VarFlow: Variational Distillation Through Score-Rule Matching of Noisy Distributions

Huiyang Shao[1,2] **Xin Xia**[2,*] **Yuxi Ren**[2] **Xing Wang**[2] **Xuefeng Xiao**[2,†]

[1]Tsinghua University [2]ByteDance Seed

Abstract

Diffusion models achieve remarkable generative performance but are hampered by slow, iterative inference. Model distillation seeks to train a fast student generator. Variational Score Distillation (VSD) offers a principled KL-divergence minimization framework for this task. This method cleverly avoids computing the teacher model's Jacobian, but its student gradient relies on the score of the student's own noisy marginal distribution, $\nabla_{\boldsymbol{x}_t} \log p_{\boldsymbol{\phi},t}(\boldsymbol{x}_t)$. VSD thus requires approximations, such as training an auxiliary network to estimate this score. These approximations can introduce biases, cause training instability, or lead to an incomplete match of the target distribution, potentially focusing on conditional means rather than broader distributional features. We introduce **VarFlow**, a novel distillation method based on a framework we term *Score-Rule Variational Distillation (SRVD)* framework. VarFlow trains a one-step generator $g_{\boldsymbol{\phi}}(\boldsymbol{z})$ by directly minimizing an energy distance (derived from the strictly proper energy score) between the student's induced noisy data distribution $p_{\boldsymbol{\phi},t}(\boldsymbol{x}_t)$ and the teacher's target noisy distribution $q_t(\boldsymbol{x}_t)$. This objective is estimated entirely using samples from these two distributions. Crucially, VarFlow bypasses the need to compute or approximate the intractable student score. By directly matching the full noisy marginal distributions, VarFlow aims for a more comprehensive and robust alignment between student and teacher, offering an efficient and theoretically grounded path to high-fidelity one-step generation.

1 Introduction

Diffusion models Sohl-Dickstein et al. [2015], Song and Ermon [2019], Ho et al. [2020], Song et al. [2021] represent a significant advance in generative modeling. They have demonstrated state-of-the-art capabilities in producing high-fidelity and diverse samples across various domains, including image synthesis Rombach et al. [2022], Ramesh et al. [2022], Ho et al. [2022], Shao et al. [2023], audio generation Liu et al. [2023], and 3D content creation Poole et al. [2022], Wang et al. [2024]. These models operate by systematically adding noise to data samples in a forward process and then learning to reverse this process, enabling generation from an initial noise distribution. However, the reverse process is iterative, often requiring tens to thousands of sequential steps. This computational burden makes inference slow and limits their practical use in scenarios demanding real-time generation or operating under resource constraints.

To mitigate this high inference cost, various distillation techniques have been developed Meng et al. [2023], Kang et al. [2024], Salimans and Ho [2022], Shao et al. [2025]. The goal of these methods is to compress the generative capabilities of a large, multi-step teacher diffusion model into a smaller, faster student model. Ideally, this student model can generate high-quality samples in significantly fewer steps, often in a single forward pass.

*Corresponding Author † Project Leader

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

 https://doi.org/10.52202/085713-1847

One prominent family of distillation methods aims to match distributions induced by the student and teacher models at different noise levels. For example, Score Distillation Sampling (SDS) Poole et al. [2022] and its variants guide a student generator $g_{\boldsymbol{\phi}}(\boldsymbol{z})$ using a pre-trained teacher diffusion model, typically represented by its noise predictor $\boldsymbol{\epsilon}_{\text{teacher}}$. A direct approach to optimize $g_{\boldsymbol{\phi}}$ might involve a denoising score matching (DSM)-like loss: generate $\boldsymbol{x}_0$ using $g_{\boldsymbol{\phi}}$, noise it to $\boldsymbol{x}_t$, and penalize differences between $\boldsymbol{\epsilon}_{\text{teacher}}(\boldsymbol{x}_t, t)$ and the actual noise $\boldsymbol{\epsilon}$. However, the gradient of such a loss with respect to the student's parameters $\boldsymbol{\phi}$ can involve the Jacobian of the teacher model, $\nabla_{\boldsymbol{x}_t}\boldsymbol{\epsilon}_{\text{teacher}}(\boldsymbol{x}_t, t)$. Computing this Jacobian is often prohibitively expensive for large teacher models and can be numerically unstable. SDS employs a stop-gradient approximation to make its gradient tractable, effectively treating $\boldsymbol{\epsilon}_{\text{teacher}}(\boldsymbol{x}_t, t) - \boldsymbol{\epsilon}$ as a gradient direction for $\boldsymbol{x}_0$. While practical, this update may not correspond to minimizing a well-defined distributional objective.

Variational Score Distillation (VSD) Wang et al. [2024] provides a more theoretically grounded approach. VSD seeks to minimize the KL divergence $D_{\mathrm{KL}}(p_{\boldsymbol{\phi},t}(\boldsymbol{x}_t) \,||\, q_t(\boldsymbol{x}_t))$ between the student's induced noisy data distribution $p_{\boldsymbol{\phi},t}(\boldsymbol{x}_t)$ and the teacher's target noisy distribution $q_t(\boldsymbol{x}_t)$ across various noise levels t. The gradient of this KL divergence with respect to the student generator's parameters $\boldsymbol{\phi}$, $\nabla_{\boldsymbol{\phi}} D_{\mathrm{KL}}(p_{\boldsymbol{\phi},t} \,||\, q_t)$, involves the difference between the student's score $\nabla_{\boldsymbol{x}_t} \log p_{\boldsymbol{\phi},t}(\boldsymbol{x}_t)$ and the teacher's score $\nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t)$ (see Equation (6)). The teacher's score is readily available from the pre-trained model $\boldsymbol{\epsilon}_{\text{teacher}}$. A key advantage of the VSD formulation is that its gradient avoids the problematic teacher Jacobian $\nabla_{\boldsymbol{x}_t}\boldsymbol{\epsilon}_{\text{teacher}}$. However, a new challenge arises: the student's score $s_{\boldsymbol{\phi},t}(\boldsymbol{x}_t) = \nabla_{\boldsymbol{x}_t} \log p_{\boldsymbol{\phi},t}(\boldsymbol{x}_t)$ is itself intractable. This intractability stems from $p_{\boldsymbol{\phi},t}(\boldsymbol{x}_t)$ being a marginal distribution, $p_{\boldsymbol{\phi},t}(\boldsymbol{x}_t) = \int q_t(\boldsymbol{x}_t | g_{\boldsymbol{\phi}}(\boldsymbol{z})) p_{\boldsymbol{z}}(\boldsymbol{z})\, \mathrm{d}\boldsymbol{z}$, obtained by integrating over the student's latent input $\boldsymbol{z}$. Consequently, VSD must approximate this student score. Common strategies include training an auxiliary score network $\boldsymbol{\epsilon}_{\text{aux}}$ (parameterized, for instance, by $\boldsymbol{\omega}$) via DSM on samples generated by $g_{\boldsymbol{\phi}}$ and then noised. Another approach involves simpler approximations, such as replacing the marginal student score $s_{\boldsymbol{\phi},t}(\boldsymbol{x}_t)$ with the score of the conditional distribution $q_t(\boldsymbol{x}_t | g_{\boldsymbol{\phi}}(\boldsymbol{z}^*))$ for a specific $\boldsymbol{z}^*$. While these approximations make VSD practical, they can introduce biases, lead to training instabilities (such as those from alternating optimization of $\boldsymbol{\phi}$ and $\boldsymbol{\omega}$ if $\boldsymbol{\epsilon}_{\text{aux}}$ is used), or result in an incomplete match of the target distribution, potentially focusing more on aligning conditional means rather than broader distributional characteristics.

