

A Experiment Details

A.1 Overview.

VarFlow trains a student generator $g_\phi(z)$ by minimizing the energy distance (Equation (8)) between its induced noisy data distribution $p_{\phi,t}(x_t)$ and the target noisy distribution $q_t(x_t)$. The target distribution $q_t(x_t)$ is derived by noising real data samples $x_0 \sim q_{\text{data}}(x_0)$ according to the forward diffusion process (Equation (8)).

A.2 Student Architectures and Initialization.

The architecture of the student generator g_ϕ is chosen to be comparable to relevant teacher models or established high-performance networks for fair evaluation.

- **Large-Scale Text-to-Image (T2I) Models (Table 2):** For models like Stable Diffusion v1.5 (SD1.5) [Rombach et al., 2022], Stable Diffusion XL (SDXL) [Podell et al., 2023], and SD3-Medium [Esser et al., 2024], we employ LoRA [Hu et al., 2022] for parameter-efficient fine-tuning. The student g_ϕ comprises the frozen pre-trained weights of the respective teacher model, with trainable LoRA layers inserted into the query, key, value, and output projection layers of their attention mechanisms. We use a LoRA rank $r = 64$ and $\alpha = 32$. The number of trainable LoRA parameters is typically $\sim 1\text{-}5\%$ of the full model, depending on the base architecture.
- **General Image Generation Models (Table 1):** For datasets like ImageNet [Deng et al., 2009] and CIFAR-10 [Krizhevsky et al., 2009], g_ϕ is a full U-Net based architecture.
 - *ImageNet 64x64 & 256x256:* The student uses a U-Net architecture with parameter counts (e.g., 296M for 64x64, 550M for 256x256) comparable to ADM [Dhariwal and Nichol, 2021].
 - *CIFAR-10 32x32:* A smaller U-Net (e.g., 56M parameters) similar to those used in DDPM [Ho et al., 2020] or EDM [Karras et al., 2024] for this dataset is employed.

In the "Training from scratch" setting in Table 1, g_ϕ is initialized with standard random weights (e.g., Kaiming He initialization). In the "Diffusion distillation" setting, g_ϕ is initialized from the weights of a pre-trained teacher diffusion model (e.g., an EDM [Karras et al., 2024] model pre-trained on the respective dataset) and then fully fine-tuned using the VarFlow objective. The teacher for ImageNet 64x64 distillation was an EDM-S model (296M params, FID 1.36), and for ImageNet 256x256, an EDM-L model (550M params, FID 2.70).

A.3 Teacher Information for VarFlow Loss.

VarFlow’s training objective (Equation (8)) directly uses noised real data samples to form $q_t(x_t)$. It does not require explicit access to a teacher model’s score function (e.g., $\epsilon_{\text{teacher}}$) during loss computation. Pre-trained teacher models are primarily leveraged for: (1) providing the architectural backbone and initial weights for g_ϕ in distillation scenarios (both full fine-tuning and LoRA-based), and (2) defining the noise schedule $(\bar{\alpha}_t, \sigma_t)$ that VarFlow uses.

A.4 Training Parameters.

Unless otherwise specified, the following parameters were used:

- **Optimizer:** AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.999$.
- **Learning Rate:** For LoRA-based T2I distillation, a constant learning rate of 1×10^{-5} was used for SD1.5 and SDXL, and SD3-Medium due to their sensitivity. For full model fine-tuning or training from scratch (Table 1), a learning rate of 1×10^{-4} with a linear warmup of 10,000 steps and cosine decay was used.
- **Batch Size:** For LoRA T2I, a per-GPU batch size of 4 was used on 8 NVIDIA A100 (80GB) GPUs, resulting in an effective batch size of 32. For Table 1 experiments, a per-GPU batch size of 32 (ImageNet 64x64, CIFAR-10) or 8 (ImageNet 256x256) was used, on 8 GPUs. The batch size K in Equation (10) refers to the per-GPU batch size.
- **Training Duration:** For LoRA T2I distillation, models were trained for 150,000 iterations. For Table 1, models were trained for 400,000 iterations (ImageNet) or 200,000 iterations (CIFAR-10).
- **VarFlow Loss Parameters:**
 - Energy distance exponent $\beta = 1.0$ (as per ablation in Table 3).
 - Time weighting $\tilde{w}(t) = \sigma_t^2$, where σ_t^2 is the variance of the noise at time t .

- The loss was estimated using the paired Monte Carlo estimator (Equation (10)).
- **Noise Schedule and Time Sampling:** The noise schedule $(\bar{\alpha}_t, \sigma_t)$ and continuous time range $[t_{\min}, t_{\max}]$ (e.g., $[0.002, 80.0]$ for EDM-like schedules) matched those of the original teacher models (for distillation) or standard VP SDE schedules [Karras et al., 2024] (for training from scratch). Time t was sampled uniformly from $[t_{\min}, t_{\max}]$.

A.5 Datasets.

- **Training Data for T2I Models:** We used a 20M sample high-quality, filtered subset of the COYO-700M dataset [Byeon et al., 2022], filtered for aesthetic score > 6.0 and CLIP ViT-L/14 image-text similarity > 0.3 .
- **Evaluation Datasets:**
 - MS COCO 2017 Validation: For T2I evaluation (Table 2 and relevant parts of Table 1), we used 30,000 prompts to generate images at 512x512 resolution, unless specified. For 1024x1024 (e.g., SDXL), 10,000 prompts were used.
 - ImageNet: 1.28M training images, 50k validation images for class-conditional generation at 64x64 and 256x256.
 - CIFAR-10: 50k training images, 10k test images for unconditional generation at 32x32.

