
DIFFRATIO: Training One-Step Diffusion Models Without Teacher Supervision

Wenlin Chen ^{*1} Mingtian Zhang ^{*2} Jiajun He ³ Zijing Ou ⁴
José Miguel Hernández-Lobato ³ Bernhard Schölkopf ⁵ David Barber ²



Figure 1. Images generated by **DIFFRATIO** on ImageNet 512×512 with a single step (FID=1.41).

Abstract

Score-based distillation methods (e.g., variational score distillation) train one-step diffusion models by first pre-training a teacher score model and then distilling it into a one-step student model. However, the gradient estimator in the distillation stage usually suffers from two sources of bias: (1) biased teacher supervision due to score estimation error incurred during pre-training, and (2) the student model’s score estimation error during distillation. These biases can degrade the quality of the resulting one-step diffusion model. To address this, we propose **DIFFRATIO**, a new framework for training one-step diffusion models: instead of estimating the teacher and student scores independently and then taking their difference, we directly estimate the score difference as the gradient of a learned log density ratio between the student and data distributions across diffusion time steps. This approach greatly simplifies the training pipeline, significantly reduces gradient estimation bias, and improves one-step generation quality. Additionally, it also reduces auxiliary network size by using a lightweight

density-ratio network instead of two full score networks, which improves computational and memory efficiency. DiffRatio achieves competitive one-step generation results on CIFAR-10 and ImageNet (64×64 and 512×512), outperforming most teacher-supervised distillation approaches.

1. Introduction

Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song & Ermon, 2019) have achieved remarkable success in modeling complex real-world data across a wide range of domains, including image synthesis (Rombach et al., 2022; Li et al., 2022), 3D generation (Poole et al., 2022), video synthesis (Ho et al., 2022), and audio generation (Liu et al., 2023). Typically, diffusion models consist of two processes: a forward noising process, which gradually perturbs data into a known noise prior (typically Gaussian), and a reverse denoising process, which learns to invert the forward corruption process to generate realistic data samples from noise. Formally, the forward process is defined over T time steps as a Markov chain of Gaussian transitions, while the reverse process is parameterized by neural networks that predict the denoising distribution given the noisy samples.

In classic diffusion models with Gaussian denoising distributions, generating high-quality data samples typically requires hundreds or thousands of sampling steps, resulting in significant sampling inefficiency due to the need for $T \gg 1$ NFEs (number of function evaluations) (Ho et al., 2020; Nichol & Dhariwal, 2021). To address this weak-

^{*}Equal contribution ¹Bosch (China) Investment Ltd. ²UCL
³University of Cambridge ⁴Imperial College London ⁵Max Planck Institute for Intelligent Systems, Tübingen. Correspondence to: Wenlin Chen <wenlin.chen@cn.bosch.com>, Mingtian Zhang <m.zhang@cs.ucl.ac.uk>.

ness, various acceleration methods have been proposed to reduce NFEs during sampling. One class of such approaches leverages advanced numerical solvers for differential equations, enabling continuous-time approximations of the diffusion process (Song et al., 2020; Liu et al., 2022; Lu et al., 2022). Another line of work improves the flexibility of the posterior distribution in the denoising process, either by estimating a more accurate covariance for the Gaussian distribution (Nichol & Dhariwal, 2021; Bao et al., 2022b;a; Ou et al., 2025) or by adopting flexible non-Gaussian denoising distributions (Bortoli et al., 2025; Xiao et al., 2021; Yu et al., 2024). While these techniques can dramatically reduce NFEs from $\sim 10^3$ to around 10–20, they still fall short of achieving high-quality generation within 5 steps.

Recently, distillation-based methods have emerged as a powerful direction for training diffusion models, enabling high-quality *one-step* generation (Zhou et al., 2024). These methods fall into two categories. Trajectory-based distillation methods (Salimans & Ho, 2022; Berthelot et al., 2023; Song et al., 2023; Heek et al., 2024; Kim et al., 2024; Frans et al., 2024; Li & He, 2024; Geng et al., 2025a; Boffi et al., 2025) aim to approximate the full sampling trajectory by training a student model to amortize multiple intermediate steps. These methods are motivated by accelerated solvers and typically perform joint training of the full diffusion model and the distillation process. Score-based distillation approaches (Luo et al., 2024; Yin et al., 2024; Salimans et al., 2024; Xie et al., 2024; Zhou et al., 2024) distill the full denoising process of the pre-trained teacher diffusion model into a one-step latent variable model. This distillation process typically involves minimizing the divergence between the student and teacher models based on their respective score estimations (Poole et al., 2022; Wang et al., 2024).

Score-based distillation methods typically follow a two-stage pipeline: (1) *pre-training stage*: training a teacher score model via denoising score matching on the data distribution, and (2) *distillation stage*: using the pre-trained teacher score to supervise the training of a one-step student model. However, the distillation-stage gradient estimator suffers from two sources of bias: (1) imperfect teacher score supervision due to score estimation error from pre-training, and (2) student score estimation error during distillation. These biases can accumulate and thus significantly degrade the quality of the resulting one-step model.

This paper proposes **DIFFRATIO**, a new framework for training one-step diffusion models by directly estimating the *score difference* between the student and data distributions via the gradient of a learned log density ratio. This approach eliminates the need for separate student score estimation and reduces gradient estimation bias. Moreover, it uses a lightweight time-conditioned density-ratio network (i.e., a density-ratio estimator) instead of two full score networks,

which improves computational and memory efficiency and achieves SOTA one-step image generation performance.

2. Background on Diffusion Models

Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song & Ermon, 2019) define a generative framework that maps a Gaussian prior $p(x_T)$ to the data distribution $p(x_0)$ through a forward noising process and a learned reverse denoising process. The forward process progressively adds Gaussian noise to the data

$$q(x_{0:T}) = p_d(x_0) \prod_{t=1}^T q(x_t|x_{t-1}), \quad (1)$$

with transition kernels defined as

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t|\sqrt{1-\beta_t}x_{t-1}, \beta_t I), \quad (2)$$

where $\beta_t \in (0, 1)$ is a pre-specified variance. The skip conditional distribution at time t can be written as:

$$q(x_t|x_0) = \mathcal{N}(x_t|\sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I), \quad (3)$$

where $\bar{\alpha}_t = \prod_{s=1}^t (1-\beta_s)$. As $T \rightarrow \infty$, the final state x_T approximates a standard normal distribution, i.e., $q(x_T) \rightarrow \mathcal{N}(0, I)$. The generative process aims to reverse this trajectory. Starting from noise $x_T \sim p(x_T) = \mathcal{N}(0, I)$, the model learns a reverse process to sequentially denoise and reconstruct data samples. Since the true reverse conditional $q(x_{t-1}|x_t)$ is intractable, a common method is to approximate it with a variational Gaussian distribution:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}|\mu_{t-1}(x_t; \theta), \Sigma_{t-1}(x_t; \theta)), \quad (4)$$

where the mean function is learned from data and the covariance function can be either learned (Nichol & Dhariwal, 2021; Ou et al., 2025; Bao et al., 2022a) or chosen to be a fixed value (Bao et al., 2022b; Ho et al., 2020).

From a score-based perspective, the denoising mean $\mu_{t-1}(x_t; \theta)$ can be written in terms of the (time-dependent) score of the forward marginal $q(x_t)$, i.e., $\nabla_{x_t} \log q(x_t)$. In practice, we learn score networks using denoising score matching (DSM) (Vincent, 2011; Song & Ermon, 2019). Specifically, for any distribution $q(x_0)$ and its noised marginal $q(x_t) = \int q(x_t|x_0) q(x_0) dx_0$, DSM trains a score network $s_\theta(x_t, t)$ to estimate $\nabla_{x_t} \log q(x_t)$ by minimizing

$$\mathbb{E}_{q(x_0) q(x_t|x_0)} \left[\|s_\theta(x_t, t) - \nabla_{x_t} \log q(x_t|x_0)\|_2^2 \right]. \quad (5)$$

Given an estimated score $s_\theta(x_t, t) \approx \nabla_{x_t} \log q(x_t)$, Tweedie's Lemma (Efron, 2011; Robbins, 1992) yields the corresponding reverse mean:

$$\mu_{t-1}(x_t; \theta) = \frac{1}{\sqrt{1-\beta_t}} (x_t + \beta_t s_\theta(x_t, t)), \quad (6)$$

Algorithm 1 Score-based distillation for training one-step diffusion models with teacher supervision (VSD)

Require: Training samples $\{x^{(1)}, \dots, x^{(N)}\} \sim p_d(x_0)$

// Stage 1: Pre-train a teacher score model (requires training data)

- 1: Pre-train the teacher model's score network $s_{\psi_1}^{p_d}(x_t, t)$ using DSM (Eq. 12) until convergence

// Stage 2: Distill a one-step student model with teacher supervision

- 2: Initialize the one-step generator with the teacher's score network $g_{\theta_{\text{init}}}(\cdot) \equiv s_{\psi_1}^{p_d}(\cdot, t = t_{\text{init}})$
- 3: **for** each training iteration **do**
- 4: Estimate the student model's score by training a score network $s_{\psi_2}^{q_\theta}(x_t, t)$ using DSM
- 5: Update the one-step generator g_θ with $s_{\psi_2}^{q_\theta}(x_t, t)$ (student score) and $s_{\psi_1}^{p_d}(x_t, t)$ (teacher supervision) using Eq. 14
- 6: **end for**

which establishes a connection between the denoising distributional perspective and the score estimation perspective of diffusion models. In the following section, we introduce score-based distillation methods through the lens of divergence minimization.

2.1. Score-Based Diffusion Distillation

Score-based distillation methods aim to distill a teacher diffusion model p_θ (pre-trained on the true data distribution $p_d(x_0)$) into a one-step implicit generative model (Goodfellow et al., 2014; Huszár, 2017; Zhang et al., 2020):

$$q_\theta(x_0) = \int \delta(x_0 - g_\theta(z))p(z)dz, \quad (7)$$

where $\delta(\cdot)$ is the Dirac delta function, $p(z)$ is a standard Gaussian prior for the latent variable z , and $g_\theta : \mathcal{Z} \rightarrow \mathcal{X}$ is a deterministic neural network that generates data x from the latent variable z in one step. We want to minimize the divergence between this one-step model and the true data distribution. However, when the function $g_\theta(\cdot)$ is not bijective, the model distribution q_θ is not absolutely continuous w.r.t. the Lebesgue measure. As a result, the corresponding density function may not be well-defined, and consequently, the commonly used f -divergence such as KL divergence between $q_\theta(x_0)$ and the data distribution $p_d(x_0)$ may also be ill-defined (Arjovsky et al., 2017; Zhang et al., 2020).