To address the limitations associated with approximating the student score, we propose VarFlow, a novel distillation method based on a framework we term Score-Rule Variational Distillation. VarFlow trains a fast, single-step generator $g_{\boldsymbol{\phi}}(\boldsymbol{z})$ by directly minimizing a statistical distance between the student's induced noisy data distribution $p_{\boldsymbol{\phi},t}(\boldsymbol{x}_t)$ and the teacher's target noisy distribution $q_t(\boldsymbol{x}_t)$, as illustrated conceptually in Figure 1. The core innovation of VarFlow is its use of an objective derived from strictly proper scoring rules—specifically, the energy score, which leads to minimizing the energy distance Gneiting and Raftery [2007], Székely et al. [2004]. This principled choice allows the distillation objective to be estimated entirely using samples drawn from both the student and teacher noisy distributions. As a result, VarFlow completely bypasses the need to compute or approximate the intractable student score $\nabla_{\boldsymbol{x}_t} \log p_{\boldsymbol{\phi},t}(\boldsymbol{x}_t)$. By directly matching the full distributions $p_{\boldsymbol{\phi},t}(\boldsymbol{x}_t)$ and $q_t(\boldsymbol{x}_t)$ via an Integral Probability Metric (IPM) like the energy distance, VarFlow aims for a more comprehensive alignment of the student's behavior with the teacher's, compared to methods that concentrate on score or conditional mean matching. This approach is inspired by the successful application of scoring rules in learning complex conditional distributions, as seen in conceptual Distributional Diffusion Models (DDMs, discussed in Section 3.3 and Section D). VarFlow adapts this principle to the distillation context, presenting an efficient and principled alternative for deriving high-quality, rapid generative models from pre-trained diffusion teachers.

Our main contributions are:

1. **A Novel Distillation Method (VarFlow):** We propose VarFlow, which operationalizes a Score-Rule Variational Distillation (SRVD) framework. VarFlow trains a single-step generator by minimizing the energy distance between the noisy data distributions induced by the student and the teacher. This objective is estimated purely from samples, thereby circumventing the need for student score computation or approximation.
2. **Theoretical Grounding and Robustness:** We provide theoretical analysis establishing the consistency of the VarFlow objective. We argue that its sample-based nature and avoidance of score approximations can lead to more stable and straightforward training dynamics.

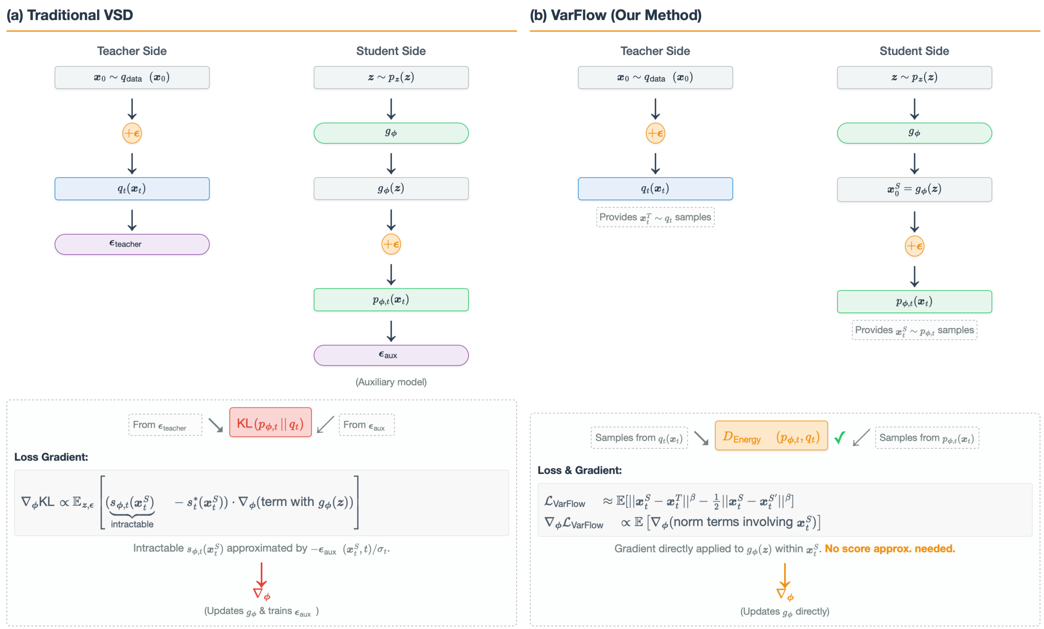

Figure 1: Conceptual comparison. (a) Variational Score Distillation (VSD) minimizes KL divergence. Its gradient requires the student's marginal score $\nabla_{\boldsymbol{x}_t} \log p_{\boldsymbol{\phi},t}(\boldsymbol{x}_t)$, which is typically intractable and approximated (e.g., via an auxiliary network $\boldsymbol{\epsilon}_{\text{aux}}$ trained to predict noise corresponding to this score). (b) VarFlow, our Score-Rule Variational Distillation (SRVD) method, directly minimizes the energy distance between the student's noisy marginal $p_{\boldsymbol{\phi},t}(\boldsymbol{x}_t)$ and the teacher's $q_t(\boldsymbol{x}_t)$. This objective is estimated using only samples from these distributions, bypassing the need for student score estimation and its associated approximations.

3. **Principled Alternative to KL-based VSD:** VarFlow offers a robust alternative to KL-divergence based distillation. By directly optimizing an IPM, it aims to capture fuller distributional characteristics, potentially leading to higher-fidelity student models.

2 Related Work

Our method, VarFlow, distills large diffusion models into one-step generators by directly matching the noisy marginal distributions of a student and teacher using an energy distance objective. This places our work at the intersection of diffusion model distillation, generative models using integral probability metrics, and generative modeling with scoring rules.

2.1 Distillation of Diffusion Models

A primary challenge in diffusion model distillation is removing the slow, iterative sampling process. Variational Score Distillation (VSD) Wang et al. [2024] offers a principled approach by minimizing the KL-divergence between the student and teacher's noisy marginals. However, VSD requires approximating the intractable student score function, $\nabla_{\boldsymbol{x}_t} \log p_{\boldsymbol{\phi},t}(\boldsymbol{x}_t)$, which can introduce bias.

Recent works have demonstrated that effective distillation is possible without direct score supervision Zhang et al. [2025]. These methods show that while the teacher's score function is not essential, initializing the student with the teacher's weights is crucial for transferring learned feature representations Zhang et al. [2025]. VarFlow shares the goal of avoiding explicit score functions. However, instead of relying on initialization, we propose directly minimizing a well-defined Integral Probability Metric (IPM) between the marginal distributions, offering a more direct path to distributional matching.

2.2 Integral Probability Metrics in Generative Models

Using statistical distances to match distributions is a well-established concept, particularly in Generative Adversarial Networks (GANs). An IPM measures the distance between two probability

https://doi.org/10.52202/085713-1847

distributions, with a prominent example being the Maximum Mean Discrepancy (MMD), which is based on a Reproducing Kernel Hilbert Space (RKHS) Gretton et al. [2012]. MMD GAN, for instance, replaces the standard discriminator with a two-sample MMD test, which can stabilize training and mitigate issues like mode collapse by training the generator to minimize the MMD between generated and real samples Li et al. [2017].

VarFlow applies this philosophy to diffusion distillation. We replace the KL-divergence objective found in VSD with an energy distance objective—which is also an IPM—thereby inheriting the benefits of direct distributional matching in a distillation context.

2.3 Scoring Rules in Diffusion Models

Our use of the energy distance is inspired by the application of scoring rules for learning *conditional* distributions in multi-step diffusion models. For example, Distributional Diffusion Models (DDM) De Bortoli et al. [2025] use proper scoring rules to learn the full conditional posterior $q(x_0|x_t)$, not just its mean. This allows for more diverse sampling at each reverse step and can accelerate inference by enabling larger step sizes De Bortoli et al. [2025].

However, the distinction is critical: DDM improves a **multi-step** sampler by enhancing each conditional step, whereas VarFlow is designed for **one-step** generation. Our method matches the **noisy marginal distributions**, $p_{\phi,t}(x_t)$ and $q_t(x_t)$, which allows the objective to be estimated purely from samples and bypasses the student score problem. While methods like DDM De Bortoli et al. [2025] and Reverse Markov Learning Shen et al. [2025] also use scoring rules, they remain multi-step frameworks focused on conditional distributions, unlike VarFlow's one-step, marginal-matching approach. Other works like Inductive Moment Matching (IMM) Sun et al. [2025] also validate distributional matching for efficient generation, a principle VarFlow adapts for distillation.