A.6 Evaluation Metrics.

Performance was assessed using the following standard metrics. FID scores were calculated against reference statistics computed on the respective training sets.

- **FID (Fréchet Inception Distance)** [Heusel et al., 2017]: Lower is better.
- **CLIP Score** [Radford et al., 2021, Hessel et al., 2021]: For MS COCO, we report CLIP similarity using the OpenCLIP ViT-H/14 model. Higher is better.
- **Aesthetic Score (AES):** For T2I, we use the LAION Aesthetic Predictor (ViT-L/14 backbone, version 2 [Schuhmann et al., 2022]) to report mean aesthetic scores. Higher is better.

Baselines. VarFlow is benchmarked against a comprehensive suite of existing methods. These are categorized as follows:

- **Foundational Multi-Step Diffusion Models:** These represent the original high-performance diffusion models, often serving as performance ceilings or the basis for teacher models in distillation.
 - DDPM [Ho et al., 2020]
 - ADM [Dhariwal and Nichol, 2021]
 - EDM [Karras et al., 2024]
- **Pre-trained Teacher/Base Models (for Distillation Context):** These are specific, widely-used large-scale pre-trained models that VarFlow and other distillation methods aim to accelerate. They are often the direct "teachers" in distillation scenarios.
 - Stable Diffusion v1.5 (SD15) [Rombach et al., 2022]
 - Stable Diffusion XL (SDXL) [Podell et al., 2023]
 - SD3-Medium [Esser et al., 2024]
- **One-Step Generative Models (Trained from Scratch or via Alternative Paradigms):** These methods are designed for one-step generation, often without relying on distillation from a pre-trained multi-step diffusion model.
 - Consistency Training (CT) [Song et al., 2023]
 - Improved CT (iCT / iCT-deep) [Song and Dhariwal, 2023]
 - ECT [Geng et al., 2024]
 - SMT Jayashankar et al. [2025]
- **Diffusion Distillation and Few-Step Acceleration Techniques:** This category includes methods that, like VarFlow, aim to distill knowledge from a pre-trained teacher diffusion model into a faster student model capable of high-quality generation in one or very few steps.
 - Progressive Distillation (PD) [Salimans and Ho, 2022]
 - TRACT [Berthelot et al., 2023]
 - Consistency Distillation (CD) [Song et al., 2023]
 - Diff-Instruct [Luo et al., 2023b]
 - MultiStep-CD [Heek et al., 2024]
 - DMD (w/o reg) [Yin et al., 2024c] and DMD2 (w/ GAN) [Yin et al., 2024a]
 - MMD-distill [Salimans et al., 2024] (referred to as MMD in Table 1)

- Score Distillation (SiD) [Zhou et al., 2024]
- SiM [Luo et al., 2024]
- Score Matching Distillation (SMD) Jayashankar et al. [2025]
- Latent Consistency Models (LCM) [Luo et al., 2023a]
- Trajectory Consistency Distillation (TCD) [Zheng et al., 2024]
- Hyper-SD [Ren et al., 2024]
- SDXL-Turbo Sauer et al. [2023]
- SDXL-Lightning [Lin et al., 2024]
- PeRFlow [Yan et al., 2024]
- EMD [Xie et al., 2024] (specifically applied to SD3.5)

For all comparisons, we strive for fairness by utilizing publicly available official implementations or reported results where possible, and by matching student architectures or using consistent fine-tuning strategies like LoRA where specified.

A.7 Training Settings

Our training methodology for large-scale generative modeling on ImageNet 256×256 and MS COCO 512×512 is built upon the well-established practices of SMT Jayashankar et al. [2025], DMD Yin et al. [2024b], and sCD Lu and Song [2024].

A.7.1 ImageNet 256x256

Dataset We use the standard ImageNet dataset, which contains 1.28 million training images and 50,000 validation images. All images were resized and center-cropped to a resolution of 256×256 pixels.

Architectures and Teachers

- **From Scratch (CT, iCT, SMT):** A U-Net architecture with approximately 296 million parameters was trained from a random initialization, consistent with baseline models in the literature.
- **Distillation (PD, CD, DMD2):** The student model is a U-Net with 296 million parameters, initialized from pre-trained EDM-S weights. The teacher model is the EDM-L, with 550 million parameters and a baseline FID score of 2.70.

Training Hyperparameters All baseline models were trained for 400,000 iterations using the AdamW optimizer with a learning rate of 1×10^{-4} , which included warmup and decay phases. The batch size was set to 64. Method-specific loss functions and weighting schemes were implemented as described in their original papers.

Table 4: Model Configurations for ImageNet 256×256 .

