	0.96
SDS-D_θ	8.13	274.3	0.04	0.11
VSD-D_θ	8.90	271.8	0.07	0.14
SDI-D_θ	19.16	2008	0.12	0.42
MSD-D_θ	-1.51	18.41	0.98	0.18

Unlabeled Pinwheel dataset

Method	**NLL** $\downarrow$	**MMD** $\downarrow$	**Precision** $\uparrow$	**Recall** $\uparrow$
DDIM-D^*	-1.19	1.05	0.97	0.94
SDS-D^*	2.29	5.18	0.85	0.03
VSD-D^*	3.34	6.78	0.65	0.04
SDI-D^*	28.31	6.13	0.17	0.001
MSD-D^*	-1.94	5.83	0.99	0.01
DDIM-D_θ	-1.1	0.270	0.97	0.97
SDS-D_θ	2.00	36.37	0.90	0.01
VSD-D_θ	2.28	33.36	0.97	0.02
SDI-D_θ	17.33	98.09	0.51	0.15
MSD-D_θ	-2.19	7.250	0.99	0.13

estimate $\hat{g}(x)$ of the gradient $g(x)$, we employ a slight variation of the Monte Carlo estimator efficiency formula $\varepsilon(\hat{g}(x)) = |g(x)|^2 / (\text{MSE}(\hat{g}(x))\,\text{cost}(\hat{g}(x)))$. We measure cost as number of invocations of the score model, since that is the typical bottleneck in diffusion.

Normalization by the squared norm of g is included to account for the fact that, due to bias and scaling, different estimators may converge to gradients of different magnitude, and the normalized MSE then roughly describes the probability of the estimated gradient pointing the "right" way. MSE and cost are accumulated over many independent estimations, and average over many values of x.
The result of these efficiency comparisons are in Tab. 3 (in log-scale). Although getting a single estimate with our method requires more score model invocations, the efficiency of our method is significantly higher than SDS and VSD, and comparable to SDI.

4.3. Practical Setting

For large-scale image datasets, idealized denoiser is no longer tractable and we contend with a learned denoising function, and the associated machinery. This introduces numerical issues. Namely, the magnitude of the kernel term may grow to where the standard first or second order inte-

Table 3. Efficiency (↑) on 2D toy density datasets. *Left*: ideal denoiser / *Right*: learned denoiser.

	Fractal	**Spiral**	**Pinwheel**
SDS	-7.37 / -6.89	-8.48 / -7.57	-7.82 / -6.99
VSD	-5.92 / -3.83	-6.85 / -4.45	-6.36 / -3.93
SDI	14.18/ 14.21	13.94 / 14.17	14.19 / 14.18
MSD	13.44 / 7.65	13.38 / 6.32	13.76 / 7.08

grators can no longer manage it (Sec. 3.4); but conversely, so does the magnitude of the learned score when z_t is far out of distribution, because the ideal denoiser (Sec. 4.1) uses the same equation as mean shift. Start of the optimization, it is likely in a high-dimensional space that x will be out of distribution and we have to choose between the integration failing because the denoiser term has a high magnitude, or because the kernel term has a high magnitude.

To alleviate this, we use two heuristic approximations: applying guidance in limited interval (Sec. 3.4) and scaling our sample in Eq. (15) by noise corresponding to time step t. In practice, we apply inversion to get the latter. These are designed to keep the iterate in a region with reasonable score magnitude and still sample a distribution that is an approximation of the product distribution.

4.4. Pre-trained Stable Diffusion

We use the latent-space diffusion model, Stable Diffusion, as the diffusion prior for text-conditioned optimization of parameters of differentiable image generators. Specifically, we optimize parameters ϑ of generator g, a rendering function that maps ϑ to an image $\mathcal{I}$. The rendered image $\mathcal{I}$ is fed to the image encoder to get x^k, our latent at optimization step k, over which the gradient is computed. We define two settings where ϑ (1) represents an RGB image, and (2) represents a 3D volume. Specifically:

1. **Text-to-2D.** We represent 2D images via a coordinated-based MLP f with learnable parameters ϑ that takes as input a 2D point p in the unit square $p = (x, y) \in [0, 1]^2$ and outputs RGB $\in [0, 1]^3$; $f(p; \vartheta) : \mathbb{R}^2 \to$ RGB. We use this non pixel-based representation of an image for two reasons, (1) to prevent our method and the baselines from taking the exact gradient step i.e. running diffusion sampling and setting x^k to the denoised latent z_0, and (2) we can directly compare with images sampled via DDIM, an unconstrained image generation setting.
2. **Text-to-3D.** We represent 3D volumes as NeRFs, following [26]. The NeRF is parameterized by two MLPs, one for foreground and one for background. The former has 64 hidden nodes and 2 layers, with input (x, y, z) coordinates encoded via HashGrid [25].