Inspired by diffusion models, one can use a set of (scaled) Gaussian convolution kernels $\mathcal{K} = \{k_1, \dots, k_T\}$ with $k_t(x_t|x_0) = \mathcal{N}(x_t|\alpha_t x_0, \sigma_t^2 I)$ to define the Diffusive KL divergence (DiKL) between the model density $q_\theta(x_0)$ and target distribution $p_d(x_0)$:

$$\text{DiKL}_{\mathcal{K}}(q_\theta||p_d) \equiv \sum_{t=1}^T w(t) \text{KL}(q_\theta(x_t)||p_d(x_t)), \quad (8)$$

where $w(t)$ is a positive scalar weighting function that sums to one, and $q_\theta(x_t)$ and $p_d(x_t)$ are noisy model density and

noisy target density, respectively, defined by

$$q_\theta(x_t) = \int k(x_t|x_0)q_\theta(x_0)dx_0, \quad (9)$$

$$p_d(x_t) = \int k(x_t|x_0)p_d(x_0)dx_0. \quad (10)$$

In this case, the distributions $q_\theta(x_t)$ and $p_d(x_t)$ are always absolutely continuous, and thus the KL divergence between them is always well-defined. For a single Gaussian kernel, the divergence was known as *Spread KL divergence* (Zhang et al., 2020; 2019). It is straightforward to show that it is a valid divergence, i.e., $\text{DiKL}_{\mathcal{K}}(q_\theta||p_d) = 0 \Leftrightarrow q_\theta = p_d$; see Zhang et al. (2020) for a proof. In addition to the diffusion distillation (Luo et al., 2024; Xie et al., 2024; Wang et al., 2025), this divergence has successfully been used in 3D generative models (Poole et al., 2022; Wang et al., 2024) and training neural samplers (He et al., 2024).

For a single kernel k_t , the DiKL term admits the following pathwise gradient (see Appendix A for a derivation):

$$\begin{aligned} & \nabla_\theta \text{DiKL}_t(q_\theta(x_0)||p_d(x_0)) \\ &= \int q_\theta(x_t) \left(\underbrace{\nabla_{x_t} \log q_\theta(x_t) - \nabla_{x_t} \log p_d(x_t)}_{\text{score difference at time } t} \right) \frac{\partial x_t}{\partial \theta} dx_t. \end{aligned} \quad (11)$$

However, neither the noisy model score $\nabla_{x_t} \log q_\theta(x_t)$ nor the noisy target score $\nabla_{x_t} \log p_d(x_t)$ are directly accessible. In the distillation setting, the noisy target score is provided by a pre-trained diffusion model. Specifically, a score network $s_{\psi_1}^{p_d}(x_t, t)$ is learned to approximate $\nabla_{x_t} \log p_d(x_t)$ by minimizing the DSM loss w.r.t. ψ_1 at time t :

$$\mathbb{E}_{p_d(x_0)k(x_t|x_0)} \left[\|s_{\psi_1}^{p_d}(x_t, t) - \nabla_{x_t} \log k(x_t|x_0)\|_2^2 \right]. \quad (12)$$

Regarding the noisy model score $\nabla_{x_t} \log q_\theta(x_t)$, since samples from q_θ can be efficiently obtained, we can approximate the score with another score network $s_{\psi_2}^{q_\theta}(x_t, t) \approx \nabla_{x_t} \log q_\theta(x_t)$ by minimizing the DSM loss w.r.t. ψ_2 :

$$\mathbb{E}_{q_\theta(x_0)k(x_t|x_0)} \left[\|s_{\psi_2}^{q_\theta}(x_t, t) - \nabla_{x_t} \log k(x_t|x_0)\|_2^2 \right]. \quad (13)$$

Thus, the gradient of DiKL w.r.t. the parameters θ of the student model can be estimated as follows, a method known as Variational Score Distillation (VSD) (Poole et al., 2022; Wang et al., 2024; Luo et al., 2024):

$$\begin{aligned} & \nabla_{\theta} \text{DiKL}(q_{\theta}(x_0) \| p_d(x_0)) \\ & \approx \sum_{t=1}^T w(t) \int q_{\theta}(x_t) (s_{\psi_2}^{q_{\theta}}(x_t, t) - s_{\psi_1}^{p_d}(x_t, t)) \frac{\partial x_t}{\partial \theta} dx_t. \end{aligned} \quad (14)$$

Unlike the teacher score function which remains fixed after pre-training, the noisy model score $\nabla_{x_t} \log q_{\theta}(x_t)$ dynamically changes as we update the student model's parameters θ during training. Therefore, the score network $s_{\psi_2}^{q_{\theta}}(x_t, t) \approx \nabla_{x_t} \log q_{\theta}(x_t)$ needs to be updated every time we update the student model, which results in an interleaved training procedure as detailed in Algorithm 1.

Limitations of the VSD. Despite the initial success of VSD in diffusion distillation, this two-stage pipeline has inherent limitations. In the first stage, the teacher score model $s_{\psi_1}^{p_d}(x_t, t)$ is imperfect due to limited network capacity and optimization. In the second stage, the distillation process solely relies on the pre-trained teacher score for supervision. Therefore, the gradient estimation in VSD (Eq. 14) has biases from *two sources*: (1) the teacher's score estimation error and (2) the student's score estimation error. These biases can accumulate and thus adversely affect the quality of the distilled student model.

In the next section, we propose a new method that directly estimates the score difference $\nabla_{x_t} \log q_{\theta}(x_t) - \nabla_{x_t} \log p_d(x_t)$ with a single lightweight network, which reduces the gradient estimation bias and improves generation quality.

3. DiffRatio: Training One-Step Diffusion Models Without Teacher Supervision

From Algorithm 1, we observe that the DiKL gradient depends on the difference between two score functions, which can be expressed as the gradient of a log density ratio:

$$\nabla_{x_t} \log q_{\theta}(x_t) - \nabla_{x_t} \log p_d(x_t) = \nabla_{x_t} \log \frac{q_{\theta}(x_t)}{p_d(x_t)}. \quad (15)$$

Therefore, instead of estimating the two score functions separately, we can directly learn the density ratio $\frac{q_{\theta}(x_t)}{p_d(x_t)}$ and then take the gradient of its logarithm to estimate the score difference. We thus refer to this approach as **DiffRatio**, which we introduce below.

3.1. Diffusive Density Ratio Gradient Estimation

We use class ratio estimation (Sugiyama et al., 2012; Qin, 1998; Gutmann & Hyvärinen, 2010; Zhang et al., 2022) to estimate the density ratio across all noise levels. We first

denote distributions $q_{\theta}(x_t)$ and $p_d(x_t)$ as two conditional distributions $m(x_t|y=0)$ and $m(x_t|y=1)$, respectively, where $y=0$ indicates samples from the one-step model $q_{\theta}(x_t)$ and $y=1$ indicates data samples from $p_d(x_t)$. With Bayes' rule, we can transform the density ratio estimation problem into a binary classification problem:

$$\begin{aligned} \frac{q_{\theta}(x_t)}{p_d(x_t)} & \equiv \frac{m(x_t|y=0)}{m(x_t|y=1)} = \frac{p(y=0|x_t)m(x_t)/p(y=0)}{p(y=1|x_t)m(x_t)/p(y=1)} \\ & = \frac{p(y=0|x_t)}{p(y=1|x_t)}, \end{aligned} \quad (16)$$

where the mixture distribution is defined as $m(x) \equiv m(x_t|y=1)p(y=1) + m(x_t|y=0)p(y=0)$, and the Bernoulli prior distribution $p(y)$ is simply set as a uniform prior $p(y=1) = p(y=0) = 0.5$. In practice, this is achieved by drawing a batch of data from $p_d(x_t)$ with the label $y=1$ and drawing an equal number of samples from $q_{\theta}(x_t)$ with the label $y=0$. We then train a neural network classifier $c_{\eta}(x_t, t)$, conditioned on the diffusion time t , to estimate the probability that a given input x_t belongs to class $y=1$. The optimal classifier approximates the posterior probability $c_{*}(x_t, t) = p(y=1|x_t)$. In this case, the log-density ratio can be written as

$$\begin{aligned} \nabla_{x_t} \log \frac{q_{\theta}(x_t)}{p_d(x_t)} & = \nabla_{x_t} \log \frac{1 - c_{*}(x_t, t)}{c_{*}(x_t, t)} \\ & = \nabla_{x_t} \text{logit}(1 - c_{*}(x_t, t)). \end{aligned} \quad (17)$$

We can then obtain a new gradient estimator for DiKL by plugging in our density-ratio estimator to Eq. 11:

$$\begin{aligned} & \nabla_{\theta} \text{DiKL}(q_{\theta}(x_0) \| p_d(x_0)) \\ & \approx \sum_{t=1}^T w(t) \int q_{\theta}(x_t) \nabla_{x_t} \text{logit}(1 - c_{\eta}(x_t, t)) \frac{\partial x_t}{\partial \theta} dx_t. \end{aligned} \quad (18)$$

Importantly, our method directly estimates the score difference $\nabla_{x_t} \log q_{\theta}(x_t) - \nabla_{x_t} \log p_d(x_t)$ using a single density-ratio estimator $c_{\eta}(x_t, t)$, which alleviates the two sources of bias in VSD (Eq. 14): (1) Instead of separately estimating the student score $s_{\psi_2}^{q_{\theta}}(x_t, t)$ and the teacher score $s_{\psi_1}^{p_d}(x_t, t)$, the density-ratio estimator directly estimates the score difference in an integrated manner, avoiding the accumulation of errors from two independently trained networks. (2) Without relying on the pre-trained teacher score for supervision, we avoid bias propagation from the teacher to the student, leading to a *consistent* estimator—given sufficient capacity and training, the estimated score difference converges to the true score difference. In contrast, *VSD cannot guarantee such consistency*, as the bias from teacher supervision cannot be removed regardless of how well the student score network is trained. In the next section, we give an empirical comparison of the gradient estimation bias between VSD and DiffRatio methods.

