

Improving Discriminator Guidance in Diffusion Models

Alexandre Verine¹ (✉), Ahmed Mehdi Inane², Florian Le Bronnec³, Benjamin Negrevergne³, and Yann Chevaleyre³

¹ École Normale Supérieure Paris, PSL University, Paris, France
`alexandre.verine@ens.fr`

² Mila, Quebec AI Institute, Université de Montréal, Montreal, Canada
`mehdi-inane.ahmed@mila.quebec`

³ LAMSADE, CNRS, Université Paris-Dauphine-PSL, Paris, France
`{florian.le-bronnec,benjamin.negrevergne,yann.chevaleyre}@dauphine.psl.eu`

Abstract. Discriminator Guidance has become a popular method for efficiently refining pre-trained Score-Matching Diffusion models. However, in this paper, we demonstrate that the standard implementation of this technique does not necessarily lead to a distribution closer to the real data distribution. Specifically, we show that training the discriminator using Cross-Entropy loss, as commonly done, can in fact increase the Kullback-Leibler divergence between the model and target distributions, particularly when the discriminator overfits. To address this, we propose a theoretically sound training objective for discriminator guidance that properly minimizes the KL divergence. We analyze its properties and demonstrate empirically across multiple datasets that our proposed method consistently improves over the conventional method by producing samples of higher quality.⁴

Keywords: Diffusion Models · Discriminator Guidance

1 Introduction

Diffusion based generative models have proven to be effective in numerous fields, including image and video generation [5, 8, 9, 15], graphs [19], and audio synthesis [17], among others. Diffusion models are trained to iteratively denoise samples from a noise distribution to approximate the target data distribution. They have gained success in the generative modeling community for their ability to generate both high-quality samples, and to cover well the estimated distribution. However, this comes with a high computational cost both for sampling new data (which requires multiple denoising steps) and for training (which requires training the model multiple time for each level of denoising).

Consequently, several approaches focus on post-training strategies to refine a pre-trained model’s distribution at a lower computational cost [6, 20, 26, 32].

⁴ Code: <https://github.com/AlexVerine/BoostDM> and Supplementary Materials <https://arxiv.org/abs/2503.16117>

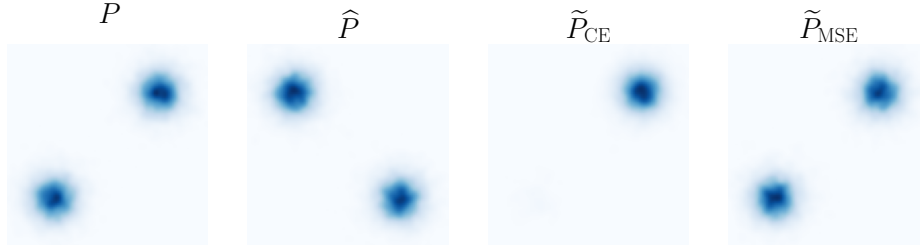


Fig. 1: Illustration of generating samples by a reverse diffusion process to sample from the target distribution P using true score function $\nabla \log p_t$, from the learned distribution $\hat{P}$ using the learned score function $\mathbf{s}_\theta$, from a poor approximation $\tilde{P}_{\text{CE}}$ using a refined score $\mathbf{s}_\theta + \nabla d_\phi$ with low cross-entropy $\mathcal{L}_{\text{CE}^d(\phi)}$ and, finally, from a good approximation $\tilde{P}_{\text{MSE}}$ using a refined score $\mathbf{s}_\theta + \nabla d_\phi$ with low loss $\mathcal{L}_{\text{MSE}^d(\phi)}$ introduced in Section 5.

Among these, Discriminator Guidance (DG) [14] has recently emerged as a promising refinement method, demonstrating strong empirical results [13, 14, 28]. This approach refines the generation process by training a discriminator to distinguish generated samples from real samples at different diffusion steps. The discriminator is then used to estimate the log density ratio between the real data distribution P and the learned distribution $\hat{P}$, with the gradient of this estimate acting as a correction term during generation.

However, this introduces a fundamental discrepancy between training and inference: while the discriminator is trained to approximate the density ratio, inference relies on its gradient, which is **not necessarily a reliable estimate of the target gradient**. As a result, refining the pre-trained model in this way can degrade generation quality.

In this work, we make the following contributions:

- We formally demonstrate in Theorem 1 that training a discriminator to minimize Cross-Entropy and using it for refinement can lead to arbitrarily poor results in the final distribution.
- In Theorem 2, we show that overfitting the discriminator is a sufficient condition for the refined distribution to deteriorate in terms of KL divergence compared to the pre-trained model.
- We propose a reformulation of the DG objective, introducing a new optimization criterion that directly improves the refined distribution by leveraging the gradient of the log-likelihood ratio rather than its value (its benefits over standard Cross-Entropy minimization are illustrated on Figure 1).
- Finally, we demonstrate that our method enhances sample quality in diffusion models across benchmark image generation datasets, including CIFAR-10, FFHQ, and AFHQ-v2.

By providing a theoretically grounded alternative to the standard DG objective, our approach improves both the understanding and effectiveness of discriminator-guided refinement in generative modeling.

Notations: We denote P , $\hat{P}$ and $\tilde{P}$ the target, the learned and the refined distributions defined on $\mathbb{R}^d$. Their densities are denoted p , $\hat{p}$ and $\tilde{p}$. We will denote with an indexed t the diffused distribution P_t , $\hat{P}_t$ and $\tilde{P}_t$ at time t and their densities p_t , $\hat{p}_t$ and $\tilde{p}_t$. We denote by $\mathcal{D}_{\text{KL}}$ the Kullback-Leibler divergence.