Implementation details. We implement all our code in PyTorch, on a single NVIDIA A100 gpu. We use the Threestudio [11] framework for experiments involving pre-trained Stable Diffusion. We use AdamW optimizer with lr$= 10^{-2}$. We set optimization steps to 400 for text-to-2D and $10k$ for text-to-3D. We use a monotonically decreasing schedule for the bandwidth λ.

Table 4. Text-to-2D quantitative comparison. We evaluate fidelity with FID and CLIP-SIM. †FID measured with DDIM as ground truth.

Method	**FID ↓**	**CLIP-SIM (L/14) ↑**
DDIM†	–	44.1± 2.8
SDS	198.90	27.7± 1.9
VDS	130.22	30.8± 1.4
SDI	166.16	31.0± 0.7
MSD	114.12	32.6± 0.8

4.5. Evaluation

Dataset. We use a subset of the prompts curated by [12, 26]. We include all prompts in the supplemental.

Metrics. For toy density dataset (Sec. 4.2), we compute negative log-likelihood scores (NLL), generative precision and recall [21], and maximum mean discrepancy (MMD). For text-to-2D, we use images produced by DDIM to represent the ground truth distribution. To evaluate fidelity of the images, FID [13] is computed for each baseline (SDS, VSD, SDI) and ours against this ground truth image set, across 10,000 images. We also compute CLIP scores [1] to measure prompt-generation alignment.

Quantitative comparisons. Table 4 reports results for FID and CLIP-based similarity, comparing our method with SDS, VSD, and SDI. We outperform all baselines in image fidelity and achieve faster convergence, as measured via *fid* vs *iterations* (see supplemental).

Qualitative comparisons. Figure 2 (top row) and supplemental figure 3 compares our method with SDS, VSD, and SDI on text-to-2D generation, qualitatively. We show the importance of the two heuristics (Sec. 4.3) to resolve numerical instabilities, absence of which can result in visual artifacts in supplemental figure 4. SDS, as discussed, produces low-fidelity results while SDI's inversion accumulates numerical errors during early stages of optimization. In Fig. 4, we qualitatively compare results for text-to-3D optimization. We restrict to qualitative comparison for this task as quantitative metrics have high variance due to the absence of a ground truth dataset.

Impact of bandwidth. Figure 5 shows the impact of the bandwidth (λ) term on the denoising process. First, we sample three parameters $\{\vartheta^k\}$ from our text-to-2D optimization pipeline at iterations $k = \{100, 200, 400\}$ and also sample three discrete λ values $\{\lambda_1 \ll \lambda_2 \ll \lambda_3\}$. Then, we run our forward pass once for each λ_i, independently. We visualize four decoded denoised latents z_0 (with different random

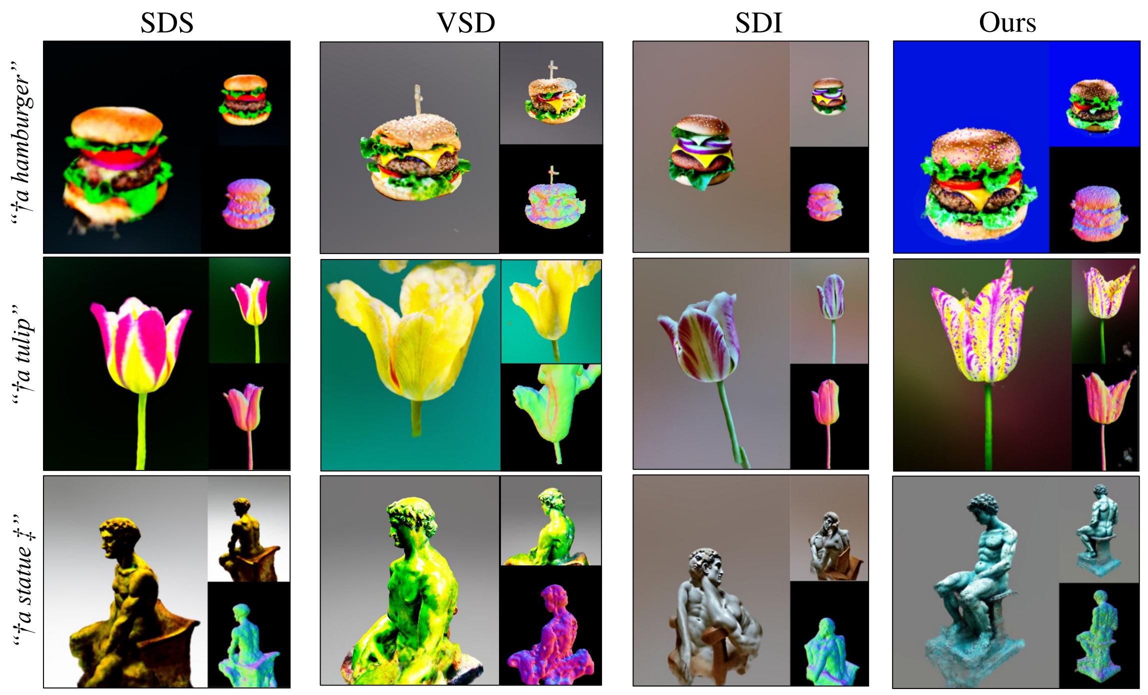

Figure 4. Comparison of 3D generation with other score distillation methods. Full prompt: † *"A DSLR photo of a ..."*, ‡ *"a Michelangelo statue of a man on a chair"*.