Table 1. Evaluation of trained generators on the 2D toy problem. Log-MMD (↓): lower is better. Log-density (↑): higher is better.

METHOD	Log-MMD (↓)	Log-density (↑)
True	/	-6.65
VSD	-5.25	-10.88
DiffRatio	-6.08	-6.79

3.2. Empirical Analysis of the Gradient Bias

To empirically validate that our density-ratio-based gradient estimator reduces bias compared to VSD, we design a controlled 2D experiment where ground-truth score differences can be computed. The data distribution $p_d(x_0)$ is a 2D mixture of Gaussians with 40 components, which allows closed-form computation of the true data score $\nabla_{x_t} \log p_d(x_t)$ at any noise level. The one-step model $q_\theta(x_0)$ is an implicit generative model that has been pre-trained to approximate p_d . Since $q_\theta(x_t)$ lacks a closed-form density, we estimate its score using kernel density estimation (KDE) with 10,000 samples, which provides highly accurate score estimates in this low-dimensional setting. This gives us the ground-truth score difference: $\Delta s_t^* \equiv \Delta s^*(x_t, t) = \nabla_{x_t} \log q_\theta(x_t) - \nabla_{x_t} \log p_d(x_t)$.

We compare the following two methods on this problem.

1. DiffRatio (ours), which trains a single classifier $c_\eta(x_t, t)$ and estimates the score difference as $\widehat{\Delta s}_t = \nabla_{x_t} \text{logit}(1 - c_\eta(x_t, t))$.
2. VSD, which trains two separate score networks $s_{\psi_1}^{p_d}$ and $s_{\psi_2}^{q_\theta}$ using DSM and computes their difference $\widehat{\Delta s}_t = s_{\psi_2}^{q_\theta}(x_t, t) - s_{\psi_1}^{p_d}(x_t, t)$.

We evaluate using two metrics: (1) the relative L2 error $\|\widehat{\Delta s}_t - \Delta s_t^*\|_2 / \|\Delta s_t^*\|_2$, and (2) cosine similarity, where $\widehat{\Delta s}_t$ denotes the estimated score difference from each method. We train both methods with 5,000 steps; other details can be found in Appendix D. As shown in Figure 2, our density-ratio-based method consistently achieves lower L2 error and higher cosine similarity than VSD across all noise levels, indicating more accurate gradient estimation. This is because VSD accumulates errors from both the student score network $s_{\psi_2}^{q_\theta}$ and the teacher score network $s_{\psi_1}^{p_d}$. In contrast, our method directly estimates the score *difference* using a single density-ratio estimator c_η , avoiding error accumulation from separately trained networks. This empirically validates that density-ratio estimation provides more accurate gradient signals for training one-step models.

To further illustrate the effect of gradient bias to generation quality, we compare the two gradient estimation methods by training one-step models under the same divergence (DiKL)

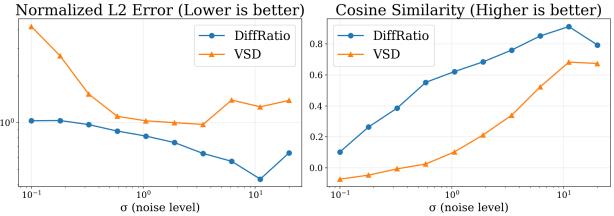


Figure 2. Score difference estimation accuracy on a 2D mixture of Gaussians problem. Our density-ratio-based method achieves lower L2 error (left) and higher cosine similarity (right) with the ground-truth score difference than VSD, which suffers from accumulated errors in separate teacher and student score estimation.

on this problem. We report log-MMD (maximum mean discrepancy) and log-density under the true data distribution in Table 1. Our method achieves significantly better log-density than VSD, indicating that the generated samples are closer to the true data distribution. The improved gradient estimation translates directly to better generation quality.

3.3. Training Criterion Extensions

In addition to the DiKL, we can use the learned classifier function c_η to obtain a family of gradient estimators for alternative training objectives. For instance, replacing the logit function with the logarithm function yields an objective that minimizes the probability of generated samples being classified as fake. This formulation aligns with GAN (Goodfellow et al., 2014; Nowozin et al., 2016) across different diffusion time steps, which approximately minimizes the *Diffusive Jensen-Shannon* (DiJS) divergence (a detailed derivation of DiJS and its gradient estimator can be found in Appendix B):

$$\begin{aligned} & \nabla_\theta \text{DiJS}(q_\theta(x_0) || p_d(x_0)) \quad (19) \\ & \approx \sum_{t=1}^T w(t) \int q_\theta(x_t) \nabla_{x_t} \log(1 - c_\eta(x_t, t)) \frac{\partial x_t}{\partial \theta} dx_t. \end{aligned}$$

Alternatively, rather than minimizing the probability of the generated images being fake as used in GAN, one can also maximize the probability of them being real. This approach is referred to as *Diffusive Realism Maximization* (DiRM), which has the following gradient estimator:

$$\begin{aligned} & \nabla_\theta \text{DiRM}(\theta) \quad (20) \\ & \approx - \sum_{t=1}^T w(t) \int q_\theta(x_t) \nabla_{x_t} \log c_\eta(x_t, t) \frac{\partial x_t}{\partial \theta} dx_t. \end{aligned}$$

Notably, the DiRM objective—maximizing the likelihood of being real—also mirrors the *non-saturating GAN* formulation (Goodfellow et al., 2014), which is known to provide more stable gradients for the generator compared to the original minimax objective.

In principle, once the density ratio is available, any ratio-based divergence (e.g., an f -divergence) can be used for

Algorithm 2 Training one-step diffusion models via diffusive density ratio estimation (DiffRatio, ours)

Require: Training samples $\{x^{(1)}, \dots, x^{(N)}\} \sim p_d(x_0)$

// Stage 1: Pre-train the score model

- 1: Pre-train the score network $s_\theta^{p_d}(x_t, t)$ using DSM (Eq. 12) until convergence

// Stage 2: Train the one-step model without teacher supervision

- 2: Fix the diffusion time argument $t = t_{\text{init}}$ in the score network to create a one-step generator $g_\theta(\cdot) \equiv s_\theta^{p_d}(\cdot, t = t_{\text{init}})$
- 3: **for** each training iteration **do**
- 4: Sample time t' for diffusive density ratio estimation
- 5: Train a classifier $c_\eta(x_{t'}, t')$ on samples from $q_\theta(x_{t'})$ and $p_d(x_{t'})$ to estimate the density ratio $q_\theta(x_{t'})/p_d(x_{t'})$
- 6: Update the one-step generator g_θ with $c_\eta(x_{t'}, t')$ using Eq. 18 or Eq. 19 or Eq. 20
- 7: **end for**

training. This flexibility is not available when one only has access to the teacher and student scores (as in VSD), making DiffRatio applicable to a broader family of criteria. Moreover, we provide a convergence discussion in Appendix C, where we justify the consistency of our DiffRatio gradient estimators under *weaker assumptions* than those required for standard ratio estimation (Hyvärinen, 2005).

3.4. Simplified Training with Improved Efficiency

The overall training procedure of our proposed DiffRatio framework is summarized in Algorithm 2. In our method, the generator is first pre-trained as a score network using DSM, and then further trained as a one-step model with the diffusion time argument in the score network fixed to $t=t_{\text{init}}$. This design leads to a more efficient training pipeline compared to VSD. Specifically, VSD requires maintaining and evaluating three networks during training: (1) the pre-trained teacher score network $s_{\psi_1}^{p_d}$, (2) the student score network $s_{\psi_2}^{q_\theta}$, and (3) the generator g_θ . In contrast, our method only requires two: (1) the generator g_θ , and (2) the density-ratio estimator c_η , which is approximately half the size of a full score network. This results in significant computational and memory savings during training.

For low-dimensional problems such as the 2D mixture of Gaussians in Section 3.2, the generator can be trained from random initialization. However, for high-dimensional image generation, we find that pre-training the generator with DSM is necessary to prevent mode collapse, which is consistent with findings in the score-based distillation literature (Luo et al., 2024; Zhou et al., 2025b; 2024). To further understand the necessity of this pre-training, we provide a detailed analysis in Appendix F, which studies the role of weight initialization from two perspectives: (1) the functional mapping perspective, and (2) the feature space perspective. Our analysis shows that multi-level features learned during diffusion pre-training are essential for preventing mode collapse in high-dimensional settings.

4. Related Work

4.1. Comparison with GANs

Estimating density ratios is central to many GAN variants (Goodfellow et al., 2014; Nowozin et al., 2016). In classic GANs, the discriminator implicitly estimates the density ratio between real and model distributions. However, for high-dimensional image modeling tasks, both data and model distributions are supported on low-dimensional manifolds and are therefore not absolutely continuous, rendering their densities and the density ratio ill-defined. This further causes the Jensen-Shannon (JS) divergence to be ill-defined and contributes to GAN training instability (Arjovsky & Bottou, 2017; Arjovsky et al., 2017; Mescheder et al., 2018; Roth et al., 2017).

Alternative Divergence. To address this, Arjovsky et al. (2017) proposed to replace the JS divergence with the Wasserstein-1 distance, which can yield meaningful gradients even when the distributions are disjoint. However, training Wasserstein GANs requires enforcing a 1-Lipschitz constraint on the critic, which is challenging in practice and has been approximated using heuristics such as weight clipping (Arjovsky et al., 2017), gradient penalties (Gulrajani et al., 2017), and spectral normalization (Miyato et al., 2018). Despite its theoretical appeal, the divergence minimized in practice often differs from the idealized objective (Mescheder et al., 2018), and that stable GAN training relies more on regularization (e.g., gradient penalties) than on strict divergence minimization (Fedus et al., 2017).

Noise as Regularization. Adding Gaussian noise to real and fake samples has been proposed as a way to ensure distributions are fully supported, making the density ratio well-defined (Sønderby et al., 2016; Roth et al., 2017; Nowozin et al., 2016; Zhang et al., 2020). This model-agnostic approach requires no architectural changes but hinges on choosing an effective noise level—something hard to fix throughout training.