The following table summarizes the key distinctions:

Method	Learns	Distribution Matched	Inference Process
RML	Sequence of generators $\{g_t(x_t, \epsilon)\}_{t=0}^{T}$	Reverse Conditional: $p(x_{t-1}|x_t)$	**T-step** sequential
DDM	Conditional posterior generator $G_\theta(x_t, t, \xi)$	Conditional Posterior: $p(x_0|x_t)$	Multi-step (DDIM-like)
VarFlow (Ours)	**Single** one-step generator $g_\phi(z)$	Noisy Marginal: $p_{\phi,t}(x_t)$	**One-step** parallel

This distinction is crucial. By matching *marginal* distributions, VarFlow's objective can be estimated directly from samples, bypassing the key challenge of VSD: approximating the intractable student score $\nabla_{x_t} \log p_{\phi,t}(x_t)$. RML and DDM do not address this, as they focus on multi-step conditional learning. Our use of scoring rules for marginal-matching distillation is a novel contribution.

3 Preliminaries

We review key concepts: diffusion models, proper scoring rules, the idea of distributional posterior learning as a conceptual precursor, and Variational Score Distillation. A summary of notation is in Table 5 (Section B).

3.1 Diffusion Models and Score Matching

Diffusion models Ho et al. [2020], Song et al. [2021] define a forward noising process that gradually transforms a data sample $\boldsymbol{x}_0 \sim q_{\text{data}}(\boldsymbol{x}_0)$ into approximate Gaussian noise over $t \in [0, T]$. A common formulation, based on the Variance Preserving (VP) SDE Song et al. [2021], defines the conditional distribution of $\boldsymbol{x}_t$ given $\boldsymbol{x}_0$ as:

$$q_t(\boldsymbol{x}_t \mid \boldsymbol{x}_0) = \mathcal{N}\Big(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\, \boldsymbol{x}_0, \sigma_t^2\, \mathbf{I}\Big), \tag{1}$$

where $\bar{\alpha}_t$ decreases with t and $\sigma_t^2 = 1 - \bar{\alpha}_t$. Generative modeling involves learning to reverse this process, typically by estimating the score function $\nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t)$, where $q_t(\boldsymbol{x}_t) = \mathbb{E}_{\boldsymbol{x}_0}[q_t(\boldsymbol{x}_t|\boldsymbol{x}_0)]$.

In practice, a network $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ predicts the noise $\boldsymbol{\epsilon}$ from $\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sigma_t\boldsymbol{\epsilon}$, often via Denoising Score Matching (DSM):

$$\mathcal{L}_{\text{DSM}}(\boldsymbol{\theta}) = \mathbb{E}_{t,\, \boldsymbol{x}_0,\, \boldsymbol{\epsilon}}\left[w(t)\, \|\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\sqrt{\bar{\alpha}_t}\, \boldsymbol{x}_0 + \sigma_t\, \boldsymbol{\epsilon},\, t) - \boldsymbol{\epsilon}\|_2^2\right]. \tag{2}$$

The learned $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$ relates to the score $s_{\boldsymbol{\theta}}$ via $s_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) \approx -\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)/\sigma_t$. Details are in Section C.1.

3.2 Proper Scoring Rules

Scoring rules evaluate probabilistic forecasts Gneiting and Raftery [2007]. A scoring rule $\mathcal{S}(P, y)$ assigns a score when forecast P is issued and outcome y occurs. The expected score for $Y \sim Q$ is $\mathcal{S}(P, Q) = \mathbb{E}_{Y\sim Q}[\mathcal{S}(P, Y)]$.

Definition 3.1 (Strictly Proper Scoring Rule)**.** A scoring rule $\mathcal{S}$ is *strictly proper* if $\mathcal{S}(Q, Q) \geq \mathcal{S}(P, Q)$ for all P, Q, with equality iff $P = Q$ (a.e.).

Strict propriety ensures forecasters report true belief. Minimizing $-\mathcal{S}(P_{\boldsymbol{\lambda}}, Q)$ drives $P_{\boldsymbol{\lambda}}$ towards Q.

Energy Score and Energy Distance. A key example is the *energy score* Gneiting and Raftery [2007], Székely et al. [2004]. For $\beta \in (0, 2)$:

$$\mathcal{S}_{\text{Energy}}^{(\beta)}(P, y) = \frac{1}{2}\mathbb{E}_{X,X' \overset{\text{iid}}{\sim} P}[\|X - X'\|^\beta] - \mathbb{E}_{X\sim P}[\|X - y\|^\beta]. \tag{3}$$

This rule is strictly proper for distributions P with finite β-th moments. Optimizing a model P to match a target Q by minimizing the negative expected energy score, $-\mathbb{E}_{Y\sim Q}[\mathcal{S}_{\text{Energy}}^{(\beta)}(P, Y)]$, is equivalent (up to terms constant in P) to minimizing the *energy distance* $D_{\text{Energy}}^{(\beta)}(P, Q)^2$. The energy distance, detailed in Section C.3 along with the explicit form of the negative expected score (Equation (17)), is an Integral Probability Metric (IPM) defined as:

$$D_{\text{Energy}}^{(\beta)}(P, Q)^2 = 2\mathbb{E}_{X\sim P, Y\sim Q}\|X - Y\|^\beta - \mathbb{E}_{X,X' \overset{\text{iid}}{\sim} P}\|X - X'\|^\beta - \mathbb{E}_{Y,Y' \overset{\text{iid}}{\sim} Q}\|Y - Y'\|^\beta. \tag{4}$$

It defines a metric on distributions with finite β-moments.

3.3 Conceptual Inspiration: Distributional Learning for Posterior Estimation (DDM)

Standard diffusion models often predict the conditional mean $\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t]$. However, the true posterior $q(\boldsymbol{x}_0|\boldsymbol{x}_t)$ can be richer. Distributional Diffusion Models (DDMs, see Section D) aim to learn this entire posterior $p_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x}_t, t)$ using a conditional generator $G_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, \boldsymbol{\xi})$, where $\boldsymbol{\xi}$ is auxiliary noise. The distribution of $G_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, \boldsymbol{\xi})$ over $\boldsymbol{\xi}$ defines $p_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x}_t, t)$. Strictly proper scoring rules, like the energy score, can train $G_{\boldsymbol{\theta}}$ by minimizing an expected negative score $-\mathbb{E}_{(\boldsymbol{x}_0,\boldsymbol{x}_t)}[\mathcal{S}(p_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x}_t, t)(\cdot|\boldsymbol{x}_t, t), \boldsymbol{x}_0)]$. Crucially, the gradient of this objective with respect to $\boldsymbol{\theta}$ does *not* require computing the score $\nabla_{\hat{\boldsymbol{x}}_0} \log p_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x}_t, t)(\hat{\boldsymbol{x}}_0|\boldsymbol{x}_t, t)$. This score-free distributional matching is a key inspiration for VarFlow.

3.4 Variational Score Distillation (VSD)

VSD Wang et al. [2024] distills a teacher diffusion model into a fast generator $g_{\boldsymbol{\phi}}(\boldsymbol{z})$, where $\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})$. The student $g_{\boldsymbol{\phi}}$ induces a noisy distribution $p_{\boldsymbol{\phi},t}(\boldsymbol{x}_t) = \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}g_{\boldsymbol{\phi}}(\boldsymbol{z}), \sigma_t^2\mathbf{I})]$. VSD trains $g_{\boldsymbol{\phi}}$ by minimizing $D_{\text{KL}}(p_{\boldsymbol{\phi},t}(\boldsymbol{x}_t)\,||\,q_t(\boldsymbol{x}_t))$ averaged over time t:

$$\mathcal{L}_{\text{VSD-KL}}(\boldsymbol{\phi}) = \mathbb{E}_{t\sim \text{Unif}[0,T]}[\tilde{w}(t)D_{\text{KL}}(p_{\boldsymbol{\phi},t}(\boldsymbol{x}_t)\,||\,q_t(\boldsymbol{x}_t))]. \tag{5}$$