Method	Category	Architecture (Params)	Teacher FID	Key Hyperparameters / Notes
CT / iCT	From Scratch	U-Net (296M)	N/A	$w(t)$ as per original CT paper
ECT	From Scratch	U-Net (280M)	N/A	Timestep discretization schedule
SMT	From Scratch	U-Net (296M)	N/A	Original loss formulation
PD / CD	Distillation	U-Net (296M)	2.70	Student initialized from EDM-S
DMD2 / SiD	Distillation	U-Net (296M)	2.70	Auxiliary student score network
VarFlow (ours)	Distillation	U-Net (296M)	2.70	Energy score $\beta = 1.0$, $w(t) = \sigma_t^2$

A.7.2 MS COCO 512x512

Dataset The training data consists of 20 million image-text pairs filtered from the COYO-700M dataset. For evaluation, we used a set of 30,000 prompts from the MS COCO 2017 validation set.

Architecture and Distillation The teacher model is the pre-trained SDXL-base model. Student models were created by fine-tuning the SDXL backbone using Low-Rank Adaptation (LoRA). The trainable LoRA layers were configured with a rank of 64 and an alpha value of 32.

Training from Scratch For methods like CT, iCT, and SMT, "from scratch" refers to training a U-Net with approximately 296 million parameters from a random initialization. The text encoder, an OpenCLIP ViT-H/14, remained frozen during this process.

Training Hyperparameters

- **From Scratch Methods:** Trained for 400,000 iterations with the AdamW optimizer, a batch size of 32, and a learning rate of 1×10^{-4} (with warmup and decay).
- **Distillation Methods:** Trained for 150,000 iterations with the AdamW optimizer, a batch size of 32, and a constant learning rate of 1×10^{-5} .
- The noise scheduling strategy was kept consistent with the ImageNet experiments.

B Notation Summary

Table 5 provides a summary of the key mathematical notation used throughout this paper.

Table 5: Key Notation Summary.

Symbol	Description	Symbol	Description
\mathbf{x}_0	Clean data sample	\mathbf{x}_t	Noisy data sample at time t
$q_{\text{data}}(\mathbf{x}_0)$	True data distribution	$q_t(\mathbf{x}_t \mathbf{x}_0)$	Forward noising kernel $q_t(\mathbf{x}_t \mathbf{x}_0)$
$\bar{\alpha}_t, \sigma_t$	Noise schedule parameters ($\sigma_t^2 = 1 - \bar{\alpha}_t$)	ϵ	Standard Gaussian noise $\mathcal{N}(\mathbf{0}, \mathbf{I})$
$q(\mathbf{x}_0 \mathbf{x}_t)$	True posterior $q(\mathbf{x}_0 \mathbf{x}_t)$	$p_{\theta}(\cdot \mathbf{x}_t, t)$	DDM's learned posterior $p_{\theta}(\mathbf{x}_0 \mathbf{x}_t, t)$
$q_t(\mathbf{x}_t)$	Teacher's (true data) noisy marginal $q_t(\mathbf{x}_t)$	$p_{\phi, t}(\mathbf{x}_t)$	Student's noisy marginal $p_{\phi, t}(\mathbf{x}_t)$
$G_{\theta}(\mathbf{x}_t, t, \xi)$	DDM conditional generator	θ	Parameters of G_{θ} or $\epsilon_{\text{teacher}}$
$g_{\phi}(\mathbf{z})$	VarFlow/VSD single-step student generator	ϕ	Parameters of g_{ϕ}
$\epsilon_{\text{teacher}}$	Pre-trained teacher noise predictor	ϵ_{aux}	VSD's auxiliary student noise predictor
$s_t^*(\mathbf{x}_t)$	Teacher's score $\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t)$	$s_{\phi, t}(\mathbf{x}_t)$	Student's score $\nabla_{\mathbf{x}_t} \log p_{\phi, t}(\mathbf{x}_t)$ (VSD)
ξ	Input noise for DDM generator $G_{\theta}(\sim p_{\xi}(\xi))$	\mathbf{z}	Input noise for student generator $g_{\phi}(\sim p_{\mathbf{z}}(\mathbf{z}))$
$\hat{\mathbf{x}}_0$	Predicted clean data sample	$\mathcal{S}(P, y)$	Scoring rule value for forecast P , outcome y
$\mathcal{S}(P, Q)$	Expected score $\mathbb{E}_{Y \sim Q}[\mathcal{S}(P, Y)]$	$\mathcal{S}_{\text{Energy}}^{(\beta)}$	Energy score with exponent β
$D_{\text{Energy}}^{(\beta)}(P, Q)^2$	Squared energy distance	$w(t), \tilde{w}(t)$	Time weighting functions
K	Batch size for VarFlow loss est.	m	MC samples for DDM loss est.
T	Maximum diffusion time	$\mathcal{N}(\mu, \Sigma)$	Normal distribution
\mathbf{I}	Identity matrix	$\ \cdot\ $	L2 norm (unless specified)
$q(\mathbf{x}_0, \mathbf{x}_t)$	Joint distribution $q(\mathbf{x}_0, \mathbf{x}_t)$	$\text{Unif}[a, b]$	Uniform distribution on $[a, b]$
$\mathbf{x}_0^{S, (k)}, \mathbf{x}_t^{S, (k)}$	k -th clean/noisy student sample	$\mathbf{x}_0^{T, (k)}, \mathbf{x}_t^{T, (k)}$	k -th clean/noisy teacher sample
$\mathbf{x}_0^{(i)}, \mathbf{x}_t^{(i)}, t^{(i)}$	i -th sample/time in batch (DDM)	$\hat{\mathbf{x}}_0^{(i, j)}$	j -th posterior sample for i -th data (DDM)