(a) CIFAR-10 32×32 (FID=2.39, IS=9.93)

(b) ImageNet 64×64 (FID=1.54)

Figure 3. Images generated by DiffRatio-DiJS on CIFAR-10 and ImageNet 64×64. Each image is generated with a single NFE.

4.2. Comparison with Diffusion GAN

Diffusion GAN (Wang et al., 2023) addresses the challenge of selecting a fixed noise level by introducing a diffusion-inspired noise schedule that gradually increases noise in tandem with the model’s learning capacity. While this represents the most closely related work to ours, our approach differs in several important aspects, as outlined below.

Agnostic to Generator Architecture. Diffusion GAN stabilizes training using a StyleGAN-based generator (Karras et al., 2024), which is implicitly trained progressively from low to high resolutions while keeping the network topology fixed. In contrast, our method does not rely on a specialized generator architecture or progressive resolution training. Instead, we adopt a generic U-Net architecture, resulting in a simpler and more broadly applicable framework.

No Additional Regularization Techniques. We do not use common GAN-specific training tricks such as gradient penalties or spectral normalization. Our method provides a clean framework for training one-step diffusion models yet still achieves stable convergence, demonstrating robustness without any additional regularization tricks during training.

Avoiding Adversarial Nature. Unlike traditional GANs, our framework is no longer adversarial in the strict sense: our generator’s performance is *not* dependent on the convergence of a discriminator, which simplifies the training process and mitigates common issues arising from adversarial dynamics (e.g., GANs require a careful balancing between discriminator and generator training).

As our method does not rely on network architectures, gradient penalties, or adversarial training, we are estimating and minimizing the actual DiJS divergence—a divergence that remains well-defined. This leads to a conceptually cleaner training objective that is more amenable to theoretical analysis and empirical diagnosis, as it avoids the complexities

Table 2. Unconditional one-step generation on CIFAR-10.

METHOD	NFE (↓)	FID (↓)	IS (↑)
Teacher model			
EDM (Karras et al., 2022)	35	2.04	9.84
EDM (Karras et al., 2022)	1	8.70	8.49
Training from scratch without teacher			
CT (Song et al., 2023)	1	8.70	8.49
iCT (Song & Dhariwal, 2024)	1	2.83	9.54
iCT-deep (Song & Dhariwal, 2024)	1	2.51	9.76
BCM (Li & He, 2024)	1	3.10	9.45
BCM-deep (Li & He, 2024)	1	2.64	9.67
sCT (Lu & Song, 2025)	1	2.85	-
Diffusion-GAN (Wang et al., 2023)	1	3.19	-
IMM (Zhou et al., 2025a)	1	3.20	-
MeanFlow (Geng et al., 2025a)	1	2.92	-
Distillation with teacher supervision			
Progressive Distillation (Salimans & Ho, 2022)	1	8.34	8.69
DSNO (Zheng et al., 2023)	1	3.78	-
TRACT (Berthelot et al., 2023)	1	3.78	-
CD (Song et al., 2023)	1	3.55	9.48
CTM (w/ GAN) (Kim et al., 2024)	1	1.98	-
CTM (w/o GAN) (Kim et al., 2024)	1	>5.0	-
sCD (Lu & Song, 2025)	1	3.66	-
Diff-Instruct (Luo et al., 2024)	1	4.53	-
ECT (Geng et al., 2025b)	1	3.60	-
SiD (Zhou et al., 2024)	1	2.03	10.02
SiDA (Zhou et al., 2025b)	1	1.52	10.32
Training without teacher supervision			
DiffRatio-DiRM (ours)	1	4.87	9.85
DiffRatio-DiKL (ours)	1	3.81	9.90
DiffRatio-DiJS (ours)	1	2.39	9.93

and instabilities associated with adversarial min-max optimization and additional regularization tricks.

5. Image Generation Experiments

We evaluate the performance of our proposed method by training one-step image diffusion models on two standard benchmarks: (1) unconditional generation on the CIFAR-10 (32×32) dataset (Krizhevsky, 2009), (2) class-label-conditioned generation on the ImageNet (64×64 and 512×512) dataset (Deng et al., 2009).



Figure 4. Images generated by DiffRatio-DiJS-M on ImageNet 512×512 (FID=1.41). Each image is generated with 1 step (1 NFE).

Table 3. Conditional one-step generation on ImageNet 64×64.

METHOD	NFE (↓)	FID (↓)
Teacher model		
EDM (Karras et al., 2022)	79	2.44
Training from scratch without teacher		
EDM2-L/XL (Karras et al., 2024)	1	13.0
CT (Song et al., 2023)	1	13.0
iCT (Song & Dhariwal, 2024)	1	4.02
iCT-deep (Song & Dhariwal, 2024)	1	3.25
BCM (Li & He, 2024)	1	4.18
BCM-deep (Li & He, 2024)	1	3.14
sCT (Lu & Song, 2025)	1	2.04
Distillation with teacher supervision		
Progressive Distillation (Salimans & Ho, 2022)	1	7.88
DSNO (Zheng et al., 2023)	1	7.83
TRACT (Berthelot et al., 2023)	1	7.43
CD (Song et al., 2023)	1	6.20
CTM (w/ GAN) (Kim et al., 2024)	1	1.92
sCD (Lu & Song, 2025)	1	2.44
Diff-Instruct (Luo et al., 2024)	1	5.57
EM Distillation (Xie et al., 2024)	1	2.20
ECT (Geng et al., 2025b)	1	2.49
SiD (Zhou et al., 2024)	1	2.02
SiDA (Zhou et al., 2025b)	1	1.35
DMD2 (w/ GAN) (Yin et al., 2024)	1	1.51
DMD2 (w/ GAN & longer training) (Yin et al., 2024)	1	1.28
Training without teacher supervision		
DiffRatio-DiJS (ours)	1	1.54

Our one-step diffusion model adopts the same neural network architecture as the pre-trained score model. Specifically, we employ the EDM U-Net model architecture (Karras et al., 2022) for CIFAR-10 and ImageNet 64×64, and use the EDM2-M U-Net model architecture (Karras et al., 2024) for ImageNet 512×512. We consider three training criteria in our framework (DiKL/DiJS/DiRM) on CIFAR-10, and use the best-performing criterion (DiJS) for higher-dimensional ImageNet experiments (64×64 and 512×512). We use the variance-exploding (VE) noise schedule to define the diffusive divergences with $t_{\max}=1000$. Our density

Table 4. Conditional one-step generation on ImageNet 512×512.

METHOD	CFG	NFE (↓)	FID (↓)
Teacher model			
EDM2-M (Karras et al., 2024)	N	63	2.25
EDM2-M (Karras et al., 2024)	Y	126	2.01
Training from scratch without teacher			
sCT-M (Lu & Song, 2025)	Y	1	5.84
sCD-M (Lu & Song, 2025)	Y	1	2.75
Distillation with teacher supervision			
SiD-M (Zhou et al., 2024)	N	1	2.06
SiDA-M (Zhou et al., 2025b)	N	1	1.55
Training without teacher supervision			
DiffRatio-DiJS-M (ours)	N	1	1.41

ratio estimator $c_\eta(x_t, t)$ is implemented using the encoder portion of the teacher’s U-Net architecture with a sigmoid function attached to the end to produce a scalar output between 0 and 1. This network is approximately half the size of the full U-Net used for score estimation, leading to improved computational and memory efficiency. The one-step generator $g_\theta(z)$ is initialized using the weights from the pre-trained score model with the diffusion time argument $t_{\text{init}}=2.5$ fixed throughout training and sampling. We follow standard hyperparameter settings for training generative models on CIFAR-10 and ImageNet as detailed in Karras et al. (2022; 2024). We use a batch size of 64 for CIFAR-10 and 4,096 for ImageNet. The weighting function $w(t)=\sigma_t^2$ is used for both datasets. We adopt a single-step update strategy for the density-ratio estimator throughout our experiments, consistent with previous works (Luo et al., 2024; Zhou et al., 2024). See Appendix E for more details.

In Tables 2, 3 and 4, we compare our method to previous methods for training one-step generative models. To highlight methodological differences, we categorize these approaches into three groups: (1) training from scratch,

training with teacher supervision (e.g., using teacher score, denoiser or ODE trajectory as supervision in the distillation loss), and (3) training without teacher supervision. We find that our proposed method, DiffRatio, achieves competitive one-step generation performance despite not using any teacher supervision, outperforming most state-of-the-art distillation methods that rely on full teacher supervision. The only baseline method with comparable performance to ours is SiDA (Zhou et al., 2025b), which depends on training data, teacher score supervision, teacher weight initialization, and student score estimation. In contrast, our method requires only training data, teacher weight initialization, and density-ratio estimation. Notably, this *eliminates* the need for teacher supervision in the training process. Also, density-ratio estimation is both simpler and *more lightweight* than student score estimation, as the density-ratio network is approximately half the size of a full score network. Moreover, unlike VSD, our approach does not require maintaining or evaluating the teacher score network during the training process. This results in a more streamlined and *efficient* training framework, since our approach does not require maintaining and evaluating the teacher and student score networks in the training procedure. Samples generated by DiffRatio-DiJS are visualized in Figures 3, 4, 6, 7 and 8.

6. Conclusions

In this paper, we introduced **DIFFRATIO**, a new framework for training one-step diffusion models by directly estimating the score difference between the one-step model and the true data distribution across diffusion time steps with a lightweight density-ratio network. Our approach mitigates the gradient-estimation bias caused by separately learning teacher and student scores in score-based distillation. Our method achieved competitive generation quality on image synthesis benchmarks with improved efficiency, outperforming most approaches with full teacher supervision.

Acknowledgements

We would like to thank Prof. Mingyuan Zhou for the useful discussion. MZ and DB acknowledge funding from AI Hub in Generative Models, under grant EP/Y028805/1. JH is supported by the University of Cambridge Harding Distinguished Postgraduate Scholars Programme. ZO is supported by the Lee Family Scholarship. JMHL acknowledges support from a Turing AI Fellowship under grant EP/V023756/1.

Impact Statement

This paper advances machine learning and may have societal impacts, none of which we believe require specific discussion.

References

Arjovsky, M. and Bottou, L. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*, 2017.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In *International conference on machine learning*, pp. 214–223. PMLR, 2017.