The gradient (see Section C.2 or Wang et al. [2024]) is:

$$\nabla_{\boldsymbol{\phi}}\mathcal{L}_{\text{VSD-KL}} = \mathbb{E}_{t,\boldsymbol{z},\boldsymbol{\epsilon}}\left[\tilde{w}(t)\left(s_{\boldsymbol{\phi},t}(\boldsymbol{x}_t) - s_t^*(\boldsymbol{x}_t)\right)\cdot \nabla_{\boldsymbol{\phi}}\left(\sqrt{\bar{\alpha}_t}g_{\boldsymbol{\phi}}(\boldsymbol{z})\right)\right], \tag{6}$$

where $\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}g_{\boldsymbol{\phi}}(\boldsymbol{z}) + \sigma_t\boldsymbol{\epsilon}$, and $s_t^*(\boldsymbol{x}_t) \approx -\epsilon_{\text{teacher}}(\boldsymbol{x}_t, t)/\sigma_t$. The challenge is the intractable student score $s_{\boldsymbol{\phi},t}(\boldsymbol{x}_t) = \nabla_{\boldsymbol{x}_t}\log p_{\boldsymbol{\phi},t}(\boldsymbol{x}_t)$. VSD approximates it, e.g., by training an auxiliary network $\epsilon_{\text{aux}}(\boldsymbol{x}_t, t)$ or using simpler conditional score approximations. These can be sources of issues, which VarFlow avoids.

https://doi.org/10.52202/085713-1847

4 VarFlow: Score-Rule Variational Distillation

We now introduce VarFlow, our method for distilling a pre-trained teacher diffusion model into a fast, single-step generator $g_{\phi}(\boldsymbol{z})$. VarFlow operationalizes a Score-Rule Variational Distillation (SRVD) framework. It leverages proper scoring rules to directly minimize a statistical distance between the teacher's and student's noisy data distributions, crucially avoiding the need for student score function estimation or approximation.

4.1 Motivation and Setup

The objective is to train $g_{\phi}(\boldsymbol{z})$, where $\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})$, such that its output $\boldsymbol{x}_0^S = g_{\phi}(\boldsymbol{z})$ resembles data from $q_{\text{data}}(\boldsymbol{x}_0)$. This student induces a noisy distribution $p_{\phi,t}(\boldsymbol{x}_t)$:

$$p_{\phi,t}(\boldsymbol{x}_t|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t} g_{\phi}(\boldsymbol{z}), \sigma_t^2 \mathbf{I}), \quad \text{so} \quad p_{\phi,t}(\boldsymbol{x}_t) = \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[p_{\phi,t}(\boldsymbol{x}_t|\boldsymbol{z})]. \tag{7}$$

The teacher implies a target noisy distribution $q_t(\boldsymbol{x}_t)$ from real data $\boldsymbol{x}_0 \sim q_{\text{data}}(\boldsymbol{x}_0)$:

$$q_t(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, \sigma_t^2 \mathbf{I}), \quad \text{so} \quad q_t(\boldsymbol{x}_t) = \mathbb{E}_{\boldsymbol{x}_0\sim q_{\text{data}}(\boldsymbol{x}_0)}[q_t(\boldsymbol{x}_t|\boldsymbol{x}_0)]. \tag{8}$$

VarFlow proposes minimizing the energy distance between $p_{\phi,t}(\boldsymbol{x}_t)$ and $q_t(\boldsymbol{x}_t)$, estimated using only samples.

4.2 VarFlow Objective using Energy Score

VarFlow trains g_{ϕ} by minimizing the energy distance (Equation (4)) between $p_{\phi,t}(\boldsymbol{x}_t)$ and $q_t(\boldsymbol{x}_t)$, averaged over t. The *VarFlow Energy Distillation Loss* uses the negative expected energy score (Section C.3, Equation (17)). Minimizing this is equivalent to minimizing energy distance:

$$\mathcal{L}_{\text{VarFlow}}^{(\beta)}(\phi) = \mathbb{E}_{t\sim\text{Unif}[0,T]}\left[\tilde{w}(t)\left(\mathbb{E}_{\substack{\boldsymbol{x}_t^S\sim p_{\phi,t}\\ \boldsymbol{x}_t^T\sim q_t}}\left[\|\boldsymbol{x}_t^S - \boldsymbol{x}_t^T\|^{\beta}\right] - \frac{1}{2}\mathbb{E}_{\substack{\boldsymbol{x}_t^S\sim p_{\phi,t}\\ \boldsymbol{x}_t^{S'}\sim p_{\phi,t}}}\left[\|\boldsymbol{x}_t^S - \boldsymbol{x}_t^{S'}\|^{\beta}\right]\right)\right].$$

Here, $\beta \in (0,2)$ and $\tilde{w}(t)$ is a time weighting. The term $-\frac{1}{2}\mathbb{E}_{\boldsymbol{x}_t^T,\boldsymbol{x}_t^{T'}\sim q_t}[\|\boldsymbol{x}_t^T - \boldsymbol{x}_t^{T'}\|^{\beta}]$ from full energy distance (Equation (4)) is omitted as it's constant w.r.t. ϕ.

Monte Carlo Estimation. For a sampled t: 1. Sample $\{\boldsymbol{z}^{(k)}\}_{k=1}^K \overset{\text{iid}}{\sim} p_{\boldsymbol{z}}$; $\boldsymbol{x}_0^{S,(k)} = g_{\phi}(\boldsymbol{z}^{(k)})$. 2. Sample $\{\boldsymbol{\epsilon}^{S,(k)}\}_{k=1}^K \overset{\text{iid}}{\sim} \mathcal{N}(\mathbf{0},\mathbf{I})$; $\boldsymbol{x}_t^{S,(k)} = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0^{S,(k)} + \sigma_t\boldsymbol{\epsilon}^{S,(k)}$. 3. Sample $\{\boldsymbol{x}_0^{T,(k)}\}_{k=1}^K \overset{\text{iid}}{\sim} q_{\text{data}}$. 4. Sample $\{\boldsymbol{\epsilon}^{T,(k)}\}_{k=1}^K \overset{\text{iid}}{\sim} \mathcal{N}(\mathbf{0},\mathbf{I})$; $\boldsymbol{x}_t^{T,(k)} = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0^{T,(k)} + \sigma_t\boldsymbol{\epsilon}^{T,(k)}$. An unbiased U-statistic estimate for the bracketed term in Equation (8) (for $K \geq 2$):

$$\hat{L}_t(\phi) = \tilde{w}(t)\left(\frac{1}{K^2}\sum_{i=1}^K\sum_{j=1}^K \|\boldsymbol{x}_t^{S,(i)} - \boldsymbol{x}_t^{T,(j)}\|^{\beta} - \frac{1}{K(K-1)}\sum_{i=1}^K\sum_{\substack{j=1\\ j\neq i}}^K \frac{1}{2}\|\boldsymbol{x}_t^{S,(i)} - \boldsymbol{x}_t^{S,(j)}\|^{\beta}\right). \tag{9}$$

A simpler (paired) estimator for the cross-term, which is often used but may have higher variance:

$$\hat{L}_t^{\text{paired}}(\phi) = \tilde{w}(t)\left(\frac{1}{K}\sum_{k=1}^K \|\boldsymbol{x}_t^{S,(k)} - \boldsymbol{x}_t^{T,(k)}\|^{\beta} - \frac{1}{K(K-1)}\sum_{k=1}^K\sum_{\substack{j=1\\ j\neq k}}^K \frac{1}{2}\|\boldsymbol{x}_t^{S,(k)} - \boldsymbol{x}_t^{S,(j)}\|^{\beta}\right). \tag{10}$$

The gradient $\nabla_{\phi}\hat{L}_t(\phi)$ (or $\nabla_{\phi}\hat{L}_t^{\text{paired}}(\phi)$) is computed via backpropagation through g_{ϕ} as it appears in $\boldsymbol{x}_t^{S,(i)}$ and $\boldsymbol{x}_t^{S,(j)}$.