C Further Details on Preliminaries

C.1 Diffusion Models: SDEs, ODEs, and Score Functions

The forward diffusion process transforming data $\mathbf{x}_0 \sim q_{\text{data}}(\mathbf{x}_0)$ into noise \mathbf{x}_t can be described by a stochastic differential equation (SDE). A common choice is the Variance Preserving (VP) SDE Song

et al. [2021]:

$$d\mathbf{x}(t) = -\frac{1}{2}\gamma(t)\mathbf{x}(t)dt + \sqrt{\gamma(t)}d\mathbf{w}(t), \quad t \in [0, T], \quad (11)$$

where $\mathbf{x}(0) = \mathbf{x}_0$, $\mathbf{w}(t)$ is a standard Wiener process, and $\gamma(t) > 0$ is a noise schedule function. The solution $\mathbf{x}(t)$ (denoted \mathbf{x}_t) conditioned on \mathbf{x}_0 follows $q_t(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1 - \bar{\alpha}_t)\mathbf{I})$, where $\bar{\alpha}_t = \exp(-\int_0^t \gamma(s)ds)$. This matches Equation (1) with $\sigma_t^2 = 1 - \bar{\alpha}_t$.

The reverse of this process is described by a reverse-time SDE Anderson [1982], Song et al. [2021]:

$$d\mathbf{x}(t) = \left[-\frac{1}{2}\gamma(t)\mathbf{x}(t) - \gamma(t)\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t) \right] dt + \sqrt{\gamma(t)}d\bar{\mathbf{w}}(t), \quad (12)$$

where $d\bar{\mathbf{w}}(t)$ is a reverse-time Wiener process, and $q_t(\mathbf{x}_t) = \int q_t(\mathbf{x}_t|\mathbf{x}_0)q_{\text{data}}(\mathbf{x}_0)d\mathbf{x}_0$ is the marginal distribution of noisy data. The crucial term is the score function $\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t)$. Alternatively, sampling can use the probability flow ODE Song et al. [2021]:

$$\frac{d\mathbf{x}(t)}{dt} = -\frac{1}{2}\gamma(t)\mathbf{x}(t) - \frac{1}{2}\gamma(t)\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t). \quad (13)$$

The true score $\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t)$ is approximated by a neural network $s_\theta(\mathbf{x}_t, t)$. Denoising Score Matching (DSM) Vincent [2011] trains this network by minimizing:

$$\mathcal{L}_{\text{DSM}}(\theta) = \mathbb{E}_{t \sim \text{Unif}[0, T]} \mathbb{E}_{\mathbf{x}_0 \sim q_{\text{data}}} \mathbb{E}_{\mathbf{x}_t \sim q_t(\cdot|\mathbf{x}_0)} [w_{\text{DSM}}(t) \|s_\theta(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t|\mathbf{x}_0)\|_2^2], \quad (14)$$

where $w_{\text{DSM}}(t)$ is a time-dependent weighting. From $q_t(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, \sigma_t^2\mathbf{I})$, the conditional score is $\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t|\mathbf{x}_0) = -(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0)/\sigma_t^2$. Using $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sigma_t\epsilon$ (where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is the sampled noise), this conditional score simplifies to $-\epsilon/\sigma_t$. Parameterizing $s_\theta(\mathbf{x}_t, t) = -\epsilon_\theta(\mathbf{x}_t, t)/\sigma_t$, where ϵ_θ predicts noise, the DSM objective becomes:

$$\begin{aligned} \mathcal{L}_{\text{DSM}}(\theta) &= \mathbb{E}_{t, \mathbf{x}_0, \epsilon} \left[w_{\text{DSM}}(t) \left\| -\frac{\epsilon_\theta(\mathbf{x}_t, t)}{\sigma_t} - \left(-\frac{\epsilon}{\sigma_t} \right) \right\|_2^2 \right] \\ &= \mathbb{E}_{t, \mathbf{x}_0, \epsilon} \left[\frac{w_{\text{DSM}}(t)}{\sigma_t^2} \|\epsilon_\theta(\mathbf{x}_t, t) - \epsilon\|_2^2 \right]. \end{aligned} \quad (15)$$

This matches Equation (2) with an appropriate overall weighting function $w(t) = w_{\text{DSM}}(t)/\sigma_t^2$. Minimizing this objective makes $s_\theta(\mathbf{x}_t, t)$ a good estimate of the marginal score $\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t)$ Song et al. [2021].

C.2 Variational Score Distillation (VSD) Details

VSD aims to minimize $D_{\text{KL}}(p_{\phi, t}(\mathbf{x}_t) \| q_t(\mathbf{x}_t))$ averaged over time (Equation (5)). For a fixed t , the gradient with respect to ϕ (see Wang et al. [2024], Appendix D.1 for a detailed derivation, or standard results such as Domke [2020]) is:

$$\nabla_\phi D_{\text{KL}}(p_{\phi, t} \| q_t) = \nabla_\phi \mathbb{E}_{\mathbf{x}_t \sim p_{\phi, t}} [\log p_{\phi, t}(\mathbf{x}_t) - \log q_t(\mathbf{x}_t)].$$