Bao, F., Li, C., Sun, J., Zhu, J., and Zhang, B. Estimating the optimal covariance with imperfect mean in diffusion probabilistic models. *arXiv preprint arXiv:2206.07309*, 2022a.

Bao, F., Li, C., Zhu, J., and Zhang, B. Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. *arXiv preprint arXiv:2201.06503*, 2022b.

Berthelot, D., Autef, A., Lin, J., Yap, D. A., Zhai, S., Hu, S., Zheng, D., Talbott, W., and Gu, E. Tract: Denoising diffusion models with transitive closure time-distillation. *arXiv preprint arXiv:2303.04248*, 2023.

Boffi, N. M., Albergo, M. S., and Vanden-Eijnden, E. Flow map matching with stochastic interpolants: A mathematical framework for consistency models. *Transactions on Machine Learning Research*, 2025.

Bortoli, V. D., Galashov, A., Guntupalli, J. S., Zhou, G., Murphy, K., Gretton, A., and Doucet, A. Distributional diffusion models with scoring rules, 2025.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. IEEE, 2009.

Efron, B. Tweedie’s formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011.

Fedus, W., Rosca, M., Lakshminarayanan, B., Dai, A. M., Mohamed, S., and Goodfellow, I. Many paths to equilibrium: Gans do not need to decrease a divergence at every step. *arXiv preprint arXiv:1710.08446*, 2017.

Frans, K., Hafner, D., Levine, S., and Abbeel, P. One step diffusion via shortcut models. *arXiv preprint arXiv:2410.12557*, 2024.

Geng, Z., Deng, M., Bai, X., Kolter, J. Z., and He, K. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025a.

Geng, Z., Pokle, A., Luo, W., Lin, J., and Kolter, J. Z. Consistency models made easy. In *The Thirteenth International Conference on Learning Representations*, 2025b.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

Gutmann, M. and Hyvärinen, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 297–304, 2010.

He, J., Chen, W., Zhang, M., Barber, D., and Hernández-Lobato, J. M. Training neural samplers with reverse diffusive kl divergence. *arXiv preprint arXiv:2410.12456*, 2024.

Heek, J., Hoogeboom, E., and Salimans, T. Multistep consistency models. *arXiv preprint arXiv:2403.06807*, 2024.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., and Fleet, D. J. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022.

Huszár, F. Variational inference using implicit distributions. *arXiv preprint arXiv:1702.08235*, 2017.

Hyvärinen, A. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35: 26565–26577, 2022.

Karras, T., Aittala, M., Lehtinen, J., Hellsten, J., Aila, T., and Laine, S. Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24174–24184, 2024.

Kim, D., Lai, C.-H., Liao, W.-H., Murata, N., Takida, Y., Uesaka, T., He, Y., Mitsuji, Y., and Ermon, S. Consistency trajectory models: Learning probability flow ODE trajectory of diffusion. In *The Twelfth International Conference on Learning Representations*, 2024.

Krizhevsky, A. Learning multiple layers of features from tiny images. *Master’s thesis, University of Toronto*, 2009.

Li, L. and He, J. Bidirectional consistency models. *arXiv preprint arXiv:2403.18035*, 2024.

Li, X., Thickstun, J., Gulrajani, I., Liang, P. S., and Hashimoto, T. B. Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems*, 35:4328–4343, 2022.

Liu, H., Chen, Z., Yuan, Y., Mei, X., Liu, X., Mandic, D., Wang, W., and Plumley, M. D. Audioldm: Text-to-audio generation with latent diffusion models. *arXiv preprint arXiv:2301.12503*, 2023.

Liu, L., Ren, Y., Lin, Z., and Zhao, Z. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778*, 2022.

Lu, C. and Song, Y. Simplifying, stabilizing and scaling continuous-time consistency models. In *International Conference on Learning Representations*, 2025.

Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., and Zhu, J. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022.

Luo, W., Hu, T., Zhang, S., Sun, J., Li, Z., and Zhang, Z. Diff-instruct: A universal approach for transferring knowledge from pre-trained diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

Mescheder, L., Geiger, A., and Nowozin, S. Which training methods for gans do actually converge? In *International conference on machine learning*, pp. 3481–3490. PMLR, 2018.

Milgrom, P. and Segal, I. Envelope theorems for arbitrary choice sets. *Econometrica*, 70(2):583–601, 2002.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

Nichol, A. Q. and Dhariwal, P. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PMLR, 2021.

Nowozin, S., Cseke, B., and Tomioka, R. f-gan: Training generative neural samplers using variational divergence minimization. *Advances in neural information processing systems*, 29, 2016.

Ou, Z., Zhang, M., Zhang, A., Xiao, T. Z., Li, Y., and Barber, D. Improving probabilistic diffusion models with optimal covariance matching. *International Conference on Learning Representations*, 2025.

Poole, B., Jain, A., Barron, J. T., and Mildenhall, B. Dreamfusion: Text-to-3d using 2d diffusion. In *The Eleventh International Conference on Learning Representations*, 2022.

Qin, J. Inferences for case-control and semiparametric two-sample density ratio models. *Biometrika*, 85(3):619–630, 1998.

Robbins, H. E. An empirical bayes approach to statistics. In *Breakthroughs in Statistics: Foundations and basic theory*, pp. 388–394. Springer, 1992.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Roth, K., Lucchi, A., Nowozin, S., and Hofmann, T. Stabilizing training of generative adversarial networks through regularization. *Advances in neural information processing systems*, 30, 2017.

Salimans, T. and Ho, J. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.

Salimans, T., Mensink, T., Heek, J., and Hoogeboom, E. Multistep distillation of diffusion models via moment matching. *arXiv preprint arXiv:2406.04103*, 2024.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.

Sønderby, C. K., Caballero, J., Theis, L., Shi, W., and Huszár, F. Amortised map inference for image super-resolution. *arXiv preprint arXiv:1610.04490*, 2016.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Song, Y. and Dhariwal, P. Improved techniques for training consistency models. In *The Twelfth International Conference on Learning Representations*, 2024.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019.

Song, Y., Dhariwal, P., Chen, M., and Sutskever, I. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

Sugiyama, M., Suzuki, T., and Kanamori, T. *Density ratio estimation in machine learning*. Cambridge University Press, 2012.

Vincent, P. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

Wang, L., Zhang, M., Ou, Z., and Barber, D. Vardiu: A variational diffusive upper bound for one-step diffusion distillation. *arXiv preprint arXiv:2508.20646*, 2025.

Wang, Z., Zheng, H., He, P., Chen, W., and Zhou, M. Diffusion-GAN: Training GANs with diffusion. In *The Eleventh International Conference on Learning Representations*, 2023.

Wang, Z., Lu, C., Wang, Y., Bao, F., Li, C., Su, H., and Zhu, J. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in Neural Information Processing Systems*, 36, 2024.

Xiao, Z., Kreis, K., and Vahdat, A. Tackling the generative learning trilemma with denoising diffusion gans. *arXiv preprint arXiv:2112.07804*, 2021.

Xie, S., Xiao, Z., Kingma, D. P., Hou, T., Wu, Y. N., Murphy, K. P., Salimans, T., Poole, B., and Gao, R. Em distillation for one-step diffusion models. *arXiv preprint arXiv:2405.16852*, 2024.

Yin, T., Gharbi, M., Park, T., Zhang, R., Shechtman, E., Durand, F., and Freeman, B. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems*, 37:47455–47487, 2024.

Yu, L., Xie, T., Zhu, Y., Yang, T., Zhang, X., and Zhang, C. Hierarchical semi-implicit variational inference with application to diffusion model acceleration. *Advances in Neural Information Processing Systems*, 36, 2024.

Zhang, M., Bird, T., Habib, R., Xu, T., and Barber, D. Variational f-divergence minimization. *arXiv preprint arXiv:1907.11891*, 2019.

Zhang, M., Hayes, P., Bird, T., Habib, R., and Barber, D. Spread divergence. In *International Conference on Machine Learning*, pp. 11106–11116. PMLR, 2020.

Zhang, M., Zhang, A., Xiao, T. Z., Sun, Y., and McDonagh, S. Out-of-distribution detection with class ratio estimation. *arXiv preprint arXiv:2206.03955*, 2022.

Zheng, H., Nie, W., Vahdat, A., Azizzadenesheli, K., and Anandkumar, A. Fast sampling of diffusion models via operator learning. In *International conference on machine learning*, pp. 42390–42402. PMLR, 2023.

Zhou, L., Ermon, S., and Song, J. Inductive moment matching. *arXiv preprint arXiv:2503.07565*, 2025a.

Zhou, M., Zheng, H., Wang, Z., Yin, M., and Huang, H.

Score identity distillation: Exponentially fast distillation of pretrained diffusion models for one-step generation. In *Forty-first International Conference on Machine Learning*, 2024.

Zhou, M., Zheng, H., Gu, Y., Wang, Z., and Huang, H.

Adversarial score identity distillation: Rapidly surpassing the teacher in one step. In *The Thirteenth International Conference on Learning Representations*, 2025b.

A. Diffusive Kullback–Leibler (DiKL) Divergence

A.1. Validity of DiKL Divergence

We follow the derivation from Spread Divergence (Zhang et al., 2020) and provide a simple proof of the validity of the diffusive KL (DiKL) divergence. See Zhang et al. (2020) for a generalized proof that includes cases with non-absolutely continuous distributions and non-Gaussian kernels.