4.3 Advantages and Connections

The VarFlow method, through its SRVD framework, offers several key advantages:

- **No Explicit Student Score Computation:** A significant advantage is that VarFlow avoids the computational and approximation challenges associated with the student score in VSD. Traditional VSD relies on the gradient (Equation (6)), which includes the term $s_{\boldsymbol{\phi},t}(\boldsymbol{x}_t) = \nabla_{\boldsymbol{x}_t} \log p_{\boldsymbol{\phi},t}(\boldsymbol{x}_t)$. This student score is the gradient of the log-density of the student's noisy marginal distribution $p_{\boldsymbol{\phi},t}(\boldsymbol{x}_t) = \int p_{\boldsymbol{\phi},t}(\boldsymbol{x}_t|g_{\boldsymbol{\phi}}(\boldsymbol{z}))p_{\boldsymbol{z}}(\boldsymbol{z})\,\mathrm{d}\boldsymbol{z}$.
- **Direct Distribution Matching via IPMs:** VarFlow minimizes a well-defined statistical distance (the energy distance) between the full distributions $p_{\boldsymbol{\phi},t}(\boldsymbol{x}_t)$ and $q_t(\boldsymbol{x}_t)$. This promotes a comprehensive alignment of their characteristics.
- **Simplicity and Potential for Enhanced Stability:** Avoiding explicit score terms and their approximations can lead to simpler implementation and potentially more stable training.
- **Flexibility with Teacher Information:** VarFlow uses samples from $q_t(\boldsymbol{x}_t)$ (noised real data) and does not require explicit access to the teacher's score $\boldsymbol{\epsilon}_{\text{teacher}}$ during $g_{\boldsymbol{\phi}}$ optimization.

Theorem 4.1 (Consistency of VarFlow Objective)**.** *Assume $\tilde{w}(t) > 0$ a.e. on $[0, T]$ and t is sampled with full support. Let $p_{\boldsymbol{\phi},t}$ be the student's noisy marginal and q_t be the teacher's. If using energy score with $\beta \in (0, 2)$ in $\mathcal{L}^{(\beta)}_{VarFlow}(\boldsymbol{\phi})$ (Equation (8)), the loss is minimized iff $p_{\boldsymbol{\phi},t}(\boldsymbol{x}_t) = q_t(\boldsymbol{x}_t)$ for a.e. $t \in [0, T]$ (assuming finite β-moments and sufficient capacity for $g_{\boldsymbol{\phi}}$).*

Proof. A sketch is provided here, full details in Section E.1. The VarFlow loss is a weighted integral of terms $-\tilde{w}(t)\mathcal{S}^{(\beta)}_{\text{Energy}}(p_{\boldsymbol{\phi},t}, q_t)$. Minimizing this w.r.t. $p_{\boldsymbol{\phi},t}$ is equivalent to minimizing $\frac{1}{2}\tilde{w}(t)D^{(\beta)}_{\text{Energy}}(p_{\boldsymbol{\phi},t}, q_t)^2$ (plus constant terms). Since $D^{(\beta)}_{\text{Energy}}$ is a metric for $\beta \in (0, 2)$, it's non-negative and zero iff distributions match. Given $\tilde{w}(t) > 0$, overall loss is minimized iff $D^{(\beta)}_{\text{Energy}}(p_{\boldsymbol{\phi},t}, q_t)^2 = 0$ for a.e. t, implying $p_{\boldsymbol{\phi},t}(\boldsymbol{x}_t) = q_t(\boldsymbol{x}_t)$ for a.e. t. □

5 The VarFlow Algorithm

The VarFlow student generator $g_{\boldsymbol{\phi}}$ is trained by minimizing the VarFlow Energy Distillation Loss $\mathcal{L}^{(\beta)}_{\text{VarFlow}}(\boldsymbol{\phi})$ (defined in Equation (8)). This objective aims to directly match the student's induced noisy data distribution $p_{\boldsymbol{\phi},t}(\boldsymbol{x}_t)$ with the teacher's target noisy distribution $q_t(\boldsymbol{x}_t)$ using a sample-based energy distance. The training procedure is outlined in Algorithm 1.

Algorithm 1 VarFlow Training Procedure

Require: Student generator $g_{\boldsymbol{\phi}}$, data source $q_{\text{data}}(\boldsymbol{x}_0)$, noise schedule $(\bar{\alpha}_t, \sigma_t)$, energy score exponent $\beta \in (0, 2)$, batch size $K \geq 2$, latent distribution $p_{\boldsymbol{z}}(\boldsymbol{z})$, time weighting $\tilde{w}(t)$, learning rate η.

```
Initialize parameters φ for the student generator g_φ.
repeat
    Sample a time t ∼ Unif[0, T] (or other suitable distribution).
    Sample K latent vectors {z^(k)}_{k=1}^K ~iid p_z(z), K real data points {x_0^{T,(k)}}_{k=1}^K ~iid q_data(x_0).
    Generate clean student samples: x_0^{S,(k)} ← g_φ(z^(k)).
    Obtain noisy student samples x_t^{S,(k)} by applying the forward process (Equation (1)) to x_0^{S,(k)}.
    Obtain noisy teacher samples x_t^{T,(k)} by applying the forward process (Equation (1)) to x_0^{T,(k)}.
    Estimate the batch loss L̂_t(φ) using {x_t^{S,(k)}} and {x_t^{T,(k)}} per Equation (8).
    Update student parameters: φ ← φ − η∇_φ L̂_t(φ).
until convergence
return Trained student generator g_φ.
```

The VarFlow algorithm iteratively refines the student generator $g_{\boldsymbol{\phi}}$. In each training step, it draws batches of noisy samples, $\boldsymbol{x}^S_t$ (derived from $g_{\boldsymbol{\phi}}(\boldsymbol{z})$) and $\boldsymbol{x}^T_t$ (derived from real data $\boldsymbol{x}_0$), corresponding to a specific noise level t. The core idea is to update $g_{\boldsymbol{\phi}}$ by minimizing the empirical energy distance between these two sets of noisy samples, as quantified by $\mathcal{L}^{(\beta)}_{\text{VarFlow}}(\boldsymbol{\phi})$ (see Equations (8) to (10)). This procedure directly encourages the student's induced noisy marginal distribution $p_{\boldsymbol{\phi},t}(\boldsymbol{x}_t)$ to match the teacher's $q_t(\boldsymbol{x}_t)$ across all relevant t. The exponent β (typically 1 or 2) in the energy distance is a hyperparameter. A key advantage of this approach is its simplicity and directness: training relies

https://doi.org/10.52202/085713-1847

entirely on samples and does not require computing or approximating the potentially intractable score of the student's noisy distribution, $\nabla_{\boldsymbol{x}_t} \log p_{\boldsymbol{\phi},t}(\boldsymbol{x}_t)$. Upon convergence, the trained $g_{\boldsymbol{\phi}}$ can generate samples $\hat{\boldsymbol{x}}_0 = g_{\boldsymbol{\phi}}(\boldsymbol{z})$ from latent codes $\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})$ in a single forward pass.

6 Experiments

In this section, we empirically validate the effectiveness of VarFlow. We conduct a comprehensive set of experiments across various standard benchmarks and diverse task settings. Refer to Section A for experiment setup and details.

Table 1: Image generation results on ImageNet 64x64 (class-conditional), ImageNet 256x256 (class-conditional), CIFAR-10 32x32 (unconditional), and MS COCO 512x512 (text-to-image, zero-shot). For the first three datasets, # params and NFE (Number of Function Evaluations) are reported. For MS COCO 512x512, its # params and NFE are identical to those for ImageNet 256x256 and are therefore omitted for brevity. For each column, the best result is highlighted in **bold** and the second best is underlined.