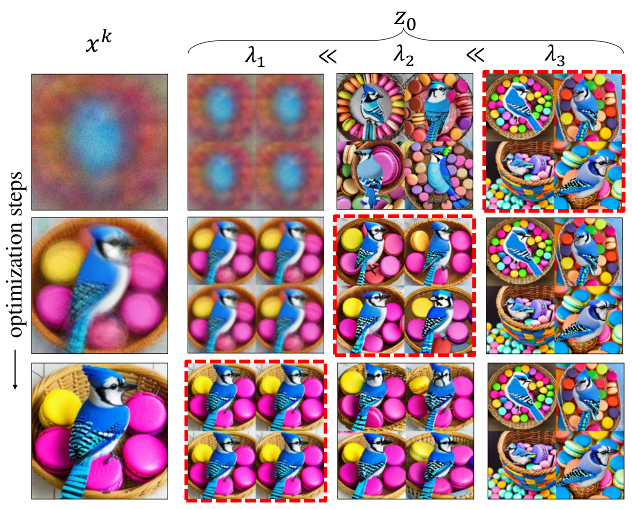

Figure 5. Impact of bandwidth (λ) on the denoised latent (z_0). We set $\lambda_3 = 10^3$, $\lambda_2 = 10$, $\lambda_1 = 10^{-2}$. Highlighted images show the optimal bandwidth value corresponding to the k^{th} optimization.

seeds). The highlighted images show the optimal choices of λ for each x^k (the encoded latent for ϑ^k). At high bandwidth value λ_3, the influence of the kernel term in the product sampling is negligible. This degenerates to vanilla denoising and we observe high variance in the output, irrespective of our current x^k. This is ideal at early stages of optimization. As bandwidth is annealed, we observe reduction in variance. Yet, the quality of the outputs can degrade if the kernel term dominates while x^k is not *"in-distribution"*. As x^k approaches the mode of the distribution corresponding to the input text-prompt at final stages of optimization (when $k = 400$), with a low bandwidth λ_1, our denoised latent $z_0 \approx x^k$. This provides us with a convergence criteria and we terminate when λ is below the threshold λ_1.

5. Conclusion

In this paper, we have reframed diffusion distillation in terms of explicitly ascending the gradient of the data distribution. We have derived mean-shift distillation as a proxy that provably aligns with this gradient, and in the limit its maxima are collocated with the modes of the data distribution. We have demonstrated that compared to SDS, this method achieves better mode alignment as well as lower gradient variance, which in practice translates to more realistic optimization results as well as improved convergence rate. Since this method simply provides optimization gradient much like SDS does, it may be used as a one-to-one replacement without retraining of the underlying model, or indeed substantial code modification. While the basic algorithm works as the theory predicts in synthetic scenarios, with real-world models we have to contend with integrator error due to large score magnitudes. We have designed heuristics to alleviate this and achieve improvements on SDS in practice, but we hope future work will be able to improve the integration and/or sampling procedure, obviating the need for heuristics, in-addition to using adaptive bandwidth annealing strategies.

Acknowledgments

The authors would like to thank Pradyumn Goyal, Dmitry Petrov, Ashish Singh, and Artem Lukoianov for their helpful discussions and insights. This project has received funding from the European Research Council (ERC) under the Horizon research and innovation programme (Grant agreement No. 101124742).

References

[1] openai/clip-vit-large-patch14. https://torchmetrics.readthedocs.io/en/stable/multimodal/clip_score.html#id3. 7

[2] Thiemo Alldieck, Nikos Kolotouros, and Cristian Sminchisescu. Score distillation sampling with learned manifold corrective, 2024. 2

[3] Sherwin Bahmani, Ivan Skorokhodov, Victor Rong, Gordon Wetzstein, Leonidas Guibas, Peter Wonka, Sergey Tulyakov, Jeong Joon Park, Andrea Tagliasacchi, and David B. Lindell. 4d-fy: Text-to-4d generation using hybrid score distillation sampling. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[4] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003. 2

[5] Arwen Bradley and Preetum Nakkiran. Classifier-free guidance is a predictor-corrector, 2024. 2, 4