Recall that DiKL is defined as

$$\text{DiKL}_{\mathcal{K}}(q_{\theta}(x_0) \parallel p_d(x_0)) = \sum_{t=1}^T w(t) \text{KL}(q_{\theta}(x_t) \parallel p_d(x_t)), \quad (21)$$

where $w(t) > 0$, and the noisy model density $q_{\theta}(x_t)$ and noisy data density $p_d(x_t)$ are respectively defined as:

$$q_{\theta}(x_t) = \int k_t(x_t | x_0) q_{\theta}(x_0) dx_0, \quad (22)$$

$$p_d(x_t) = \int k_t(x_t | x_0) p_d(x_0) dx_0, \quad (23)$$

with $k_t(x_t | x_0)$ denoting the Gaussian transition kernel. We will show that

$$\text{DiKL}_{\mathcal{K}}(q_{\theta}(x_0) \parallel p_d(x_0)) = 0 \iff q_{\theta}(x_0) = p_d(x_0). \quad (24)$$

Since $w(t) > 0$ and the KL divergence is non-negative, it suffices to show that:

$$\text{KL}(q_{\theta}(x_t) \parallel p_d(x_t)) = 0 \iff q_{\theta}(x_t) = p_d(x_t). \quad (25)$$

First, by definition of KL divergence, we have

$$\text{KL}(q_{\theta}(x_t) \parallel p_d(x_t)) = 0 \iff q_{\theta}(x_t) = p_d(x_t). \quad (26)$$

Now, assume that $k_t(\epsilon) = \mathcal{N}(0, \sigma^2 I)$, and rewrite the noisy densities as convolutions:

$$q_{\theta}(x_t) = (q_{\theta} * k_t)(x_t), \quad (27)$$

$$p_d(x_t) = (p_d * k_t)(x_t). \quad (28)$$

Suppose $q_{\theta}(x_t) = p_d(x_t)$. Applying the Fourier transform \mathcal{F} , we obtain:

$$\mathcal{F}(q_{\theta} * k_t) = \mathcal{F}(q_{\theta}) \cdot \mathcal{F}(k_t), \quad (29)$$

$$\mathcal{F}(p_d * k_t) = \mathcal{F}(p_d) \cdot \mathcal{F}(k_t). \quad (30)$$

Since the characteristic function of Gaussian $\mathcal{F}(k_t) > 0$, we have:

$$q_{\theta}(x_t) = p_d(x_t) \iff \mathcal{F}(q_{\theta}) \cdot \mathcal{F}(k_t) = \mathcal{F}(p_d) \cdot \mathcal{F}(k_t) \iff \mathcal{F}(q_{\theta}) = \mathcal{F}(p_d) \iff q_{\theta}(x_0) = p_d(x_0). \quad (31)$$

This completes the proof.

A.2. Derivation of the Density-Ratio Gradient Estimator for DiKL Divergence

Without loss of generality, we first consider DiKL with a single kernel k_t . We will first follow He et al. (2024) and show that the gradient of DiKL w.r.t. the model parameter θ is given by

$$\nabla_{\theta} \text{DiKL}_{k_t}(q_{\theta} \parallel p_d) = \int q_{\theta}(x_t) (\nabla_{x_t} \log q_{\theta}(x_t) - \nabla_{x_t} \log p_d(x_t)) \frac{\partial x_t}{\partial \theta} dx_t. \quad (32)$$

Recall that the DiKL divergence for a single kernel k_t is defined as

$$\text{DiKL}_{k_t}(q_{\theta} \parallel p_d) = \int (\log q_{\theta}(x_t) - \log p_d(x_t)) q_{\theta}(x_t) dx_t. \quad (33)$$

We first reparameterize x_t as a function of z and ϵ :

$$x_t = \alpha_t g_\theta(z) + \sigma_t \epsilon_t \equiv h_\theta(z, \epsilon_t), \quad (34)$$

where $z \sim p(z) \equiv \mathcal{N}(z|0, I)$ and $\epsilon_t \sim p(\epsilon_t) \equiv \mathcal{N}(\epsilon_t|0, I)$. It then follows that

$$\nabla_\theta \text{DiKL}_{k_t}(q_\theta(x_0) \parallel p_d(x_0)) = \nabla_\theta \int (\log q_\theta(x_t) - \log p_d(x_t)) q_\theta(x_t) dx_t \quad (35)$$

$$= \nabla_\theta \int \int \int (\log q_\theta(x_t) - \log p_d(x_t)) \delta(x_t - h_\theta(z, \epsilon_t)) p(z) p(\epsilon_t) dx_t dz d\epsilon \quad (36)$$

$$= \nabla_\theta \int \int (\log q_\theta(x_t) - \log p_d(x_t)) \Big|_{x_t=h_\theta(z, \epsilon_t)} p(z) p(\epsilon_t) dz d\epsilon \quad (37)$$

$$= \int \left(\nabla_\theta \log q_\theta(x_t) + \nabla_{x_t} \log q_\theta(x_t) \frac{\partial x_t}{\partial \theta} - \nabla_{x_t} \log p_d(x_t) \frac{\partial x_t}{\partial \theta} \right) q_\theta(x_t) dx_t \quad (38)$$

$$= \int \left(\nabla_{x_t} \log q_\theta(x_t) \frac{\partial x_t}{\partial \theta} - \nabla_{x_t} \log p_d(x_t) \frac{\partial x_t}{\partial \theta} \right) q_\theta(x_t) dx_t, \quad (39)$$

where the last line follows since

$$\int \nabla_\theta \log q_\theta(x_t) q_\theta(x_t) dx_t = \int \nabla_\theta q_\theta(x_t) dx_t = \nabla_\theta \int q_\theta(x_t) dx_t = \nabla_\theta 1 = 0. \quad (40)$$

For a set of kernels $\mathcal{K} = \{k_1, \dots, k_T\}$, it then follows that

$$\nabla_\theta \text{DiKL}_{\mathcal{K}}(q_\theta \parallel p_d) = \sum_{t=1}^T w(t) \int q_\theta(x_t) (\nabla_{x_t} \log q_\theta(x_t) - \nabla_{x_t} \log p_d(x_t)) \frac{\partial x_t}{\partial \theta} dx_t \quad (41)$$

$$= \sum_{t=1}^T w(t) \int q_\theta(x_t) \left(\nabla_{x_t} \log \frac{q_\theta(x_t)}{p_d(x_t)} \right) \frac{\partial x_t}{\partial \theta} dx_t \quad (42)$$

$$= \sum_{t=1}^T w(t) \int q_\theta(x_t) \nabla_{x_t} \text{logit}(1 - c_*(x_t, t)) \frac{\partial x_t}{\partial \theta} dx_t, \quad (43)$$

where the last line follows by Eq. 17. This completes the proof.

B. Diffusive Jensen-Shannon (DiJS) Divergence

B.1. Validity of DiJS Divergence

We follow a similar derivation for the validity of DiKL in Appendix A.1. Recall that DiJS is defined as

$$\text{DiJS}_{\mathcal{K}}(q_\theta(x_0) \parallel p_d(x_0)) = \sum_{t=1}^T w(t) \text{JS}(q_\theta(x_t) \parallel p_d(x_t)), \quad (44)$$

where $w(t) > 0$, and the noisy model density $q_\theta(x_t)$ and noisy data density $p_d(x_t)$ are respectively defined as:

$$q_\theta(x_t) = \int k_t(x_t | x_0) q_\theta(x_0) dx_0, \quad (45)$$

$$p_d(x_t) = \int k_t(x_t | x_0) p_d(x_0) dx_0, \quad (46)$$

with $k_t(x_t | x_0)$ denoting the Gaussian transition kernel. We will show that

$$\text{DiJS}_{\mathcal{K}}(q_\theta(x_0) \parallel p_d(x_0)) = 0 \iff q_\theta(x_0) = p_d(x_0). \quad (47)$$

Since $w(t) > 0$ and the JS divergence is non-negative, it suffices to show that:

$$\text{JS}(q_\theta(x_t) \parallel p_d(x_t)) = 0 \iff q_\theta(x_t) = p_d(x_t). \quad (48)$$

First, by definition of JS divergence, we have

$$\text{JS}(q_\theta(x_t) \parallel p_d(x_t)) = 0 \iff q_\theta(x_t) = p_d(x_t). \quad (49)$$

Now, assume that $k_t(\epsilon) = \mathcal{N}(0, \sigma^2 I)$, and rewrite the noisy densities as convolutions:

$$q_\theta(x_t) = (q_\theta * k_t)(x_t), \quad (50)$$

$$p_d(x_t) = (p_d * k_t)(x_t). \quad (51)$$

Suppose $q_\theta(x_t) = p_d(x_t)$. Applying the Fourier transform \mathcal{F} , we obtain:

$$\mathcal{F}(q_\theta * k_t) = \mathcal{F}(q_\theta) \cdot \mathcal{F}(k_t), \quad (52)$$

$$\mathcal{F}(p_d * k_t) = \mathcal{F}(p_d) \cdot \mathcal{F}(k_t). \quad (53)$$

Since the characteristic function of Gaussian $\mathcal{F}(k_t) > 0$, we have:

$$q_\theta(x_t) = p_d(x_t) \iff \mathcal{F}(q_\theta) \cdot \mathcal{F}(k_t) = \mathcal{F}(p_d) \cdot \mathcal{F}(k_t) \iff \mathcal{F}(q_\theta) = \mathcal{F}(p_d) \iff q_\theta(x_0) = p_d(x_0). \quad (54)$$

This completes the proof.

B.2. Derivation of the Density-Ratio Gradient Estimator for DiJS Divergence

Recall that the Jensen–Shannon (JS) divergence between two densities q and p is

$$\text{JS}(q \parallel p) = \frac{1}{2} \text{KL}\left(q \parallel \frac{q+p}{2}\right) + \frac{1}{2} \text{KL}\left(p \parallel \frac{q+p}{2}\right). \quad (55)$$

Define the mixture density at diffusion time t as

$$m(x_t) \equiv \frac{q_\theta(x_t) + p_d(x_t)}{2}. \quad (56)$$

We first consider a single diffusion kernel k_t and then extend the result to the weighted multi-kernel setting.