	ImageNet 64x64			ImageNet 256x256			CIFAR-10 32x32			MS COCO 512x512	
Method	# params	NFE	FID↓	# params	NFE	FID↓	# params	NFE	FID↓	FID↓	CLIP score↑
Training from scratch: Diffusion models											
DDPM [Ho et al., 2020]	-	-	-	572M	1000	12.50	56M	1000	3.17	12.80	28.5
ADM [Dhariwal and Nichol, 2021]	296M	250	2.07	550M	250	3.87	-	-	-	10.50	29.0
EDM [Karras et al., 2024]	296M	512	1.36	550M	60	2.70	56M	35	1.97	7.50	29.5
Training from scratch: One-step models											
CT [Song et al., 2023]	296M	1	13.0	296M	1	22.0	56M	1	8.70	15.00	28.0
iCT [Song and Dhariwal, 2023]	296M	1	4.02	296M	1	7.50	56M	1	2.83	12.50	28.5
iCT-deep [Song and Dhariwal, 2023]	592M	1	3.25	592M	1	6.80	112M	1	2.51	11.00	29.0
ECT [Geng et al., 2024]	280M	1	5.51	280M	1	9.50	56M	1	3.60	12.00	28.8
SMT Jayashankar et al. [2025]	296M	1	<u>3.23</u>	296M	1	<u>6.50</u>	56M	1	3.13	<u>9.50</u>	30.0
VarFlow (ours)	296M	1	**3.19**	296M	1	**6.44**	56M	1	**2.56**	**9.26**	**30.2**
Diffusion distillation											
PD [Salimans and Ho, 2022]	296M	1	10.7	296M	1	19.0	60M	1	9.12	14.00	28.5
TRACT [Berthelot et al., 2023]	296M	1	7.43	296M	1	13.5	56M	1	3.78	11.50	29.0
CD (LPIPS) [Song et al., 2023]	296M	1	6.20	296M	1	11.0	56M	1	4.53	10.80	29.2
Diff-Instruct [Luo et al., 2023b]	296M	1	5.57	296M	1	9.80	56M	1	4.53	10.00	29.5
MultiStep-CD [Heek et al., 2024]	1200M	1	3.20	1200M	1	4.50	-	-	-	8.50	30.5
DMD w/o reg [Yin et al., 2024c]	296M	1	5.60	296M	1	9.90	56M	1	5.58	11.49	29.0
DMD2 w/ GAN [Yin et al., 2024a]	296M	1	1.51	296M	1	3.10	56M	1	2.43	8.17	28.7
MMD [Salimans et al., 2024]	400M	1	3.00	400M	1	4.20	-	-	-	8.80	30.0
SiD [Zhou et al., 2024]	296M	1	1.52	296M	1	3.15	56M	1	**1.92**	7.90	<u>31.0</u>
SiM [Luo et al., 2024]	-	-	-	-	-	-	56M	1	<u>2.02</u>	-	-
SMD Jayashankar et al. [2025]	296M	1	<u>1.48</u>	296M	1	<u>2.45</u>	56M	1	2.08	<u>7.42</u>	29.2
VarFlow (ours)	296M	1	**1.46**	296M	1	**2.42**	56M	1	2.22	**7.40**	**31.2**

6.1 Main Results and Comparisons

We present quantitative results in Table 1 and Table 2, comparing VarFlow with existing methods across diverse datasets, model architectures, and different settings.

Performance on General Benchmarks. Table 1 summarizes results on different datasets. The table is divided into models trained from scratch and models obtained via diffusion distillation.

- *Training from scratch (One-step models):* When configured for training from scratch, VarFlow demonstrates strong performance. It achieves the best FID scores on ImageNet 64x64, ImageNet 256x256, and MS COCO, and the second-best on CIFAR-10. Its CLIP score on MS COCO is also SOTA in this category. This highlights VarFlow's capability as a efficient generative modeling framework.
- *Diffusion distillation:* In the distillation setting (where $g_{\boldsymbol{\phi}}$ is typically initialized from a pre-trained teacher and fine-tuned, or its architecture is based on a teacher), VarFlow consistently achieves SOTA or highly competitive results. It obtains the best metrics for most cases. The CLIP score on MS COCO is also the best among distillation methods. These results underscore VarFlow's effectiveness in compressing powerful teacher models into fast one-step generators while preserving high generation quality.

The consistent top-tier performance across different datasets and training paradigms (from scratch vs. distillation) for one-step generation (NFE=1) demonstrates the robustness and efficacy of the VarFlow approach.

Method		1-Step		2-Step		4-Step		8-Step	
		FID ↓	CLIP ↑	FID ↓	CLIP ↑	FID ↓	CLIP ↑	FID ↓	CLIP ↑
Stable Diffusion V1.5 Comparison									
SD15-Base Rombach et al. [2022]	UNet	19.9±0.04	27.4±0.03	12.3±0.05	28.1±0.05	11.4±0.03	28.9±0.04	10.8±0.04	29.2±0.03
SD15-PeRFlow [Yan et al., 2024]	LoRA	5.42±0.05	30.2±0.04	5.38±0.04	30.5±0.08	5.25±0.07	30.1±0.06	5.18±0.06	30.3±0.05
SD15-LCM [Luo et al., 2023a]	LoRA	5.31±0.06	30.3±0.05	5.08±0.08	30.6±0.06	5.19±0.05	30.7±0.04	5.10±0.05	30.9±0.04
SD15-TCD [Zheng et al., 2024]	LoRA	5.52±0.08	29.9±0.06	5.12±0.05	30.4±0.07	5.26±0.08	30.2±0.04	5.20±0.07	30.4±0.05
Hyper-SD15 [Ren et al., 2024]	LoRA	5.38±0.05	30.1±0.04	5.09±0.07	30.6±0.05	5.18±0.06	30.3±0.06	5.12±0.05	30.5±0.05
SD15-VarFlow	LoRA	**5.07**±0.05	**30.8**±0.04	**4.81**±0.05	**31.2**±0.03	**4.73**±0.06	**31.4**±0.02	**4.62**±0.05	**31.7**±0.04
Stable Diffusion XL Comparison									
SDXL-Base Podell et al. [2023]	UNet	15.8±0.04	28.2±0.03	10.6±0.03	28.6±0.04	9.48±0.02	28.9±0.04	8.95±0.03	29.1±0.03
SDXL-Turbo Sauer et al. [2023]	UNet	4.35±0.06	30.4±0.07	4.19±0.04	30.6±0.05	4.05±0.05	30.8±0.08	3.98±0.04	31.0±0.06
SDXL-PeRFlow [Yan et al., 2024]	UNet	4.23±0.06	30.2±0.04	4.25±0.06	30.1±0.07	4.08±0.07	30.6±0.05	4.01±0.06	30.8±0.05
SDXL-LCM [Luo et al., 2023a]	LoRA	4.29±0.08	30.0±0.04	4.24±0.05	29.8±0.07	4.15±0.06	30.8±0.05	4.07±0.05	31.0±0.04
SDXL-TCD [Zheng et al., 2024]	LoRA	4.53±0.07	29.9±0.05	4.13±0.04	29.7±0.09	4.23±0.06	30.7±0.06	4.16±0.05	30.9±0.05
SDXL-Lightning [Lin et al., 2024]	LoRA	4.38±0.05	30.4±0.05	4.21±0.08	29.8±0.06	4.11±0.04	30.8±0.08	4.04±0.04	31.0±0.07
Hyper-SDXL [Ren et al., 2024]	LoRA	4.24±0.06	30.0±0.04	4.19±0.05	29.8±0.07	4.14±0.06	30.8±0.05	4.08±0.05	31.0±0.04
SDXL-DMD2 [Yin et al., 2024a]	LoRA	4.22±0.03	30.8±0.04	4.09±0.04	30.9±0.05	3.95±0.03	31.5±0.03	3.88±0.03	31.7±0.03
SDXL-VarFlow	LoRA	**4.11**±0.05	**31.2**±0.03	**3.90**±0.04	**31.4**±0.02	**3.75**±0.04	**31.9**±0.03	**3.65**±0.03	**32.1**±0.01
SD3.5-Medium Comparison									
SD3.5-EMD [Xie et al., 2024]	LoRA	4.08±0.02	30.6±0.01	4.01±0.03	30.9±0.02	3.96±0.01	31.1±0.03	3.90±0.02	31.3±0.02
SD3.5-VarFlow	LoRA	**3.89**±0.03	**31.0**±0.01	**3.83**±0.02	**31.2**±0.02	**3.77**±0.03	**31.5**±0.01	**3.70**±0.03	**31.8**±0.01

Table 2: Quantitative comparison of state-of-the-art models across various architectures and steps for FID and CLIP scores on the COCO-10k dataset.