2 Related Works

In a trained diffusion model, the divergence between the learned distribution $\hat{P}$ and the target distribution P arises from two primary sources of error: *sampling* errors, which stem from the discretization scheme used to solve the stochastic differential equation for sample generation, and *estimation* errors, which originate from the discrepancy between the model’s final estimate of the score, $\nabla \log \hat{p}_t$ and the target score $\nabla \log p_t$. This section reviews existing approaches that aim to mitigate this discrepancy by addressing one or both sources of error:

- **Sampling strategies:** A first line of work attempts to correct the sampling error due to the discretization of the backward SDE (Equation (2)). Since solving this equation numerically is fundamental to sample generation in diffusion models, the choice of the discretization scheme significantly impacts sample quality. Traditional solvers, such as the Euler-Maruyama method [16], are widely used but can introduce bias or require a large number of function evaluations to achieve high fidelity. To address these limitations, researchers have explored improved numerical schemes specifically tailored to score-based generative modeling. For example, Jolicœur-Martineau et al. [11] propose an adaptive step size strategy for the SDE solver, allowing for more efficient sample generation by dynamically adjusting the discretization step. This approach reduces numerical error and enables high-quality sample synthesis in fewer iterations compared to fixed-step solvers. Xu et al. [33] proposed the Restart sampling algorithm, which alternates between adding noise through additional forward steps and strictly following a backward ordinary differential equation (ODE). This approach balances discretization errors and the contraction property of stochasticity, resulting in accelerated sampling speeds while maintaining or enhancing sample quality.
- **Correcting the estimation:** Another line of work attempts to correct the estimated score function with additional information from auxiliary models. For instance, the classifier-guided approach refines score estimation by incorporating gradients from an external classifier trained to predict class probabilities. These gradients are used to adjust the predicted score of the diffusion model, steering the sampling process toward more semantically meaningful outputs [6]. Since the estimation error term is given by the gradient of the log density ratio between the target distribution P and model distribution $\hat{P}$, Kim et al.

[14] introduced *discriminator guidance*, a method leveraging a traditional density estimation technique by training a discriminator d_ϕ . The gradient of the output of d_ϕ is then added to the score estimate during sampling, which provably reduces the discrepancy between P and $\hat{P}$. This method builds on a broad body of literature that leverages discriminators to estimate the log-density ratio for enhancing the generation process of Generative Adversarial Networks [2, 3, 4, 29, 30]. Recent works have proposed complementary algorithmic improvements to this approach: Kelvinius and Lindsten [13] propose a sequential Monte-Carlo based algorithm to correct the estimation errors of the discriminator for autoregressive diffusion models, and Tsonis et al. [28] propose a hybrid algorithm that merges discriminator guidance and a scaling factor to mitigate the exposure bias induced by the training process of diffusion models. Our work focuses on improving the training of the discriminator itself, and can be integrated with all the aforementioned methods to further improve sample quality and model performance.

3 Background on Score-matching Diffusion Models

3.1 Diffusion Process

Let $\{\mathbf{x}_t\}_{t \in [0, T]}$ be a diffusion process defined by an Itô SDE:

$$d\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t, t)dt + g(t)d\mathbf{w}, \quad (1)$$

where $\mathbf{f}(\cdot, t) \in \mathbb{R}^d \rightarrow \mathbb{R}^d$ is the drift, $g(t) : \mathbb{R} \rightarrow \mathbb{R}$ is the diffusion coefficient and $\mathbf{w} \in \mathbb{R}^d$ is a standard Wiener process. It defines a sequence of distributions $\{P_t\}_{t \in [0, T]}$, with densities p_t . With this definition, the target distribution is $P = P_0$ and $\mathbf{f}$, g and T are chosen such that P_T tends toward a tractable distribution Q with density q . In practice, Q is a normal distribution in $\mathbb{R}^d$. Using the time-reversed diffusion process [1] of Equation (1) we can define a generation scheme to sample from the target distribution P . We have the reverse SDE :

$$d\mathbf{x}_t = [f(\mathbf{x}_t, t) - g(t)^2 \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)] dt + g(t)d\bar{\mathbf{w}}, \quad (2)$$

where $d\bar{\mathbf{w}}$ denotes a different standard Wiener process.

3.2 Score-Based Generative Models

Taking advantage of the reverse SDE in Equation (2), we can train generative models based on scores by training a neural network $\mathbf{s}_\theta(\mathbf{x}_t, t)$ to estimate the value of the score $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$. To do so, we train U-Net Ronneberger et al. [23], a function $\mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}^d$, to minimize the *score matching loss* [10]:

$$\mathcal{L}_{\text{SM}}^s(\theta) = \frac{1}{2} \int_0^T \lambda(t) \mathbb{E}_{P_t} [\|\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) - \mathbf{s}_\theta(\mathbf{x}_t, t)\|_2^2] dt. \quad (3)$$

Note that the weighting function $\lambda : [0, T] \rightarrow]0, +\infty[$ depends on the type of SDE. Although the score matching loss is not directly computable, there exist methods to estimate it using Sliced Score Matching [25]. Vincent [31] shows that the score matching loss is equivalent, up to a constant additive term independent of θ , to the *denoising score matching* loss:

$$\mathcal{L}_{\text{MSE}}^s(\theta) = \int_0^T \lambda(t) \mathbb{E}_{P_0, P_t|\mathbf{x}_0} [\|\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{x}_0) - \mathbf{s}_\theta(\mathbf{x}_t, t)\|_2^2] dt, \quad (4)$$

where $P_t|\mathbf{x}_0$ is conditional distribution of $\mathbf{x}_t$ given $\mathbf{x}_0$. This distribution is typically chosen to be a Gaussian distribution with a mean depending on $\mathbf{x}_0$ and t and a variance depending on t . Therefore, this loss is based on the mean square error of the denoised reconstruction, which is widely used in practice as it is easier to compute and optimize [12]. The learned score function defines a new reverse diffusion process:

$$d\mathbf{x}_t = [\mathbf{f}(\mathbf{x}_t, t) - g(t)^2 \mathbf{s}_\theta(\mathbf{x}_t, t)] dt + g(t) d\bar{\mathbf{w}}. \quad (5)$$

It defines learned distributions $\{\hat{P}_t\}_{t \in [0, T]}$ with densities $\hat{p}_t$ such that:

$$\begin{cases} \hat{p}_T(\mathbf{x}_T) = q(\mathbf{x}_T), \\ \nabla_{\mathbf{x}_t} \log \hat{p}_t(\mathbf{x}_t, t) = \mathbf{s}_\theta(\mathbf{x}_t, t). \end{cases} \quad (6)$$

In general, the induced distribution $\hat{P}_0 = \hat{P}$ does not perfectly match the target distribution. Song et al. [24] shows that under the right assumptions (detailed in Appendix A.1), the Kullback-Leibler divergence is given by:

$$\mathcal{D}_{\text{KL}}(P\|\hat{P}) = \mathcal{D}_{\text{KL}}(P_T\|Q) + \frac{1}{2} \int_0^T g^2(t) \mathbb{E}_{P_t} [\|\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) - \mathbf{s}_\theta(\mathbf{x}_t, t)\|_2^2] dt. \quad (7)$$

In practice, the score function $\mathbf{s}_\theta$ is learned using a U-Net architecture [23] trained to minimize $\mathcal{L}_{\text{MSE}}^s$ and thus helps to reduce the dissimilarity between the learned distribution $\hat{P}$ and the target distribution P .