Density-ratio identity via an optimal classifier. Following the GAN view (Goodfellow et al., 2014), consider a time-conditional binary classification problem between data and model samples at diffusion time t , where $y = 1 \sim p_d(x_t)$ and $y = 0 \sim q_\theta(x_t)$, let the Bayes-optimal classifier be $c_*(x_t, t) \equiv p(y = 1 \mid x_t, t)$. Then

$$c_*(x_t, t) = \frac{p_d(x_t)}{p_d(x_t) + q_\theta(x_t)}, \quad 1 - c_*(x_t, t) = \frac{q_\theta(x_t)}{p_d(x_t) + q_\theta(x_t)}. \quad (57)$$

Consequently, the density ratio appearing in $\text{KL}(q_\theta \parallel m)$ can be written as

$$\frac{q_\theta(x_t)}{m(x_t)} = \frac{q_\theta(x_t)}{\frac{1}{2}(p_d(x_t) + q_\theta(x_t))} = 2 \frac{q_\theta(x_t)}{p_d(x_t) + q_\theta(x_t)} = 2(1 - c_*(x_t, t)). \quad (58)$$

Gradient of the DiJS objective under the GAN approximation. Consider the first JS term:

$$\frac{1}{2} \text{KL}(q_\theta(x_t) \parallel m(x_t)) = \frac{1}{2} \int q_\theta(x_t) \log\left(\frac{q_\theta(x_t)}{m(x_t)}\right) dx_t. \quad (59)$$

Under the standard GAN approximation, we treat the density-ratio estimate (equivalently the optimal classifier) as fixed when differentiating w.r.t. θ , i.e., we detach $m(x_t)$.¹ Using the same reparameterization argument as in Appendix A.2, we obtain

$$\nabla_\theta \frac{1}{2} \text{KL}(q_\theta(x_t) \parallel m(x_t)) \approx \frac{1}{2} \int q_\theta(x_t) \nabla_{x_t} \log\left(\frac{q_\theta(x_t)}{m(x_t)}\right) \frac{\partial x_t}{\partial \theta} dx_t. \quad (60)$$

¹This is justified by the envelope theorem (Milgrom & Segal, 2002) for the inner maximization over the classifier parameters, and corresponds to a two-timescale training regime where the classifier tracks its optimum for the current θ . See also Theorem 2.4 in (Arjovsky & Bottou, 2017) for a discussion of vanishing gradients in the GAN setting.

Substituting (58) and dropping the constant $\log 2$ (whose gradient w.r.t. x_t is zero), this becomes

$$\nabla_{\theta} \frac{1}{2} \text{KL}(q_{\theta}(x_t) \| m(x_t)) \approx \frac{1}{2} \int q_{\theta}(x_t) \nabla_{x_t} \log(1 - c_*(x_t, t)) \frac{\partial x_t}{\partial \theta} dx_t. \quad (61)$$

For the second JS term, $\frac{1}{2} \text{KL}(p_d \| m)$, the data density p_d does not depend on θ , and under the same $\text{detach-}m$ approximation its contribution to ∇_{θ} is ignored. Therefore, the resulting gradient estimator is given by (61).

C. Discussion on the Consistency of the Diffusive Density-Ratio Estimator

Appendices A and B establish that DiKL and DiJS are valid divergences, i.e., $\text{DiKL}_{\mathcal{K}}(q_{\theta} \| p_d) = 0$ or $\text{DiJS}_{\mathcal{K}}(q_{\theta} \| p_d) = 0$ implies $q_{\theta}(x_0) = p_d(x_0)$. Therefore, if the score difference term used in our gradient estimators is computed exactly (equivalently, if the underlying density ratio is estimated consistently), driving the corresponding diffusive divergence to zero yields convergence of the model distribution to the data distribution.

It remains to justify consistency of the density-ratio (class-ratio) estimator. We estimate $\frac{q_{\theta}(x_t)}{p_d(x_t)}$ via a time-conditional binary logistic regression classifier (equivalently, noise-contrastive estimation). Consistency of this maximum-likelihood estimator follows from Theorem 2 of [Gutmann & Hyvärinen \(2010\)](#) under conditions (a)–(c) therein.

Theorem C.1 (Consistency of the diffusive density-ratio estimator). *Fix any noise level $t > 0$. Let $p_1(x_t) = p_d(x_t)$ and $p_0(x_t) = q_{\theta}(x_t)$, and let $c_{\hat{\eta}}(x_t, t)$ be a well-specified time-conditional logistic classifier trained by maximum likelihood to distinguish samples from p_1 versus p_0 with nonzero class priors. Under standard regularity conditions (uniform convergence of the empirical objective and a full-rank information matrix; see Theorem 2 of [Gutmann & Hyvärinen \(2010\)](#)), the MLE $\hat{\eta}$ is consistent and the implied log-density-ratio estimate*

$$\widehat{\log \frac{q_{\theta}(x_t)}{p_d(x_t)}} = \log \frac{1 - c_{\hat{\eta}}(x_t, t)}{c_{\hat{\eta}}(x_t, t)}$$

converges (in probability) to the true $\log \frac{q_{\theta}(x_t)}{p_d(x_t)}$. Consequently, $\nabla_{x_t} \log \frac{q_{\theta}(x_t)}{p_d(x_t)}$ is consistently estimated by $\nabla_{x_t} \log \frac{1 - c_{\hat{\eta}}(x_t, t)}{c_{\hat{\eta}}(x_t, t)}$, yielding a consistent estimate of the score difference $\nabla_{x_t} \log q_{\theta}(x_t) - \nabla_{x_t} \log p_d(x_t)$.

Condition (a) (support overlap) requires that $p_n(u) > 0$ whenever $p_d(u) > 0$ (equivalently, $\text{supp}(p_d) \subseteq \text{supp}(p_n)$). This is the main subtlety for implicit one-step generators, where $q_{\theta}(x_0)$ may not share support with $p_d(x_0)$. In our setting, we estimate the ratio in the noised space: for any $t > 0$, both $q_{\theta}(x_t)$ and $p_d(x_t)$ are obtained by convolving the respective x_0 -distributions with a Gaussian kernel, and hence are absolutely continuous with overlapping support on x_t . Thus, condition (a) is automatically satisfied for the diffused distributions, and our method is *consistent under weaker conditions* than requiring support overlap at $t = 0$. Excluding $t = 0$ in the DiKL/DiJS definition ensures that ratio estimation is always performed in this well-supported noisy space, enabling convergence under weaker conditions even when $q_{\theta}(x_0)$ and $p_d(x_0)$ do not share support.

D. Detailed Setup of Toy Experiment

This section provides the detailed experimental setup for the gradient bias study in Section 3.2.

Data Distribution. The data distribution $p_d(x_0)$ is a mixture of Gaussians with 40 components in 2D. Component means are sampled uniformly in $[-40, 40]^2$ and variances are sampled log-uniformly, providing a multi-modal distribution with varying local densities. This setup allows closed-form computation of the true score $\nabla_{x_t} \log p_d(x_t)$ at any noise level t .

One-Step Diffusion Model. The one-step diffusion model $q_{\theta}(x_0)$ is an implicit generative model parameterized by a 4-layer MLP with hidden dimension 400 and SiLU activations, conditioned on the noise level t . The model has been pre-trained to approximate p_d . Since q_{θ} lacks a closed-form density, we estimate its score using kernel density estimation (KDE) with 10,000 generated samples and bandwidth selected via Scott's rule. This provides the ground-truth score difference:

$$\Delta s_t^* \equiv \Delta s^*(x_t, t) = \nabla_{x_t} \log q_{\theta}(x_t) - \nabla_{x_t} \log p_d(x_t). \quad (62)$$

Methods Compared:

1. **DiffRatio (ours):** A time-conditioned classifier $c_\eta(x_t, t)$ is trained to distinguish between samples from $q_\theta(x_t)$ and $p_d(x_t)$. The classifier is a 4-layer MLP with hidden dimension 400 and SiLU activations, taking concatenated (x_t, t) as input and outputting a scalar logit. The score difference is obtained via Eq. 17: $\widehat{\Delta s_t} = \nabla_{x_t} \text{logit}(1 - c_\eta(x_t, t))$.
2. **VSD:** Two separate score networks $s_{\psi_2}^{q_\theta}(x_t, t)$ and $s_{\psi_1}^{p_d}(x_t, t)$ are trained independently using DSM (Eqs. 12 and 13), and the score difference is computed as their difference $\widehat{\Delta s_t} = s_{\psi_2}^{q_\theta}(x_t, t) - s_{\psi_1}^{p_d}(x_t, t)$. Both networks use the same architecture as the classifier.

Training Details. All networks are trained for 10,000 iterations with batch size 1024 using Adam optimizer with learning rate 10^{-4} . Noise levels are sampled from a power schedule $t \sim t_{\min} + u^{1.5} \cdot (t_{\max} - t_{\min})$ where $u \sim \mathcal{U}(0, 1)$, $t_{\min} = 0.1$, and $t_{\max} = 20.0$. The classifier is trained with binary cross-entropy loss using softplus for numerical stability.

Evaluation. We evaluate each method by comparing estimated score differences against the KDE-based ground truth at 10 logarithmically-spaced noise levels in $[0.1, 20]$. At each noise level, we sample 1,000 evaluation points from the noised student distribution. We report the cosine similarity $\langle \widehat{\Delta s_t}, \Delta s_t^* \rangle / (\|\widehat{\Delta s_t}\| \|\Delta s_t^*\|)$, measuring directional alignment of the gradient estimates.

E. Detailed Setup of Image Generation Experiments

This section provides the detailed experimental setup for the image generation experiments in Section 5.

We evaluate the performance of our proposed method by training one-step image diffusion models on two standard benchmarks: (1) unconditional generation on the CIFAR-10 (32×32) dataset (Krizhevsky, 2009), (2) class-label-conditioned generation on the ImageNet (64×64 and 512×512) dataset (Deng et al., 2009).

Our one-step diffusion model adopts the same neural network architecture as the pre-trained score model. Our implementation builds upon the EDM and EDM2 codebases (Karras et al., 2022; 2024). Specifically, we employ the EDM U-Net model architecture (Karras et al., 2022) for CIFAR-10 and ImageNet 64×64 , and use the EDM2-M U-Net model architecture (Karras et al., 2024) for ImageNet 512×512 . We use the EDM preconditioning for the inputs. We choose not to use the EDM output preconditioning as we are training a one-step generator which outputs images that differ significantly from the input noise.

Our density-ratio estimator $c_\eta(x_t, t)$ is implemented using the encoder portion of the teacher’s U-Net architecture with a sigmoid function attached to the end to produce a scalar output between 0 and 1. This network is approximately half the size of the full U-Net used for score estimation, leading to improved computational and memory efficiency. The one-step generator $g_\theta(z)$ is initialized using the weights from the pre-trained score model with the diffusion time argument $t_{\text{init}}=2.5$ fixed throughout training and sampling.