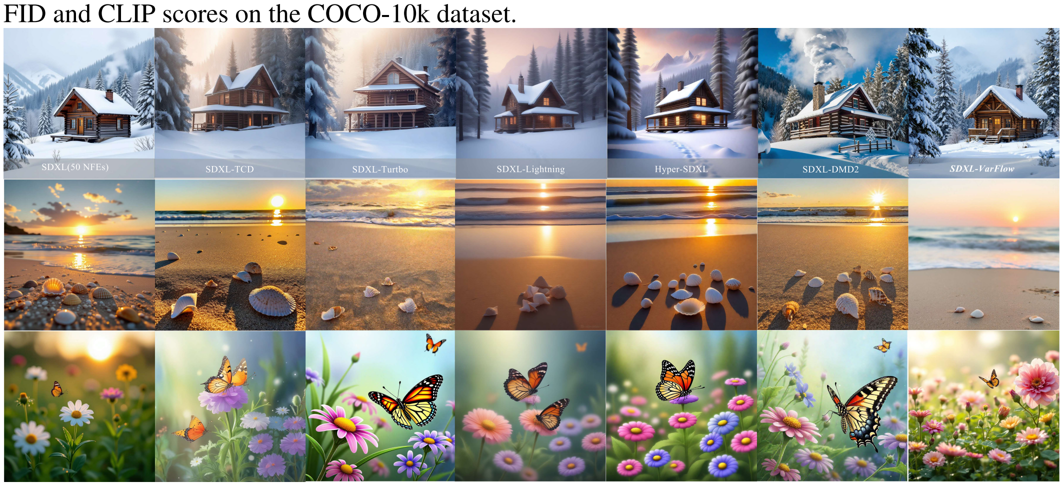

Figure 2: Qualitative comparison of VarFlow against other few-step text-to-image models like SDXL-Turbo, SDXL-TCD, SDXL-Lightning, Hyper-SDXL, and DMD2, alongside the full SDXL (NFE=50) teacher. Prompts cover diverse scenes. Please zoom in to compare details, lighting, and aesthetic quality. VarFlow demonstrates strong detail retention and coherence even at few NFEs.

Performance on Large-Scale Text-to-Image Models. Table 2 presents a detailed comparison of VarFlow (using LoRA for fine-tuning) against other leading few-step generation methods on the COCO-10k dataset, for various base models: SD1.5, SDXL, SD3.5 DiT, and SD3.5-Medium. The evaluation spans multiple NFE from 1-step to 8-steps. VarFlow consistently achieves the best FID and CLIP scores across all tested base model architectures (U-Net and DiT) and for all NFE settings (1, 2, 4, and 8 steps). These results highlight VarFlow's adaptability and superior performance in distilling large-scale, state-of-the-art T2I diffusion models into highly efficient few-step (and particularly one-step) generators using parameter-efficient LoRA fine-tuning. The consistent gains across different model families (SD U-Nets, SD3 DiTs) suggest that the VarFlow distillation principle is broadly applicable and effective.

Qualitative Comparison with Baselines. Figure 2 provides visual comparisons between images generated by SDXL-VarFlow (using 4 NFEs by default) and other leading few-step methods, as well as the original SDXL teacher (50 NFEs). Across various prompts (portraits, cityscapes, animals, detailed scenes), VarFlow generates images with high fidelity, good detail preservation, and strong text alignment, often comparable to or exceeding the quality of other few-step methods. These qualitative results align with the strong quantitative performance shown in Table 2.

https://doi.org/10.52202/085713-1847

6.2 Ablation Studies

Ablation studies are primarily performed by distilling SD1.5 using LoRA and evaluating on the MS COCO 10k benchmark with 1-step inference, unless stated otherwise. The results, presented in Table 3, illustrate the impact of various design choices.

Table 3: Ablation studies for VarFlow on SD1.5 (LoRA) / MS COCO 10k (1-step). The "Optimal" configuration provides the baseline.

Ablation Focus	Configuration	FID (↓)	CLIP Score (↑)	AES (↑)
VarFlow (Optimal)	$\beta = 1$**,** $\tilde{w}(t) = \sigma_t^2$**, Full Loss,** $K = 16$	**5.08**	**30.70**	**5.85**
Energy Distance β	$\beta = 0.5$	5.15	30.52	5.78
	$\beta = 1.0$ (Optimal)	5.08	30.70	5.85
	$\beta = 1.5$	5.12	30.61	5.80
	$\beta = 1.9$ (Near MMD-like)	5.18	30.48	5.75
Time Weighting $\tilde{w}(t)$	$\tilde{w}(t) = 1.0$ (Uniform)	5.25	30.33	5.65
	$\tilde{w}(t) = \sigma_t^2$ (Optimal, VSD-like)	5.08	30.70	5.85
	$\tilde{w}(t) = 1/\sigma_t$ (Score-like)	5.18	30.58	5.77
	$\tilde{w}(t) = \bar{\alpha}_t/\sigma_t^2$ (SNR-based)	5.11	30.65	5.82
Batch Size K (per GPU)	$K = 4$	5.28	30.25	5.60
	$K = 8$	5.16	30.50	5.76
	$K = 16$ (Optimal)	5.08	30.70	5.85
	$K = 32$	5.05	30.75	5.82
Estimator for Cross-Term	Paired Estimator ($\frac{1}{K}\sum \|\boldsymbol{x}_t^{S,(k)} - \boldsymbol{x}_t^{T,(k)}\|^{\beta}$) (Optimal)	5.08	30.70	5.85
	U-statistic Estimator ($\frac{1}{K^2}\sum\sum \|\boldsymbol{x}_t^{S,(i)} - \boldsymbol{x}_t^{T,(j)}\|^{\beta}$)	5.10	30.63	5.81

Choice of Energy Distance Exponent $\boldsymbol{\beta}$: Varying the exponent β for the energy distance term showed that $\beta = 1.0$ (L1-like distance in the energy score) achieves an optimal balance across differen metrics. While performance is relatively stable for $\beta \in [0.5, 1.5]$, extreme values or those approaching $\beta = 2.0$ (which relates to MMD with a squared Euclidean kernel) showed a slight decline in overall quality. This confirms $\beta = 1.0$ as a robust and effective default.

Impact of Time Weighting $\tilde{\boldsymbol{w}}(\boldsymbol{t})$: Different time weighting schemes for the loss across time steps t significantly impacted performance. Uniform weighting ($\tilde{w}(t) = 1.0$) was suboptimal compared to adaptive schemes. The VSD-like weighting $\tilde{w}(t) = \sigma_t^2$ (proportional to noise variance) and an SNR-based weighting ($\tilde{w}(t) = \bar{\alpha}_t/\sigma_t^2$) yielded the best results across all metrics, indicating the importance of carefully balancing contributions from different noise levels to achieve high perceptual quality, semantic alignment, and aesthetics.

Influence of Batch Size $\boldsymbol{K}$ (per GPU): The batch size K used for Monte Carlo estimation of the VarFlow loss is crucial. Smaller batch sizes ($K = 4, 8$) resulted in degraded performance across all metrics, particularly FID and AES, likely due to higher variance in gradient estimates. Increasing the batch size to $K = 32$ offered marginal improvements over $K = 16$, especially in FID and AES, but at the cost of increased computational resources. $K = 16$ provides a strong balance between performance and efficiency.

Estimator for the Cross-Term in $\mathcal{L}_t$: We compared the default paired estimator for the cross-term $\mathbb{E}[\|\boldsymbol{x}_t^S - \boldsymbol{x}_t^T\|^{\beta}]$ with a full U-statistic estimator. The U-statistic, while theoretically offering lower bias, is more computationally intensive. In practice, the simpler paired estimator performed comparably, or even slightly better in some configurations for overall balance, making it the more efficient and effective choice for the cross-term.

7 Conclusion

We introduced VarFlow, a novel distillation framework that trains a fast, one-step student generator by directly minimizing the energy distance between the student's and teacher's (or data-derived) noisy marginal distributions. A key advantage of VarFlow is its sample-based objective, which entirely bypasses the need for explicit computation or approximation of the intractable student score function, a common challenge in methods like VSD. Extensive experiments demonstrate that VarFlow achieves state-of-the-art or highly competitive performance on various benchmarks.

References

Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

David Berthelot, Arnaud Autef, Jierui Lin, Dian Ang Yap, Shuangfei Zhai, Siyuan Hu, Daniel Zheng, Walter Talbott, and Eric Gu. Tract: Denoising diffusion models with transitive closure time-distillation. *arXiv preprint arXiv:2303.04248*, 2023.