3.3 Discriminator Guided Diffusion

To enhance the generative process and minimize the discrepancy between the generated distribution $\hat{P}$ and the target distribution P , Kim et al. [14] propose leveraging the density ratio $p_t(\mathbf{x}_t)/\hat{p}_t(\mathbf{x}_t)$. By incorporating this density ratio, the score estimation can be refined using the following identity:

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) = \nabla_{\mathbf{x}_t} \log \hat{p}_t(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)/\hat{p}_t(\mathbf{x}_t). \quad (8)$$

However, the density ratio cannot be computed directly. To address this, the authors propose training a discriminator to approximate it. In practice, the

discriminator is implemented as a neural network $d_\phi(\mathbf{x}_t, t)$, which is trained to minimize the cross-entropy (CE) loss between real and generated data:

$$\mathcal{L}_{\text{CE}}^d(\phi) = \int_0^T \lambda(t) \left[\mathbb{E}_{P_t} [-\log \sigma(d_\phi(\mathbf{x}_t, t))] + \mathbb{E}_{\hat{P}_t} [-\log (1 - \sigma(d_\phi(\mathbf{x}_t, t)))] \right] dt, \quad (9)$$

where σ is the sigmoid function. If the discriminator $d_\phi(\cdot, t)$ were capable of representing any measurable function from $\mathbb{R}^d$ to $\mathbb{R}$, then the optimal discriminator could be used to compute the density ratio [21, 22, 27]:

$$r^*(\mathbf{x}_t, t) = p_t(\mathbf{x}_t) / \hat{p}_t(\mathbf{x}_t) = e^{d_{\phi^*}(\mathbf{x}_t, t)}. \quad (10)$$

However, in practice, the expressivity of the discriminator is limited, and thus the discriminator does not perfectly estimate the density ratio, and the estimated density ratio is defined as $r_\phi(\mathbf{x}_t, t) = \exp(d_\phi(\mathbf{x}_t, t))$. Using this estimation, the score refinement can be computed as $\nabla_{\mathbf{x}_t} \log r_\phi(\mathbf{x}_t, t) = \nabla_{\mathbf{x}_t} d_\phi(\mathbf{x}_t, t)$. The reverse diffusion process using the discriminator guidance defines a sequence of refined distributions $\{\tilde{P}_t\}_{t \in [0, T]}$, with densities $\tilde{p}_t$ such that:

$$\begin{cases} \tilde{p}_T(\mathbf{x}_T) = \pi(\mathbf{x}_T), \\ \nabla_{\mathbf{x}_t} \log \tilde{p}_t(\mathbf{x}_t, t) = \mathbf{s}_\theta(\mathbf{x}_t, t) + \nabla_{\mathbf{x}_t} d_\phi(\mathbf{x}_t, t) := \tilde{\mathbf{s}}_{\theta, \phi}(\mathbf{x}_t, t). \end{cases} \quad (11)$$

Similarly to the classical generative process, $\tilde{P} = \tilde{P}_0$ does not perfectly match the target distribution. The Kullback-Leibler divergence between the refined distribution and the target distribution can be computed as follows applying the same assumptions as Equation (7).

$$\mathcal{D}_{\text{KL}}(P \| \tilde{P}) = \mathcal{D}_{\text{KL}}(P_T \| Q) + \frac{1}{2} \int_0^T g(t)^2 \mathbb{E}_{P_t} \left[\|\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) - \tilde{\mathbf{s}}_{\theta, \phi}(\mathbf{x}_t, t)\|_2^2 \right] dt. \quad (12)$$

In Equation (12), we note that the dissimilarity between the target distribution and the refined distribution depends on the MSE between the difference in scores and the discriminator *gradient*. However, the model is trained to minimize CE and there is no guaranty that the CE is the optimal loss for the refinement.

4 Misalignment between the cross-entropy and the Kullback-Leibler divergence

The DG framework [14] approximates the density ratio $p_t / \hat{p}_t$ by training a discriminator with the CE and using its gradient for refinement. In this section, we formally show that minimizing cross-entropy does not necessarily improve the refined distribution. Furthermore, we establish that this issue is not limited to pathological cases but naturally arises in the common overfitting regime, where the refined distribution deteriorates.

Well-trained discriminator with poorly refined distribution: Our first result, stated in Theorem 1, shows that it is possible to construct a discriminator with an arbitrarily low CE, while the refined distribution is arbitrarily far from the target:

Theorem 1. *Let $\{\mathbf{x}(t)\}_{t \in [0, T]}$ be a diffusion process defined by Equation (1). Assume that $\nabla \log \hat{p}_t = \mathbf{s}_\theta$ and $\nabla \log \tilde{p}_t = \mathbf{s}_\theta + \nabla d_\phi$ and that the induced distribution P , $\hat{P}$, and $\tilde{P}$ satisfies the assumptions detailed in Appendix A.1. Then, for every $\varepsilon > 0$ and for every $\delta > 0$, there exists a discriminator $d : \mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}$ trained to minimize the cross-entropy such that:*

$$\mathcal{L}_{\text{CE}}^d(\phi) \leq \varepsilon \quad \text{and} \quad \mathcal{D}_{\text{KL}}(P \| \tilde{P}) \geq \delta, \quad (13)$$

where $\tilde{P}$ is the distribution induced by discriminator guidance with d .

Sketch of Proof. The detailed proof of Theorem 1 is provided in Appendix A.2. The key insight is that the CE evaluates the discriminator’s values $d_\phi(\mathbf{x}, t)$ but not its gradient $\nabla_{\mathbf{x}} d_\phi(\mathbf{x}, t)$, which affects the generation process. The main argument is that a learned discriminator d_ϕ oscillating around the optimal discriminator d^* can still achieve a low CE (Figure 2a). Specifically, the magnitude of these oscillations determines how low the CE is, while their frequency degrades the approximation of $\nabla_{\mathbf{x}} d_\phi(\mathbf{x}, t)$, leading to an increase in $\mathcal{D}_{\text{KL}}(P \| \tilde{P})$.