Using reparameterization $\mathbf{x}_t(\phi, \mathbf{z}, \epsilon) = \sqrt{\bar{\alpha}_t}g_\phi(\mathbf{z}) + \sigma_t\epsilon$, where samples from $p_{\phi, t}$ are obtained by sampling $\mathbf{z} \sim p_\mathbf{z}(\mathbf{z})$, $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and applying this transformation, the gradient can be transformed (e.g., via integration by parts or identities involving score functions, as shown in Wang et al. [2024]) to:

$$\nabla_\phi D_{\text{KL}}(p_{\phi, t} \| q_t) = \mathbb{E}_{\mathbf{z} \sim p_\mathbf{z}(\mathbf{z}), \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} [(\nabla_{\mathbf{x}_t} \log p_{\phi, t}(\mathbf{x}_t) - \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t)) \cdot \nabla_\phi \mathbf{x}_t(\phi, \mathbf{z}, \epsilon)],$$

where $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}g_\phi(\mathbf{z}) + \sigma_t\epsilon$. Since $\nabla_\phi \mathbf{x}_t(\phi, \mathbf{z}, \epsilon) = \sqrt{\bar{\alpha}_t}\nabla_\phi g_\phi(\mathbf{z})$, substituting this gives:

$$\nabla_\phi D_{\text{KL}}(p_{\phi, t} \| q_t) = \mathbb{E}_{\mathbf{z} \sim p_\mathbf{z}(\mathbf{z}), \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} [(s_{\phi, t}(\mathbf{x}_t) - s_t^*(\mathbf{x}_t)) \cdot (\sqrt{\bar{\alpha}_t}\nabla_\phi g_\phi(\mathbf{z}))].$$

Including the expectation over t and the weighting $\tilde{w}(t)$ yields Equation (6).

The Student Score Approximation in VSD: The student score $s_{\phi, t}(\mathbf{x}_t) = \nabla_{\mathbf{x}_t} \log p_{\phi, t}(\mathbf{x}_t)$ is intractable. VSD typically approximates it:

1. **Auxiliary Network:** Train a network $\epsilon_{\text{aux}}(\mathbf{x}_t, t; \omega)$ to approximate the noise corresponding to $s_{\phi, t}(\mathbf{x}_t)$. That is, ϵ_{aux} is trained via DSM on samples generated by g_ϕ : $\mathcal{L}_{\text{aux}}(\omega) = \mathbb{E}_{t, \mathbf{z}, \epsilon} [\tilde{w}'(t) \|\epsilon_{\text{aux}}(\sqrt{\bar{\alpha}_t}g_\phi(\mathbf{z}) + \sigma_t\epsilon, t; \omega) - \epsilon\|_2^2]$. Then, in the VSD gradient for ϕ , $s_{\phi, t}(\mathbf{x}_t)$ is replaced by $-\epsilon_{\text{aux}}(\mathbf{x}_t, t; \omega)/\sigma_t$. This typically involves alternating optimization of ϕ and ω .

2. **Conditional Score Approximation (SDS-like):** Replace the marginal student score $s_{\phi,t}(\mathbf{x}_t)$ with the score of the conditional distribution $p_{\phi,t}(\mathbf{x}_t|\mathbf{z}^*)$ for the specific \mathbf{z}^* that generated \mathbf{x}_t : $s_{\phi,t}(\mathbf{x}_t) \approx \nabla_{\mathbf{x}_t} \log p_{\phi,t}(\mathbf{x}_t|\mathbf{z}^*) = -\epsilon_{\text{true}}/\sigma_t$, where ϵ_{true} is the noise used to generate $\mathbf{x}_t = \sqrt{\alpha_t}g_{\phi}(\mathbf{z}^*) + \sigma_t\epsilon_{\text{true}}$.

Using approximation (2) and $s_t^*(\mathbf{x}_t) \approx -\epsilon_{\text{teacher}}(\mathbf{x}_t, t)/\sigma_t$, the score difference term in the VSD gradient (Equation (6)) becomes (proportional to):

$$\left(\frac{-\epsilon_{\text{true}}}{\sigma_t} - \frac{-\epsilon_{\text{teacher}}(\mathbf{x}_t, t)}{\sigma_t} \right) = \frac{1}{\sigma_t} (\epsilon_{\text{teacher}}(\mathbf{x}_t, t) - \epsilon_{\text{true}}). \quad (16)$$

The full gradient update using this approximation involves multiplying by $\tilde{w}(t)\sqrt{\alpha_t}\nabla_{\phi}g_{\phi}(\mathbf{z}^*)$ and taking expectations. VarFlow aims to avoid such approximations.

C.3 Energy Score and Energy Distance Details

The energy score $\mathcal{S}_{\text{Energy}}^{(\beta)}(P, y)$ is given by Equation (3). The negative expected energy score, when a forecast P is issued and the outcome Y is drawn from a true underlying distribution Q , is:

$$-\mathcal{S}_{\text{Energy}}^{(\beta)}(P, Q) = -\mathbb{E}_{Y \sim Q}[\mathcal{S}_{\text{Energy}}^{(\beta)}(P, Y)] = \mathbb{E}_{X \sim P, Y \sim Q}[\|X - Y\|^\beta] - \frac{1}{2}\mathbb{E}_{X, X' \stackrel{\text{iid}}{\sim} P}[\|X - X'\|^\beta]. \quad (17)$$