[6] Defang Chen, Zhenyu Zhou, Jian-Ping Mei, Chunhua Shen, Chun Chen, and Can Wang. A geometric perspective on diffusion models, 2024. 2

[7] Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1995. 1

[8] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):603–619, 2002. 1, 3, 5

[9] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021. 1, 3

[10] Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, and David Duvenaud. Scalable reversible generative models with free-form continuous dynamics. In *International Conference on Learning Representations*, 2019. 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. 7

[12] Amir Hertz, Kfir Aberman, and Daniel Cohen-Or. Delta denoising score. 2023. 2, 7

[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. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 6629–6640, Red Hook, NY, USA, 2017. Curran Associates Inc. 7

[14] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 2, 4, 6

[15] Ajay Jain, Amber Xie, and Pieter Abbeel. Vectorfusion: Text-to-svg by abstracting pixel-based diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1911–1920, 2023. 2

[16] Jihyeon Je, Jiayi Liu, Guandao Yang, Boyang Deng, Shengqu Cai, Gordon Wetzstein, Or Litany, and Leonidas Guibas. Robust symmetry detection via riemannian langevin dynamics. In *SIGGRAPH Asia 2024 Conference Papers*, New York, NY, USA, 2024. Association for Computing Machinery. 2, 5

[17] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Proc. NeurIPS*, 2022. 2, 5

[18] Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 2, 3, 4, 6

[19] Oren Katzir, Or Patashnik, Daniel Cohen-Or, and Dani Lischinski. Noise-free score distillation. In *The Twelfth International Conference on Learning Representations*, 2024. 2

[20] Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 5

[21] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *CoRR*, abs/1904.06991, 2019. 7

[22] 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. *arXiv preprint arXiv:2311.11284*, 2023. 2

[23] Artem Lukoianov, Haitz Sáez de Ocáriz Borde, Kristjan Greenewald, Vitor Campagnolo Guizilini, Timur Bagautdinov, Vincent Sitzmann, and Justin Solomon. Score distillation via reparametrized ddim, 2024. 2, 5

[24] Weijian Luo, Tianyang Hu, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhihua Zhang. Diff-instruct: A universal approach for transferring knowledge from pre-trained diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 2

[25] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, 2022. 7

[26] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv*, 2022. 1, 2, 7

[27] Sam T. Roweis and Lawrence K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 2000. 2

[28] Noam Rozen, Aditya Grover, Maximilian Nickel, and Yaron Lipman. Moser flow: Divergence-based generative modeling on manifolds, 2021. 5

[29] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 2

[30] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. 2, 3

[31] Vikas Thamizharasan, Difan Liu, Matthew Fisher, Nanxuan Zhao, Evangelos Kalogerakis, and Michal Lukac. Nivel: Neural implicit vector layers for text-to-vector generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4589–4597, 2024. 2

[32] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and the scikit-image contributors. scikit-image: image processing in Python. *PeerJ*, 2: e453, 2014. 5

[33] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A. Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. *arXiv preprint arXiv:2212.00774*, 2022. 1, 2, 4

[34] Peihao Wang, Zhiwen Fan, Dejia Xu, Dilin Wang, Sreyas Mohan, Forrest Iandola, Rakesh Ranjan, Yilei Li, Qiang Liu, Zhangyang Wang, et al. Steindreamer: Variance reduction for text-to-3d score distillation via stein identity. *arXiv preprint arXiv:2401.00604*, 2023. 2

[35] Peihao Wang, Dejia Xu, Zhiwen Fan, Dilin Wang, Sreyas Mohan, Forrest Iandola, Rakesh Ranjan, Yilei Li, Qiang Liu, Zhangyang Wang, and Vikas Chandra. Taming mode collapse in score distillation for text-to-3d generation. *arXiv preprint: 2401.00909*, 2024. 2

[36] 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 *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 2, 5

[37] Sirui Xie, Zhisheng Xiao, Diederik P Kingma, Tingbo Hou, Ying Nian Wu, Kevin Patrick Murphy, Tim Salimans, Ben Poole, and Ruiqi Gao. EM distillation for one-step diffusion models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 2

[38] Yanbo Xu, Jayanth Srinivasa, Gaowen Liu, and Shubham Tulsiani. Diverse score distillation, 2024. 2

[39] Runjie Yan, Yinbo Chen, and Xiaolong Wang. Consistent flow distillation for text-to-3d generation, 2025. 2

[40] Taoran Yi, Jiemin Fang, Junjie Wang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. Gaussiandreamer: Fast generation from text to 3d gaussians by bridging 2d and 3d diffusion models. In *CVPR*, 2024. 2

[41] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *CVPR*, 2024. 2

[42] Xin Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Song-Hai Zhang, and XIAOJUAN QI. Text-to-3d with classifier score distillation. In *The Twelfth International Conference on Learning Representations*, 2024. 2