Theorem 1 establishes that minimizing cross-entropy does not necessarily yield a better-refined distribution. Theorem 2 further demonstrates that this issue is not limited to rare or pathological cases but naturally emerges in the overfitting regime, a common occurrence in practical settings. As the discriminator memorizes the training data, its learned function develops high-frequency oscillations:

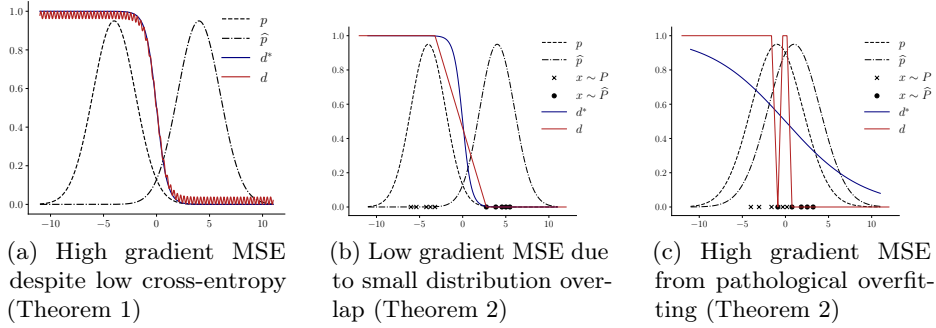


Fig. 2: Illustration of pathological cases from Theorem 1 and Theorem 2. (Left) The cross-entropy loss $\mathcal{L}_{\text{CE}}^d(\phi)$ is low, yet the MSE loss $\mathcal{L}_{\text{MSE}}^d(\phi)$ is high due to substantial gradient mismatch. (Middle) Low cross-entropy loss with low MSE, as distributions minimally overlap. (Right) Despite low cross-entropy loss, the MSE loss diverges due to significant overlap, highlighting pathological overfitting.

Theorem 2. *Let P and $\hat{P}$ be distributions on $\mathbb{R}$ with intersecting supports, admitting L -Lipschitz densities p and $\hat{p}$. Let $x_1, \dots, x_N \sim P^N$ and $x'_1, \dots, x'_N \sim \hat{P}^N$. Assume that there exists $\epsilon > 0$ such that the discriminator d_ϕ achieves logistic loss for each sample*

$$-\log \sigma(d_\phi(x_i)) \quad \text{and} \quad -\log(1 - \sigma(d_\phi(x'_i))) \leq \epsilon.$$

Then there exists a constant $c > 0$, depending asymptotically on $\log^2(\epsilon)$ as $\epsilon \rightarrow 0$, such that:

$$\lim_{N \rightarrow \infty} \frac{\mathbb{E} [\mathcal{L}_{\text{MSE}}^d(\phi)]}{N} = c, \quad \text{with} \quad c \sim (1 - \text{TV}(P, \hat{P}))^4 \log^2(\epsilon),$$

where $\text{TV}(P, \hat{P})$ denotes the total variation distance between P and $\hat{P}$. In other words, when $\text{TV}(P, \hat{P}) > 0$, as the discriminator overfits the cross-entropy loss ($\epsilon \rightarrow 0$), the mean-squared error scales as $N \log^2(\epsilon)$ and becomes arbitrarily large.

Sketch of Proof. The proof of the theorem is provided in Appendix A.4. For clarity, we focus on the one-dimensional case, which explicitly illustrates the link between the similarity of P and $\hat{P}$ and the discriminator’s behavior. However, the argument extends to higher dimensions. Overfitting leads the discriminator to assign highly different values to real and generated samples. As P and $\hat{P}$ get closer, generated samples increasingly appear near real ones, forcing the discriminator to separate them sharply. This induces high-frequency oscillations in d_ϕ , where small input changes cause large output variations, resulting in excessive gradients even where the true gradient should be smooth.

Interpretation. Theorem 2 highlights the behavior when the discriminator is *overly confident*, with logits approaching $\pm\infty$. Two distinct scenarios arise:

- When P and $\hat{P}$ differ significantly ($\text{TV}(P, \hat{P}) \approx 1$), the discriminator’s confidence is well-founded, resulting in stable MSE (Figure 2b).
- However, if the distributions P and $\hat{P}$ overlap significantly ($\text{TV}(P, \hat{P}) \ll 1$), forcing high discriminator confidence ($\epsilon \rightarrow 0$) constitutes overfitting. In this scenario, the constant c grows unbounded, causing the gradient MSE to diverge even though the training CE is minimized (Figure 2c). The situation of overlapping P and $\hat{P}$ is the common framework for refining diffusion models, as the pre-trained model ideally generates a distribution $\hat{P}$ that closely aligns with P .

5 Improved Discriminator Guidance

In this section, we introduce a discriminator loss designed to explicitly approximate the gradient of the density ratio. We analyze its theoretical properties and practical implications.

5.1 Proposed loss function

To overcome the limitations we exposed in Section 4, we propose to add a term to the loss that explicitly accounts for the gradient of the density ratio. Ideally, if we wanted the gradient of the discriminator to approximate the gradient of the true density ratio, we could optimize the following loss:

$$\mathcal{L}_{\text{SM}}^d(\phi) = \int_0^T \lambda(t) \mathbb{E}_{P_0, P_t} \left[\left\| \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) / \hat{p}_t(\mathbf{x}_t) - \nabla_{\mathbf{x}_t} d_\phi(\mathbf{x}_t, t) \right\|_2^2 \right] dt. \quad (14)$$

But, minimizing this loss suffers from the same obstacle as the score matching loss defined in Equation (3): the gradient of the true log-likelihood $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$ is unknown. Hopefully, we can use the same argument as Vincent [31] and instead use the denoising score matching loss Equation 15. Proposition 1 shows that minimizing $\mathcal{L}_{\text{SM}}^d(\phi)$ is equivalent to minimize $\mathcal{L}_{\text{MSE}}^d(\phi)$.

$$\mathcal{L}_{\text{MSE}}^d(\phi) = \int_0^T \lambda(t) \mathbb{E}_{P_0, P_t | \mathbf{x}_0} \left[\left\| \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t | \mathbf{x}_0) - \mathbf{s}_\theta(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} d_\phi(\mathbf{x}_t, t) \right\|_2^2 \right] dt. \quad (15)$$

Proposition 1. *Assume that P and $\hat{P}$ satisfy the assumptions detailed in Appendix A.1. Then, the following holds:*

$$\underset{\phi}{\operatorname{argmin}} \mathcal{L}_{\text{SM}}^d(\phi) = \underset{\phi}{\operatorname{argmin}} \mathcal{L}_{\text{MSE}}^d(\phi). \quad (16)$$

Sketch of Proof. The proof of Proposition 1 is given in Appendix A.5. It follows from the fact that the losses differ only by an additive constant independent of ϕ .

Therefore, training a discriminator with $\mathcal{L}_{\text{MSE}}^d(\phi)$ will correctly make its gradient a reliable estimate of the log-likelihood ratio.