Minimizing this quantity with respect to P is a common objective in probabilistic forecasting and distributional learning. This minimization is equivalent to minimizing the squared energy distance $D_{\text{Energy}}^{(\beta)}(P, Q)^2$, defined in Equation (4), up to a term that is constant with respect to P . Specifically:

$$\begin{aligned} D_{\text{Energy}}^{(\beta)}(P, Q)^2 &= 2\mathbb{E}_{X \sim P, Y \sim Q}\|X - Y\|^\beta - \mathbb{E}_{X, X' \stackrel{\text{iid}}{\sim} P}\|X - X'\|^\beta - \mathbb{E}_{Y, Y' \stackrel{\text{iid}}{\sim} Q}\|Y - Y'\|^\beta \\ &= 2\left(\mathbb{E}_{X \sim P, Y \sim Q}\|X - Y\|^\beta - \frac{1}{2}\mathbb{E}_{X, X' \stackrel{\text{iid}}{\sim} P}\|X - X'\|^\beta\right) - \mathbb{E}_{Y, Y' \stackrel{\text{iid}}{\sim} Q}\|Y - Y'\|^\beta \\ &= 2\left(-\mathcal{S}_{\text{Energy}}^{(\beta)}(P, Q)\right) - \mathbb{E}_{Y, Y' \stackrel{\text{iid}}{\sim} Q}\|Y - Y'\|^\beta. \end{aligned}$$

Since $\mathbb{E}_{Y, Y' \stackrel{\text{iid}}{\sim} Q}\|Y - Y'\|^\beta$ does not depend on P , minimizing $-\mathcal{S}_{\text{Energy}}^{(\beta)}(P, Q)$ with respect to P is equivalent to minimizing $D_{\text{Energy}}^{(\beta)}(P, Q)^2$. The energy distance is an Integral Probability Metric (IPM) and defines a true metric on the space of probability distributions with finite β -moments for $\beta \in (0, 2)$. For $\beta = 1$, $D_{\text{Energy}}^{(1)}(P, Q)$ is related to a Maximum Mean Discrepancy (MMD) with a specific kernel Székely et al. [2004].

D Distributional Diffusion Models (DDM): Formulation and Details

This appendix details Distributional Diffusion Models (DDMs), a conceptual framework for learning the complete conditional posterior distribution $q(\mathbf{x}_0|\mathbf{x}_t)$ using scoring rules.

D.1 Motivation and Parameterization

Standard diffusion models (Section 3.1) typically train $\epsilon_{\theta}(\mathbf{x}_t, t)$ to predict noise ϵ . This is equivalent to predicting the mean of the posterior $\mathbb{E}[\mathbf{x}_0|\mathbf{x}_t]$ (up to scaling factors). However, the true posterior $q(\mathbf{x}_0|\mathbf{x}_t)$ might have richer structures (such as multimodality or specific variance characteristics) that are lost when only considering the mean. DDMs aim to model this entire distribution $p_{\theta}(\cdot|\mathbf{x}_t, t)(\cdot|\mathbf{x}_t, t)$.

We parameterize this learned posterior using a conditional generator $G_{\theta} : \mathbb{R}^d \times [0, T] \times \mathbb{R}^k \rightarrow \mathbb{R}^d$. It takes the noisy data \mathbf{x}_t , time t , and auxiliary noise $\xi \sim p_{\xi}(\xi)$ (where $p_{\xi}(\xi)$ is a base distribution, e.g., $\mathcal{N}(\mathbf{0}, \mathbf{I}_k)$) to output a sample of clean data $\hat{\mathbf{x}}_0 = G_{\theta}(\mathbf{x}_t, t, \xi)$. The distribution of $\hat{\mathbf{x}}_0$ over different samples of ξ (for fixed \mathbf{x}_t, t) defines the learned posterior $p_{\theta}(\cdot|\mathbf{x}_t, t)(\cdot|\mathbf{x}_t, t)$, which aims to approximate the true posterior $q(\mathbf{x}_0|\mathbf{x}_t)$.

D.2 Training Objectives from Scoring Rules for DDM

The DDM generator G_θ is trained by minimizing the negative expected score of its output distribution $p_\theta(\cdot|\mathbf{x}_t, t)(\cdot|\mathbf{x}_t, t)$ when evaluated against true samples $\mathbf{x}_0 \sim q(\cdot|\mathbf{x}_t)$. This minimization is averaged over the joint distribution of true data and noised data, $q(\mathbf{x}_0, \mathbf{x}_t)(\mathbf{x}_0, \mathbf{x}_t) = q_t(\mathbf{x}_t|\mathbf{x}_0)q_{\text{data}}(\mathbf{x}_0)$, and over time t .

Energy Diffusion Loss for DDM. Using the energy score $\mathcal{S}_{\text{Energy}}^{(\beta)}$ (Equation (3)) with an exponent $\beta \in (0, 2)$, the objective for a fixed (\mathbf{x}_t, t) is to minimize $-\mathbb{E}_{\mathbf{x}_0 \sim q(\cdot|\mathbf{x}_t, t)}[\mathcal{S}_{\text{Energy}}^{(\beta)}(p_\theta(\cdot|\mathbf{x}_t, t)(\cdot|\mathbf{x}_t, t), \mathbf{x}_0)]$. This equals (from Equation (17), treating $P = p_\theta(\cdot|\mathbf{x}_t, t)(\cdot|\mathbf{x}_t, t)$ and the observation Y as $\mathbf{x}_0 \sim q(\cdot|\mathbf{x}_t, t)$):

$$\mathbb{E}_{\substack{\hat{\mathbf{x}}_0 \sim p_\theta(\cdot|\mathbf{x}_t, t)(\cdot|\mathbf{x}_t, t) \\ \mathbf{x}_0 \sim q(\cdot|\mathbf{x}_t, t)}}[\|\hat{\mathbf{x}}_0 - \mathbf{x}_0\|^\beta] - \frac{1}{2}\mathbb{E}_{\hat{\mathbf{x}}_0, \hat{\mathbf{x}}'_0 \stackrel{\text{iid}}{\sim} p_\theta(\cdot|\mathbf{x}_t, t)(\cdot|\mathbf{x}_t, t)}[\|\hat{\mathbf{x}}_0 - \hat{\mathbf{x}}'_0\|^\beta].$$