We consider three training criteria in our framework (DiKL/DiJS/DiRM) on CIFAR-10, and use the best-performing criterion (DiJS) for higher-dimensional ImageNet experiments (64×64 and 512×512). We use the variance-exploding (VE) noise schedule to define the diffusive divergences with $t_{\max}=1000$. We follow standard hyperparameter settings for training generative models on CIFAR-10 and ImageNet as detailed in Karras et al. (2022; 2024). We use a batch size of 64 for CIFAR-10 and 4,096 for ImageNet. The weighting function $w(t)=\sigma_t^2$ is used for both datasets. For CIFAR-10 and ImageNet 64×64 , we use a fixed learning rate of 5×10^{-5} for both generator and density-ratio network. For ImageNet 512×512 , we use a fixed learning rate of 10^{-4} for the density-ratio network; for the generator we use a learning rate of 10^{-4} at the beginning and reduce the learning rate by a factor of 10 when the FID does not improve for 500 ticks. We adopt a single-step update strategy for the density-ratio estimator throughout our experiments, consistent with previous works (Luo et al., 2024; Zhou et al., 2024). All experiments are conducted on 8 NVIDIA H100 80GB GPUs.

F. Analysis of the Role of Weight Initialization in Training One-Step Diffusion Models

Our one-step diffusion model was initialized with the pre-trained score model’s weights. We observed that training from random initialization led to mode collapse; see Figure 5c for an example of mode collapse. One possible explanation is that mode collapse arises from the training objectives (i.e., reverse KL or JS divergence), a phenomenon also observed in GAN literature (Goodfellow et al., 2014). To understand why initializing the one-step model with the pre-trained score model’s weights prevents mode collapse in the training process, we investigate the following two hypotheses.

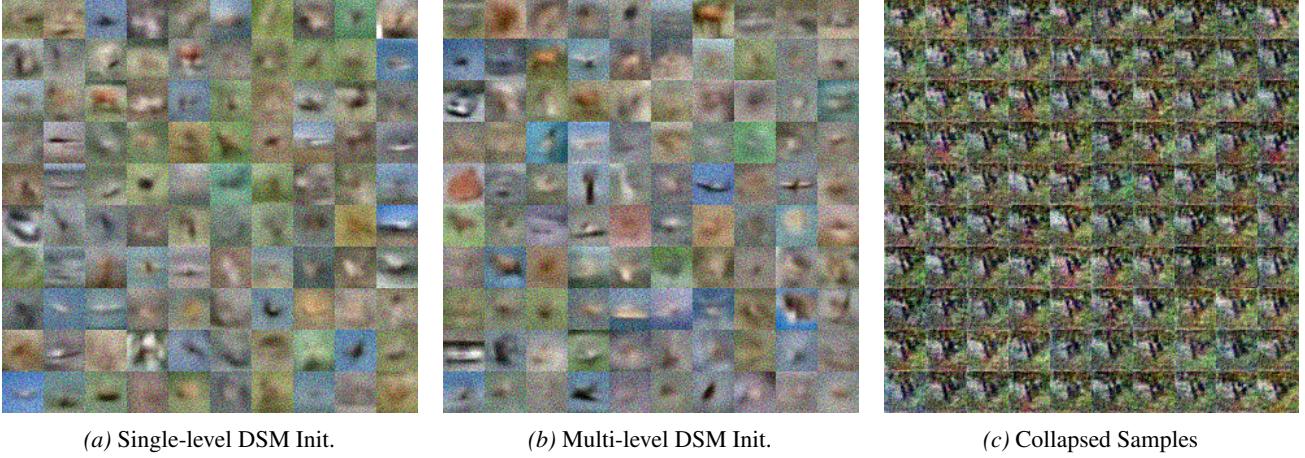


Figure 5. Visualization of different initializations and collapsed samples on CIFAR-10.

Function Space Hypothesis. *Teacher weight initialization provides a more structured latent-to-output functional mapping, i.e., different locations in the latent space are initially mapped to distinct images, preventing mode collapse.*

This hypothesis originally arose from visualizing initialized samples as shown in Figure 5a, showing that initialization already induces diverse mappings, with the one-step model training stage primarily refining these initializations into sharper images. Somewhat surprisingly, however, we find that functional initialization alone is insufficient to prevent mode collapse. To show this, instead of training the teacher model across different diffusion time steps t and selecting a single time step t_{init} for initialization, we only pre-train the teacher model at the selected time step t_{init} and use its weight to initialize the one-step model. This setup ensures identical latent-to-output mappings for the one-step model at initialization as shown in Figure 5b. However, with this initialization, the one-step model still exhibits mode collapse early in the one-step model training stage, which suggests that the functional mapping perspective alone does not fully explain the mode-collapse issue.

Feature Space Hypothesis. *Teacher weight initialization provides a rich set of multi-level features learned when pre-training the teacher diffusion model, which help prevent mode collapse.*

To verify this hypothesis and isolate the role of learned features from functional mapping effects, we pre-train the teacher model on CIFAR-100 while excluding all classes that overlap with CIFAR-10. This ensures that images from the target classes that the one-step model aims to generate are absent during pre-training, allowing us to focus solely on the contribution of the learned features. We train the teacher model using increasingly larger subsets of CIFAR-100 with (10, 50, 90) classes, creating a setting with increasing feature diversity. Table 5 shows the performance of our one-step model on CIFAR-10 initialized with the weights of teacher models trained on varying numbers of CIFAR-100 classes. We find that when the teacher model is trained on only 10 classes, mode collapse still occurs. However, as the number of training classes increases, the one-step model no longer collapses, indicating that feature richness plays a crucial role in preventing mode collapse. Nevertheless, despite mitigating mode collapse, this initialization strategy achieves an FID of 6.01 when the teacher model is pre-trained on all 90 non-overlapping classes in CIFAR-100, which is significantly worse than the FID (2.39) obtained when directly using CIFAR-10 as the pre-training dataset. This suggests that while feature richness is essential for stabilizing training, functional mapping initialization remains important for achieving higher sample quality.

Table 5. Performance of one-step models trained by DiJS with different initializations on various CIFAR subsets.

Initialization method	Initialization dataset	FID
No initialization	-	collapsed
Single-level DSM	full CIFAR-10	collapsed
Multi-level DSM	10 classes in CIFAR-100	collapsed
	50 classes in CIFAR-100	6.20
	90 classes in CIFAR-100	6.01
	full CIFAR-10	2.39

G. Additional Image Generation Results



Figure 6. Visualization of samples from DiffRatio-DiJS trained on CIFAR10 (FID=2.39, IS=9.93).

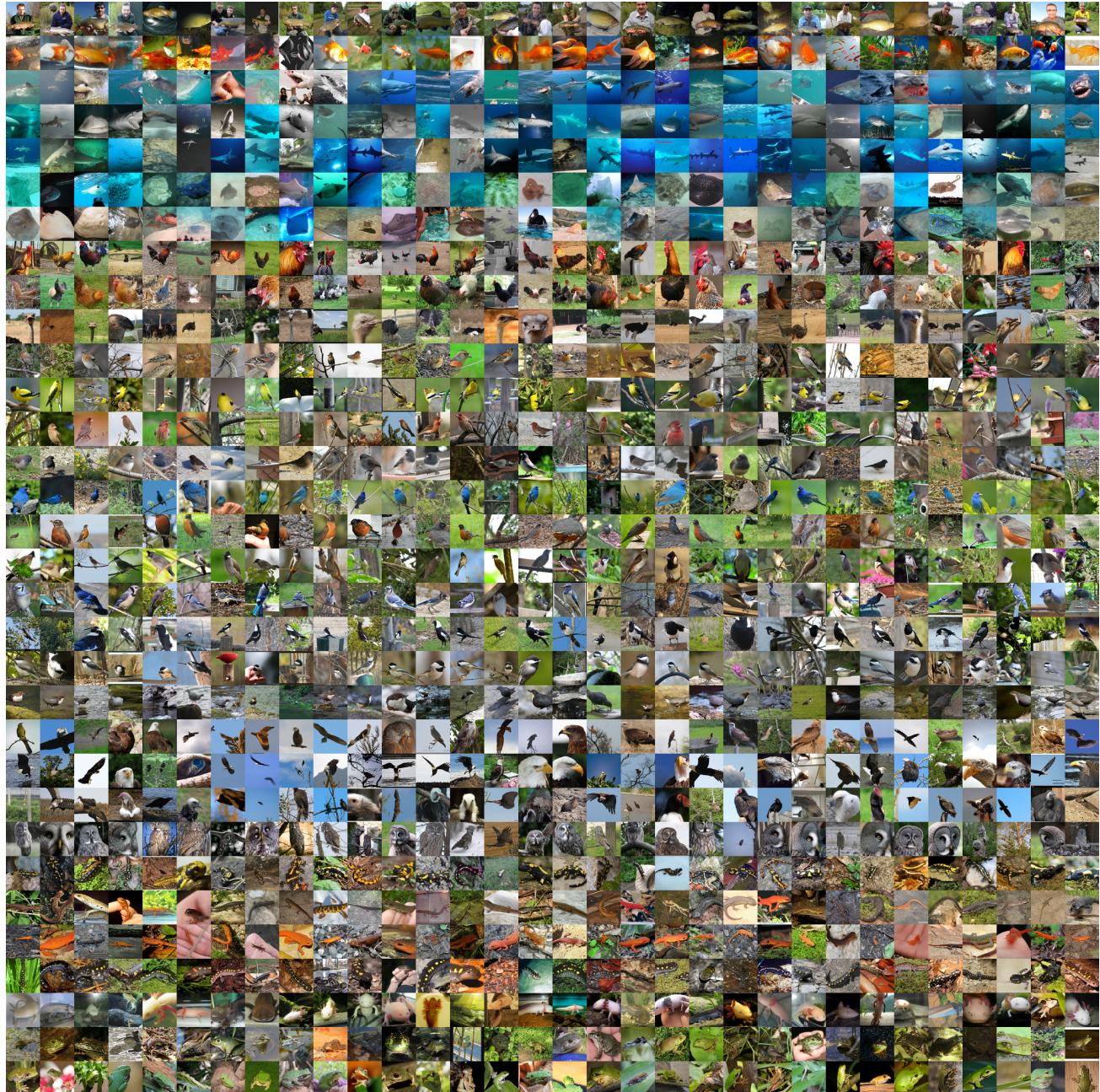


Figure 7. Visualization of samples from DiffRatio-DiJS trained on ImageNet 64×64 (FID=1.54).



Figure 8. Visualization of samples from DiffRatio-DiJS-M trained on ImageNet 512×512 (FID=1.41).