5.2 Practical Considerations

In practice, we combine our introduced loss term with the standard cross-entropy loss to facilitate learning, and optimize the loss equation 17.

$$\mathcal{L}_{\text{train}}^d(\phi) = \mathcal{L}_{\text{MSE}}^d(\phi) + \gamma \mathcal{L}_{\text{CE}}^d(\phi), \quad (17)$$

where γ is a hyperparameter that controls the importance of the cross-entropy loss. We detail the algorithmic implementation of our method in Algorithm 1.

Optimizing $\mathcal{L}_{\text{CE}}^d(\phi)$ and $\gamma \mathcal{L}_{\text{MSE}}^d(\phi)$ Optimizing both terms yields to different considerations:

Algorithm 1 Training Discriminator with $\mathcal{L}_{\text{CE}}^d$ and $\mathcal{L}_{\text{MSE}}^d$

1. Input: Pre-trained model s_θ , Set of real data $\mathcal{X}$, Set of generated samples $\hat{\mathcal{X}}$, weighting function $\lambda(t)$, Distribution of timesteps $\mathcal{T}$, batch size b
2. Initialize discriminator parameters ϕ
3. Repeat until convergence:
 - (a) Sample a batch $\{\mathbf{x}_i\}_{i=1}^b$ from the training data $\mathcal{X}$ and $\{t\}_{i=1}^b \sim \mathcal{T}$
 - (b) Perturb the samples $\{\mathbf{x}_i\}_{i=1}^b$ to timesteps t_i to $\{\mathbf{x}_i^t\}_{i=1}^b$
 - (c) **Cross-Entropy Optimization (Baseline Loss):**
 - Sample a batch $\{\hat{\mathbf{x}}_i\}_{i=1}^b$ from $\hat{\mathcal{X}}$ and perturb to timesteps t_i : $\{\hat{\mathbf{x}}_i^t\}_{i=1}^b$
 - Compute the CE loss:

$$\widehat{\mathcal{L}_{\text{CE}}^d}(\phi) = \frac{1}{b} \sum_{i=1}^n \lambda(t_i) \left[-\log(\sigma(d_\phi(\mathbf{x}_i^t, t))) - \log(1 - \sigma(d_\phi(\hat{\mathbf{x}}_i^t, t))) \right].$$

- (d) **Gradient Matching Optimization (Proposed Loss):**
 - Compute target and pre-trained model’s scores on the perturbed training batch $\{\mathbf{x}_i^t\}_{i=1}^b$: $\nabla_{\mathbf{x}_i^t} \log p_t(\mathbf{x}_i^t | \mathbf{x}_i)$ and $s_\theta(\mathbf{x}_i^t, t)$
 - Compute discriminator gradient: $\nabla_{\mathbf{x}_i^t} d_\phi(\mathbf{x}_i^t, t)$ (e.g autodifferentiation)
 - Compute gradient-matching loss:

$$\widehat{\mathcal{L}_{\text{MSE}}^d}(\phi) = \frac{1}{b} \sum_{i=1}^b \lambda(t_i) \|\nabla_{\mathbf{x}_i^t} \log p_t(\mathbf{x}_i^t | \mathbf{x}_i) - s_\theta(\mathbf{x}_i^t, t) - \nabla_{\mathbf{x}_i^t} d_\phi(\mathbf{x}_i^t, t)\|^2$$

- (e) Update ϕ via gradient descent on $\widehat{\mathcal{L}_{\text{train}}^d}(\phi) = \widehat{\mathcal{L}_{\text{CE}}^d}(\phi) + \gamma \widehat{\mathcal{L}_{\text{MSE}}^d}(\phi)$
-

- **Optimizing $\mathcal{L}_{\text{CE}}^d(\phi)$.** In Kim et al. [14], the discriminator is trained on real and generated samples, drawn from the target distribution P and the refined distribution $\hat{P}$. Since generated samples are typically precomputed and stored, training with $\mathcal{L}_{\text{CE}}^d(\phi)$ exposes the discriminator to a fixed dataset, increasing the risk of overfitting. However, cross-entropy loss is easier to optimize than MSE loss, as it does not require computing the target distribution gradient. This results in faster training and lower memory usage.
- **Optimizing $\mathcal{L}_{\text{MSE}}^d(\phi)$.** Our proposed loss introduces dynamic perturbations to training samples, exposing the model to greater variability. While it correctly estimates the target gradient, this comes with computational overhead: each sample requires computing both $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t | \mathbf{x}_0)$ and $s_\theta(\mathbf{x}_t, t)$. Additionally, backpropagation is more complex, as gradients must be propagated through $\nabla_{\mathbf{x}_t} d_\phi(\mathbf{x}_t, t)$ rather than directly through d_ϕ . These factors lead to increased computational time and memory requirements.

6 Experiments

In this section we compare our proposed method to the baseline from Kim et al. [14]. We demonstrate the benefits of our approach on both synthetic and

real-world datasets. We will (1) observe the effect of overfitting on the quality of the refinement, (2) show the behavior of the training loss on the discriminator guidance, and (3) compare effectiveness of the proposed method for different training/generation settings and finally (4) compare the quality of the samples generated by the EDM, the EMD+DG and our method.

6.1 Visualizing the Discriminator Guidance in low-dimension

We consider two distinct Gaussian mixtures in $\mathbb{R}^2$, representing P and $\hat{P}$. Using a subVP-SDE [26], we derive the closed-form expression of the score ratio $\nabla_{\mathbf{x}} \log p(\mathbf{x})/\hat{p}(\mathbf{x})$ and employ it to refine samples from $\hat{P}$ towards P . We evaluate the performance of a discriminator trained with cross-entropy loss $\mathcal{L}_{\text{CE}}^d$ and mean squared error loss $\mathcal{L}_{\text{MSE}}^d$.

Training with $\mathcal{L}_{\text{MSE}}^d$ gives a better gradient approximation. We plot the resulting gradient norms in Figure 3 (full vector fields are depicted in Figure 2 in Appendix B). On this synthetic examples, we see that the discriminator trained with $\mathcal{L}_{\text{MSE}}^d$ achieves a much more precise gradient estimation, resulting in improved samples refinement than the discriminator trained with $\mathcal{L}_{\text{CE}}^d$. This is confirmed in Figure 1, where we compare the estimated density of the refined distribution for both methods, demonstrating that the MSE loss yields superior refinement quality.