Averaging over $(\mathbf{x}_0, \mathbf{x}_t) \sim q(\mathbf{x}_0, \mathbf{x}_t)$ and $t \sim \text{Unif}[0, T]$, the *Energy Diffusion Loss for DDM* is:

$$\mathcal{L}_{\text{Energy-DDM}}^{(\beta)}(\theta) = \mathbb{E}_{t, (\mathbf{x}_0, \mathbf{x}_t) \sim q(\mathbf{x}_0, \mathbf{x}_t)} \left[w_D(t) \left(\mathbb{E}_{\xi \sim p_\xi(\xi)} [\|G_\theta(\mathbf{x}_t, t, \xi) - \mathbf{x}_0\|^\beta] - \frac{1}{2} \mathbb{E}_{\xi, \xi' \stackrel{\text{iid}}{\sim} p_\xi(\xi)} [\|G_\theta(\mathbf{x}_t, t, \xi) - G_\theta(\mathbf{x}_t, t, \xi')\|^\beta] \right) \right]. \quad (18)$$

$w_D(t)$ is a time weighting function specific to DDM training. The first term encourages the generated samples $G_\theta(\mathbf{x}_t, t, \xi)$ to be close to the true clean data \mathbf{x}_0 . The second term (a diversity term) encourages the spread of samples generated by G_θ (by varying ξ) to match the spread of the true posterior $q(\mathbf{x}_0|\mathbf{x}_t)$.

Generalized Kernel Diffusion Loss for DDM. More generally, for a score derived from a kernel-like function $\rho(\cdot, \cdot)$ of the form $\mathcal{S}(P, y) = \frac{1}{2}\mathbb{E}_{X, X' \stackrel{\text{iid}}{\sim} P}[\rho(X, X')] - \mathbb{E}_{X \sim P}[\rho(X, y)]$, the DDM loss becomes:

$$\mathcal{L}_{\text{Kernel-DDM}}^{(\rho)}(\theta) = \mathbb{E}_{t, (\mathbf{x}_0, \mathbf{x}_t) \sim q(\mathbf{x}_0, \mathbf{x}_t)} \left[w_D(t) \left(\mathbb{E}_\xi [\rho(G_\theta(\mathbf{x}_t, t, \xi), \mathbf{x}_0)] - \frac{1}{2} \mathbb{E}_{\xi, \xi'} [\rho(G_\theta(\mathbf{x}_t, t, \xi), G_\theta(\mathbf{x}_t, t, \xi'))] \right) \right]. \quad (19)$$

The energy score uses $\rho(\mathbf{x}, \mathbf{y}) = -\|\mathbf{x} - \mathbf{y}\|^\beta$.

Monte Carlo Estimation for DDM Loss. For $\mathcal{L}_{\text{Energy-DDM}}^{(\beta)}$, given a batch sample $(\mathbf{x}_0, \mathbf{x}_t, t)$, we draw $m \geq 2$ samples of auxiliary noise $\{\xi^{(j)}\}_{j=1}^m \stackrel{\text{iid}}{\sim} p_\xi(\xi)$. An unbiased empirical estimator for the term in the large parentheses in Equation (18) is:

$$\ell_{\text{DDM-sample}}(\mathbf{x}_0, \mathbf{x}_t, t; \theta) = \frac{1}{m} \sum_{j=1}^m \|G_\theta(\mathbf{x}_t, t, \xi^{(j)}) - \mathbf{x}_0\|^\beta - \frac{1}{m(m-1)} \sum_{j=1}^m \sum_{\substack{k=1 \\ k \neq j}}^m \frac{1}{2} \|G_\theta(\mathbf{x}_t, t, \xi^{(j)}) - G_\theta(\mathbf{x}_t, t, \xi^{(k)})\|^\beta. \quad (20)$$

The batch loss is then $\hat{\mathcal{L}}_{\text{Energy-DDM}} = \frac{1}{B} \sum_{i=1}^B w_D(t^{(i)}) \ell_{\text{DDM-sample}}(\mathbf{x}_0^{(i)}, \mathbf{x}_t^{(i)}, t^{(i)}; \theta)$, where B is the batch size. This loss is backpropagated through G_θ .