Minwoo Byeon, Beomhee Park, Haecheon Kim, Sungjun Lee, Woonhyuk Baek, and Saehoon Kim. Coyo-700m: Image-text pair dataset. `https://github.com/kakaobrain/coyo-dataset`, 2022.

Valentin De Bortoli, Alexandre Galashov, J Swaroop Guntupalli, Guangyao Zhou, Kevin Murphy, Arthur Gretton, and Arnaud Doucet. Distributional diffusion models with scoring rules. *arXiv preprint arXiv:2502.02483*, 2025.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Justin Domke. Provable smoothness guarantees for black-box variational inference. In *International Conference on Machine Learning*, pages 2587–2596. PMLR, 2020.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, and Frederic et al. Boesel. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024.

Zhengyang Geng, Ashwini Pokle, William Luo, Justin Lin, and J Zico Kolter. Consistency models made easy. *arXiv preprint arXiv:2406.14548*, 2024.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012.

Jonathan Heek, Emiel Hoogeboom, and Tim Salimans. Multistep consistency models. *arXiv preprint arXiv:2403.06807*, 2024.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 6626–6637, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851, 2020.

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

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Tejas Jayashankar, J Jon Ryu, and Gregory Wornell. Score-of-mixture training: Training one-step generative models made simple. *arXiv preprint arXiv:2502.09609*, 2025.

https://doi.org/10.52202/085713-1847

S Kang et al. Distilling diffusion models into conditional gans. *ECCV*, 2024.

Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24174–24184, 2024.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. Mmd gan: Towards deeper understanding of moment matching network. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Shanchuan Lin, Anran Wang, and Xiao Yang. Sdxl-lightning: Progressive adversarial diffusion distillation. *arXiv preprint arXiv:2402.13929*, 2024.

Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley. Audioldm: Text-to-audio generation with latent diffusion models. *arXiv preprint arXiv:2301.12503*, 2023.

Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. *arXiv preprint arXiv:2410.11081*, 2024.

Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023a.

Weijian Luo, Tianyang Hu, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhihua Zhang. Diff-instruct: A universal approach for transferring knowledge from pre-trained diffusion models. *Advances in Neural Information Processing Systems*, 36:76525–76546, 2023b.

Weijian Luo, Zemin Huang, Zhengyang Geng, J Zico Kolter, and Guo-jun Qi. One-step diffusion distillation through score implicit matching. *Advances in Neural Information Processing Systems*, 37:115377–115408, 2024.

Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14297–14306, 2023.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

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

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

Yuxi Ren, Xin Xia, Yanzuo Lu, Jiacheng Zhang, Jie Wu, Pan Xie, Xing Wang, and Xuefeng Xiao. Hyper-sd: Trajectory segmented consistency model for efficient image synthesis. *arXiv preprint arXiv:2404.13686*, 2024.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

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

Tim Salimans, Thomas Mensink, Jonathan Heek, and Emiel Hoogeboom. Multistep distillation of diffusion models via moment matching. *Advances in Neural Information Processing Systems*, 37: 36046–36070, 2024.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*, 2023.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022.

Huiyang Shao, Qianqian Xu, Peisong Wen, Peifeng Gao, Zhiyong Yang, and Qingming Huang. Building bridge across the time: Disruption and restoration of murals in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20259–20269, 2023.

Huiyang Shao, Xin Xia, Yuhong Yang, Yuxi Ren, Xing Wang, and Xuefeng Xiao. Rayflow: Instance-aware diffusion acceleration via adaptive flow trajectories. *arXiv preprint arXiv:2503.07699*, 2025.

Xinwei Shen, Nicolai Meinshausen, and Tong Zhang. Reverse markov learning: Multi-step generative models for complex distributions. *arXiv preprint arXiv:2502.13747*, 2025.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. *arXiv preprint arXiv:2310.14189*, 2023.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, volume 32, pages 11895–11907, 2019.

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

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. 2023.

Yansheng Sun, Jiarui Wu, and Run Chen. Inductive moment matching. *arXiv preprint arXiv:2503.05727*, 2025.

Gábor J Székely, Maria L Rizzo, et al. Testing for equal distributions in high dimension. *InterStat*, 5 (16.10):1249–1272, 2004.

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

Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in Neural Information Processing Systems*, 36, 2024.

Sirui Xie, Zhisheng Xiao, Diederik P Kingma, Tingbo Hou, Ying Nian Wu, Kevin Murphy, Tim Salimans, Ben Poole, and Ruiqi Gao. Em distillation for one-step diffusion models. *arXiv preprint arXiv:2405.16852*, 2024.

Hanshu Yan, Xingchao Liu, Jiachun Pan, Jun Hao Liew, Qiang Liu, and Jiashi Feng. Perflow: Piecewise rectified flow as universal plug-and-play accelerator. *arXiv preprint arXiv:2405.07510*, 2024.

 https://doi.org/10.52202/085713-1847

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. *arXiv preprint arXiv:2405.14867*, 2024a.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6613–6623, 2024b.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6613–6623, 2024c.

Mingtian Zhang, Jiajun He, Wenlin Chen, Zijing Ou, José Miguel Hernández-Lobato, Bernhard Schölkopf, and David Barber. Towards training one-step diffusion models without distillation. *arXiv preprint arXiv:2502.08005*, 2025.

Jianbin Zheng, Minghui Hu, Zhongyi Fan, Chaoyue Wang, Changxing Ding, Dacheng Tao, and Tat-Jen Cham. Trajectory consistency distillation. *arXiv preprint arXiv:2402.19159*, 2024.

Mingyuan Zhou, Huangjie Zheng, Zhendong Wang, Mingzhang Yin, and Hai Huang. Score identity distillation: Exponentially fast distillation of pretrained diffusion models for one-step generation. In *Forty-first International Conference on Machine Learning*, 2024.

https://doi.org/10.52202/085713-1847

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction accurately summarize the proposed framework, the claimed benefits, and the intention to provide theoretical backing, which are reflected in the subsequent sections (Method, Theoretical Guarantees, Proof Details).

2. **Limitations**

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

   Answer: [Yes]

   Justification: We provide limitation in Section F.

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: The paper includes dedicated sections for Theoretical Analysis (Sections D and E). Assumptions are stated there, and detailed proofs are provided for the theorems and propositions presented.

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We describe the algorithms and specific details (Sec. A) about the experimental setup, such as datasets used, specific hyperparameters (learning rates, batch sizes, parameters), training duration, evaluation metrics implementation, or baseline implementation details necessary to fully reproduce the claimed experimental results.

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

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

   Answer: [No]

   Justification: We will provide the source code and data after the paper is accpeted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Key details about the experimental setup are provided, including specific hyperparameter values (e.g., learning rates, optimizer types, batch sizes), data splits for training/testing on the mentioned benchmarks, and specific schedules used.

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: We conducted extensive experiments to demonstrate the effectiveness of our method.

 https://doi.org/10.52202/085713-1847

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We provide information about the computational resources (Sec. A) used for the experiments.

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: This paper appears to be focused on algorithmic contributions and does not inherently conflict with the NeurIPS Code of Ethics.

10. **Broader impacts**

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

    Answer: [No]

    Justification: Our work primarily focuses on algorithm design and technical aspects, with limited direct societal impact. As such, the paper does not extensively discuss broader societal implications, positive or negative.

11. **Safeguards**

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

    Answer: [NA]

    Justification: This question is not applicable.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We cites prior works.

13. **New assets**

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

    Answer: [NA]

    Justification: Documentation requirements for new assets are not applicable.

14. **Crowdsourcing and research with human subjects**

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

    Answer: [NA]

    Justification: The research presented in the paper does not involve crowdsourcing experiments or research with human subjects.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

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

Answer: [NA]

Justification: The research presented does not involve human subjects, so IRB approval is not applicable.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: Large language models (or derived text encoders like CLIP/T5) are used to obtain text embeddings for conditioning, which is standard practice in text-to-image generation. LLMs are not an important, original, or non-standard component of the core novel methodology presented in this paper.

https://doi.org/10.52202/085713-1847