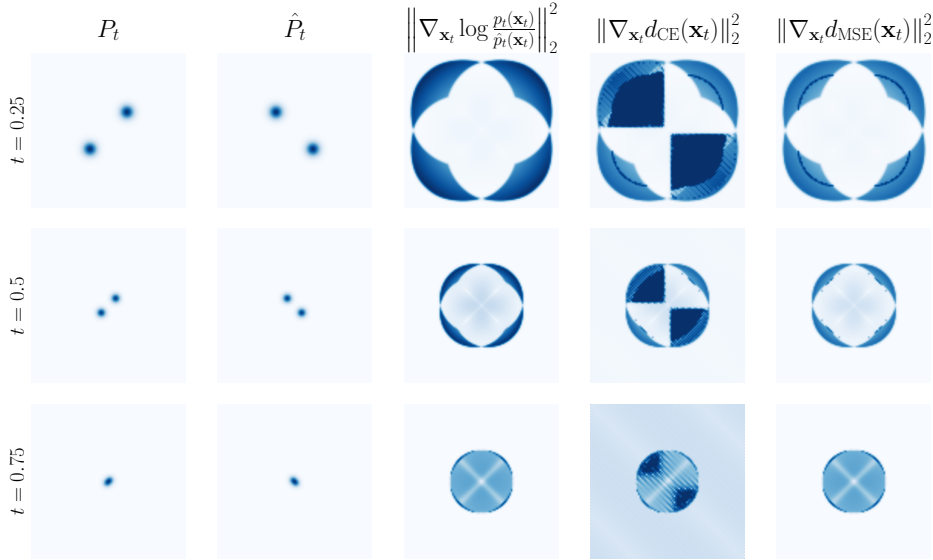


Fig. 3: Visualizing the estimation of $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)/\log \hat{p}_t(\mathbf{x}_t)$ for the discriminator trained with low CE or low MSE loss. We plot the norms for better readability.

6.2 Testing our approach the methods on real-world dataset

For high-dimensional image datasets, we implement our approach using the unconditional pre-trained EDM model [12] on CIFAR-10, FFHQ in resolution 64×64 , and AFHQv2 in resolution 64×64 . We apply the generation algorithm introduced by Kim et al. [14] to the EDM framework. To generate samples from the pre-trained model with a discriminator, they introduce a weight w to balance the contributions of the pre-trained model and the discriminator:

$$\tilde{s}_\theta(\mathbf{x}) = s_\theta(\mathbf{x}) + w \nabla_{\mathbf{x}} d_\phi(\mathbf{x}). \quad (18)$$

As a baseline, we use a discriminator trained with cross-entropy loss $\mathcal{L}_{\text{CE}}^d$, denoted as (EDM+DG), and compare it with our method, which employs $\mathcal{L}_{\text{train}}^d$ introduced in Equation (17). The generation processes are evaluated using the Fréchet Inception Distance (FID) [7], as well as Precision and Recall with $k = 3$ [18]. For FID computation, we use 50k samples for the CIFAR-10 and FFHQ datasets and 15k samples for the AFHQv2 dataset. For Precision and Recall, we set $k = 3$ and use 10k samples for each dataset.

Finally, the discriminators are parametrized following Kim et al. [14]: we adopt the ADM architecture, freezing the upper layers and training only the final layers. This results in training only 2.88M parameters, compared to the total of 50.6M, 68.3M, and 68.3M parameters for CIFAR-10, FFHQ, and AFHQv2, respectively. For comparison, the pre-trained EDM model consists of 55.7M, 61.8M, and 61.8M parameters for these datasets. Generation is conducted on $2 \times \text{GPU-H100}$, while evaluation is performed on $2 \times \text{GPU-V100}$. A complete list of hyperparameters is available at <https://github.com/AlexVerine/BoostDM>.

Observing Overfitting in the Discriminator. We analyze how the training loss of the discriminator evolves on CIFAR-10 by varying the number of training samples from 500 to 50,000. Figure 4 presents the key metrics. While Precision remains relatively stable regardless of dataset size, both Recall and the FID score change notably: Recall decreases as the number of training samples increases, while the FID score rises. This indicates that the discriminator overfits the training set,

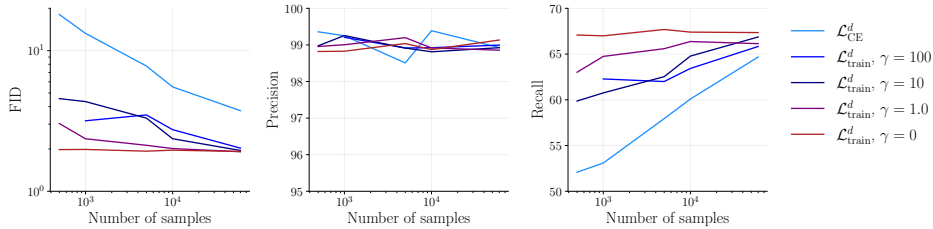


Fig. 4: Discriminator guidance using a different number of samples for the training set. Generation is performed on CIFAR-10 with $w = 1$ for all methods. FID ($\downarrow$), Precision ($\uparrow$), and Recall ($\uparrow$) are reported.

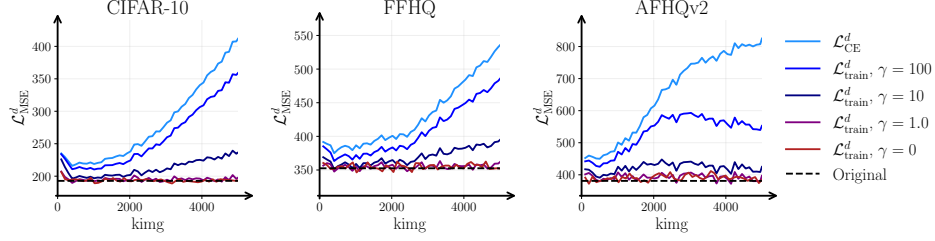


Fig. 5: Comparison of the proposed loss $\mathcal{L}_{\text{train}}^d$ and the standard loss $\mathcal{L}_{\text{CE}}^d$ for the discriminator guidance. We plot the evolution of the loss during training.

ultimately degrading the refinement quality. Additionally, our proposed method shows greater robustness to dataset size. However, when the balance between MSE and CE losses increases (i.e., for higher γ values), sensitivity to the number of training samples also increases.

Observing the effect of the regularization parameter γ . We compare how the estimated score behaves with the two different training losses. To assess how the discriminator captures the gradient of the log density ratio, we plot in Figure 5 the evolution during of $\mathcal{L}_{\text{MSE}}^d$ introduced in Equation (15). We observe that the lower γ is, the better the discriminator is at estimating the gradient of the log density ratio. This is consistent with the results of the previous section.

Comparing the effect of the weight w . Depending on the training loss, the gradient of the discriminator can have different ranges and for the refinement to be effective, the weight w must be adjusted. We evaluate the effect of w on the FID score for the EDM+DG and EDM+Ours methods on CIFAR-10, FFHQ, and AFHQv2. The results are shown in Figure 6. We observe that the optimal w is different for each dataset but that the proposed method typically leads to lower effect of the discriminator and therefore a larger w is needed. For each method and dataset, we report the FID, Precision, and Recall scores for the parameters w and γ that yield the best FID score in Table 1.

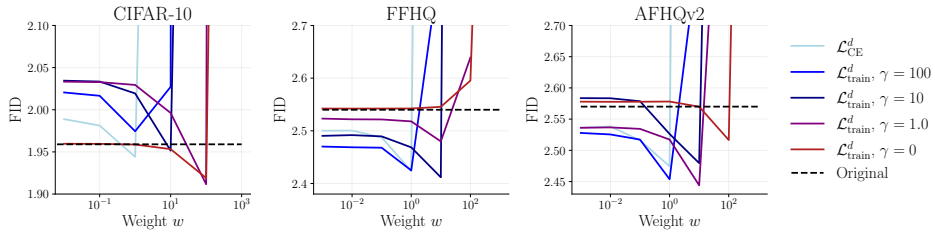


Fig. 6: FID score for different values of w on CIFAR-10, FFHQ, and AFHQv2.

Table 1: Comparison of the proposed method with optimal w on CIFAR-10, FFHQ and AFHQv2. We report the FID ($\downarrow$), Precision ($\uparrow$), and Recall ($\uparrow$) scores and the number of GPUs used for training and generation, the time for training step in seconds per 1000 images, and the memory consumption in Gi.

Dataset	Method	FID	P	R	GPU	Time	Mem. (Gi)
CIFAR-10	EDM	1.96	99.10	67.48	-	-	-
	EDM+DG	1.94	98.91	67.44	4xV100	1.18	2.24
	EDM+Ours	1.91	98.86	66.14	4xV100	4.00	9.18
FFHQ	EDM	2.54	99.69	69.69	-	-	-
	EDM+DG	2.42	99.60	69.16	4xH100	1.05	7.03
	EDM+Ours	2.41	99.56	69.09	4xH100	4.74	20.26
AFHQv2	EDM	2.57	99.99	75.64	-	-	-
	EDM+DG	2.47	99.83	74.41	4xH100	1.05	7.03
	EDM+Ours	2.44	99.96	74.58	4xH100	4.74	20.26

Comparing EDM, EDM+DG and our method. In Table 1, we compare the FID, Precision, and Recall scores for the EDM, EDM+DG, and EDM+Ours methods on CIFAR-10, FFHQ, and AFHQv2. We observe that our method consistently outperforms the EDM+DG method in terms of FID score. The Precision is not significantly affected by the method used, while the Recall can be slightly lower for our method. We can observe on Figure 7 that the samples generated by our method are closer to the samples generated by the EDM model than the EDM+DG method. On this uncurated set of samples from the AFHQv2

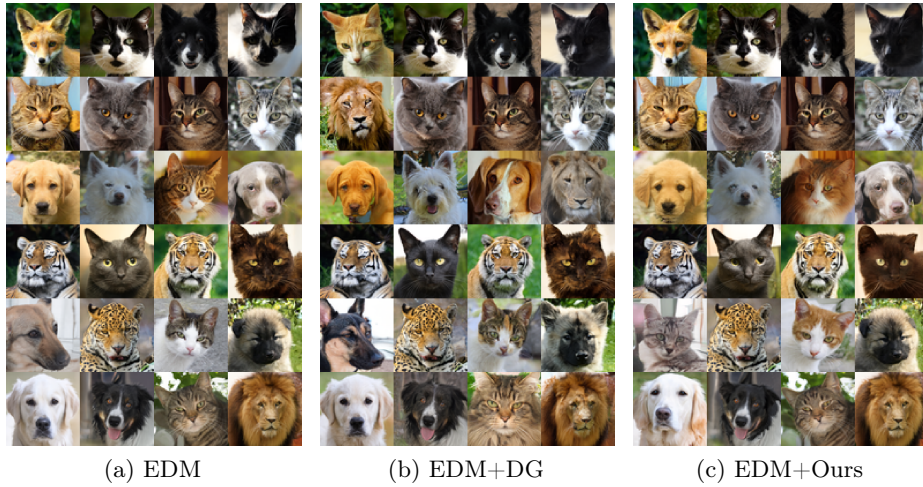


Fig. 7: Samples generated on the AFHQv2 dataset.

dataset, we can see that the samples generated by the EDM+DG method often changes the class of the sample while our method does not. We plot more samples in Appendix B.

7 Conclusion and Discussion

In this paper, we have demonstrated that discriminator guidance can be significantly improved by refining the loss function used for training the discriminator. Our theoretical analysis reveals that minimizing cross-entropy can lead to unreliable refinement, particularly in the overfitting regime, where the discriminator learns to separate real and generated samples too aggressively. To mitigate this issue, we introduced an alternative training approach that minimizes the reconstruction error, resulting in more accurate gradient estimation and improved sample quality in EDM diffusion models across diverse datasets.

Our findings highlight a fundamental challenge: while a discriminator can estimate the density ratio, accurately capturing its gradient remains difficult in practice. This issue is exacerbated by overfitting, where the discriminator’s learned function develops high-frequency oscillations that distort the refinement process. Although discriminator guidance is theoretically optimal with a perfect discriminator, real-world constraints—such as finite data, model capacity, and training instability—limit its practical effectiveness.

Acknowledgments. This work was granted access to the HPC resources of IDRIS under the allocations 2025-A0181016159, 2025-AD011014053R2 made by GENCI.

Bibliography

- [1] Anderson, B.D.O.: Reverse-time diffusion equation models. *Stochastic Processes and their Applications* **12**(3), 313–326 (May 1982), ISSN 0304-4149, [https://doi.org/10.1016/0304-4149\(82\)90051-5](https://doi.org/10.1016/0304-4149(82)90051-5)
- [2] Ansari, A.F., Ang, M.L., Soh, H.: Refining Deep Generative Models via Discriminator Gradient Flow (Jun 2021), URL <http://arxiv.org/abs/2012.00780>, arXiv:2012.00780 [cs, stat]
- [3] Azadi, S., Olsson, C., Darrell, T., Goodfellow, I., Odena, A.: Discriminator Rejection Sampling (Feb 2019), URL <http://arxiv.org/abs/1810.06758>, arXiv:1810.06758 [cs, stat]
- [4] Che, T., Zhang, R., Sohl-Dickstein, J., Larochelle, H., Paull, L., Cao, Y., Bengio, Y.: Your GAN is Secretly an Energy-based Model and You Should Use Discriminator Driven Latent Sampling. In: *Advances in Neural Information Processing Systems*, vol. 33, pp. 12275–12287, Curran Associates, Inc. (2020), URL <https://proceedings.neurips.cc/paper/2020/hash/90525e70b7842930586545c6f1c9310c-Abstract.html>
- [5] Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis (2021)
- [6] Dhariwal, P., Nichol, A.: Diffusion Models Beat GANs on Image Synthesis (Jun 2021), URL <http://arxiv.org/abs/2105.05233>, arXiv:2105.05233 [cs, stat]
- [7] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In: *Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc. (2017)
- [8] Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A., Kingma, D.P., Poole, B., Norouzi, M., Fleet, D.J., Salimans, T.: Imagen video: High definition video generation with diffusion models (2022)
- [9] Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., Fleet, D.J.: Video diffusion models (2022)
- [10] Hyvarinen, A.: Estimation of Non-Normalized Statistical Models by Score Matching (2005)
- [11] Jolicoeur-Martineau, A., Li, K., Piché-Taillefer, R., Kachman, T., Mitliagkas, I.: Gotta Go Fast When Generating Data with Score-Based Models (May 2021), URL <http://arxiv.org/abs/2105.14080>, arXiv:2105.14080 [cs, math, stat]
- [12] Karras, T., Aittala, M., Aila, T., Laine, S.: Elucidating the Design Space of Diffusion-Based Generative Models (Oct 2022), URL <http://arxiv.org/abs/2206.00364>, arXiv:2206.00364 [cs, stat]
- [13] Kelvinius, F.E., Lindsten, F.: Discriminator guidance for autoregressive diffusion models (2023)
- [14] Kim, D., Kim, Y., Kwon, S.J., Kang, W., Moon, I.C.: Refining Generative Process with Discriminator Guidance in Score-based Diffusion Models. In:

- Proceedings of the 40 th International Conference on Machine Learning., vol. 202, JMLR, Honolulu, Hawaii, USA (Apr 2023), URL <http://arxiv.org/abs/2211.17091>, arXiv:2211.17091 [cs] version: 3
- [15] Kim, D., Lai, C.H., Liao, W.H., Murata, N., Takida, Y., Uesaka, T., He, Y., Mitsufuji, Y., Ermon, S.: Consistency trajectory models: Learning probability flow ode trajectory of diffusion (2024)
 - [16] Kloeden, P.E., Platen, E., Kloeden, P.E., Platen, E.: Stochastic differential equations. Springer (1992)
 - [17] Kong, Z., Ping, W., Huang, J., Zhao, K., Catanzaro, B.: Diffwave: A versatile diffusion model for audio synthesis (2021)
 - [18] Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., Aila, T.: Improved Precision and Recall Metric for Assessing Generative Models. In: 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada. (Oct 2019), arXiv: 1904.06991
 - [19] Liu, C., Fan, W., Liu, Y., Li, J., Li, H., Liu, H., Tang, J., Li, Q.: Generative diffusion models on graphs: Methods and applications (2023)
 - [20] Lu, H., Tunanyan, H., Wang, K., Navasardyan, S., Wang, Z., Shi, H.: Specialist diffusion: Plug-and-play sample-efficient fine-tuning of text-to-image diffusion models to learn any unseen style. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 14267–14276 (June 2023)
 - [21] Nguyen, X., Wainwright, M.J., Jordan, M.I.: On surrogate loss functions and $f\beta$ -divergences. The Annals of Statistics **37**(2) (Apr 2009), ISSN 0090-5364, <https://doi.org/10.1214/08-AOS595>
 - [22] Nowozin, S., Cseke, B., Tomioka, R.: f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization (Jun 2016), URL <http://arxiv.org/abs/1606.00709>
 - [23] Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation (May 2015), <https://doi.org/10.48550/arXiv.1505.04597>, URL <http://arxiv.org/abs/1505.04597>, arXiv:1505.04597 [cs]
 - [24] Song, Y., Durkan, C., Murray, I., Ermon, S.: Maximum Likelihood Training of Score-Based Diffusion Models (Oct 2021), URL <http://arxiv.org/abs/2101.09258>, arXiv:2101.09258 [cs, stat]
 - [25] Song, Y., Garg, S., Shi, J., Ermon, S.: Sliced Score Matching: A Scalable Approach to Density and Score Estimation (Jun 2019), URL <http://arxiv.org/abs/1905.07088>, arXiv:1905.07088 [cs, stat]
 - [26] Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: SCORE-BASED GENERATIVE MODELING THROUGH STOCHASTIC DIFFERENTIAL EQUATIONS (2021)
 - [27] Sugiyama, M., Suzuki, T., Kanamori, T.: Density Ratio Estimation: A Comprehensive Review (2010)
 - [28] Tsonis, E., Tzouveli, P., Voulodimos, A.: Mitigating exposure bias in discriminator guided diffusion models (2023)
 - [29] Verine, A., Negrevergne, B., Pydi, M.S., Chevaleryre, Y.: Precision-Recall Divergence Optimization for Generative Modeling with

- GANs and Normalizing Flows. *Advances in Neural Information Processing Systems* **36**, 32539–32573 (Dec 2023), URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/67159f1c0cab15dd34c76a5dd830a389-Abstract-Conference.html
- [30] Verine, A., Pydi, M.S., Negrevergne, B., Chevaleyre, Y.: Optimal Budgeted Rejection Sampling for Generative Models. *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics* (Mar 2024), URL <http://arxiv.org/abs/2311.00460>, arXiv:2311.00460 [cs]
 - [31] Vincent, P.: A Connection Between Score Matching and Denoising Autoencoders. *Neural Computation* **23**(7), 1661–1674 (Jul 2011), ISSN 0899-7667, 1530-888X, https://doi.org/10.1162/NECO_a_00142
 - [32] Xie, E., Yao, L., Shi, H., Liu, Z., Zhou, D., Liu, Z., Li, J., Li, Z.: DiffFit: Unlocking transferability of large diffusion models via simple parameter-efficient fine-tuning (2023)
 - [33] Xu, Y., Deng, M., Cheng, X., Tian, Y., Liu, Z., Jaakkola, T.: Restart Sampling for Improving Generative Processes (Nov 2023), URL <http://arxiv.org/abs/2306.14878>, arXiv:2306.14878 [cs, stat]