D.3 Theoretical Guarantees for DDM

Theorem D.1 (Consistency for DDM Learning). *Assume $w_D(t) > 0$ almost everywhere on $t \in [0, T]$, and t is sampled with full support on this interval. If the DDM loss \mathcal{L}_{DDM} uses a strictly proper scoring rule (such as the energy score with $\beta \in (0, 2)$), then $\mathcal{L}_{\text{DDM}}(\theta)$ is minimized if and only if the learned posterior $p_\theta(\cdot|\mathbf{x}_t, t)(\cdot|\mathbf{x}_t, t)$ matches the true posterior $q(\mathbf{x}_0|\mathbf{x}_t)$ for q_t -almost every \mathbf{x}_t and for almost every $t \in [0, T]$ (assuming sufficient model capacity for G_θ and standard regularity conditions).*

Proof. See Section D.5.1. □

Proposition D.2 (DDM Recovery of Standard MSE Loss). *If the DDM generator $G_\theta(\mathbf{x}_t, t, \xi)$ is deterministic with respect to ξ , i.e., $G_\theta(\mathbf{x}_t, t, \xi) = f_\theta(\mathbf{x}_t, t)$, and the energy score exponent $\beta = 2$ is used in $\mathcal{L}_{\text{Energy-DDM}}^{(\beta)}$, then the DDM loss reduces to the standard Mean Squared Error (MSE) objective for predicting \mathbf{x}_0 : $\mathbb{E}_{t, (\mathbf{x}_0, \mathbf{x}_t) \sim q(\mathbf{x}_0, \mathbf{x}_t)} [w_D(t) \|f_\theta(\mathbf{x}_t, t) - \mathbf{x}_0\|^2]$.*

Proof. See Section D.5.2. □

This proposition highlights that standard diffusion model training, which often aims to predict the mean of the posterior (related to \mathbf{x}_0 or ϵ), can be viewed as a special, deterministic case of the DDM framework.

Proposition D.3 (DDM Connection to Integrated Energy Distance). *Minimizing $\mathcal{L}_{\text{Energy-DDM}}^{(\beta)}(\theta)$ is equivalent (up to terms constant with respect to θ) to minimizing the expected energy distance between the learned and true posteriors: $\mathbb{E}_{t \sim \text{Unif}[0, T]} \mathbb{E}_{\mathbf{x}_t \sim q_t} [w_D(t) D_{\text{Energy}}^{(\beta)}(p_\theta(\cdot | \mathbf{x}_t, t) \| q(\mathbf{x}_0 | \mathbf{x}_t))^2]$.*

Proof. See Section D.5.3. □

This proposition explicitly shows that DDM, when using the energy score, aims to match the learned posterior $p_\theta(\cdot | \mathbf{x}_t, t)$ to the true posterior $q(\mathbf{x}_0 | \mathbf{x}_t)$ in the sense of energy distance, averaged over \mathbf{x}_t and t .

D.4 DDM Algorithms

Algorithm 2 outlines a general training procedure for a DDM. Algorithm 3 illustrates how a trained DDM generator G_θ could be incorporated into a DDPM-style sampling loop to generate samples.

Algorithm 2 DDM Training

Require: DDM Generator G_θ , dataset \mathcal{D} (samples $\mathbf{x}_0 \sim q_{\text{data}}$), noise schedule $(\bar{\alpha}_t, \sigma_t)$, exponent $\beta \in (0, 2)$, number of posterior MC samples $m \geq 2$, latent distribution $p_\xi(\xi)$ (e.g., $\mathcal{N}(\mathbf{0}, \mathbf{I}_k)$), time weighting $w_D(t)$, batch size B_{DDM} , learning rate η_{DDM} .

- 1: Initialize parameters θ of G_θ .
 - 2: **repeat**
 - 3: Sample a minibatch of clean data $\{\mathbf{x}_0^{(i)}\}_{i=1}^{B_{\text{DDM}}} \stackrel{\text{iid}}{\sim} q_{\text{data}}(\mathbf{x}_0)$ from \mathcal{D} .
 - 4: $\mathcal{L}_{\text{batch}} \leftarrow 0$.
 - 5: **for** $i = 1$ to B_{DDM} **do**
 - 6: Sample $t^{(i)} \sim \text{Unif}[0, T]$.
 - 7: Sample $\epsilon^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Form $\mathbf{x}_t^{(i)} \leftarrow \sqrt{\bar{\alpha}_{t^{(i)}}} \mathbf{x}_0^{(i)} + \sigma_{t^{(i)}} \epsilon^{(i)}$.
 - 8: Sample m latent noise vectors $\{\xi^{(i,j)}\}_{j=1}^m \stackrel{\text{iid}}{\sim} p_\xi(\xi)$.
 - 9: Generate m posterior samples: $\hat{\mathbf{x}}_0^{(i,j)} \leftarrow G_\theta(\mathbf{x}_t^{(i)}, t^{(i)}, \xi^{(i,j)})$ for $j = 1, \dots, m$.
 - 10: Compute sample loss $\ell_{\text{DDM}}^{(i)}$ for $(\mathbf{x}_0^{(i)}, \mathbf{x}_t^{(i)}, t^{(i)})$ using $\{\hat{\mathbf{x}}_0^{(i,j)}\}_{j=1}^m$ via Equation (20).
 - 11: $\mathcal{L}_{\text{batch}} \leftarrow \mathcal{L}_{\text{batch}} + w_D(t^{(i)}) \ell_{\text{DDM}}^{(i)}$.
 - 12: **end for**
 - 13: Compute overall batch loss $\hat{\mathcal{L}}_{\text{Energy-DDM}} = \frac{1}{B_{\text{DDM}}} \mathcal{L}_{\text{batch}}$.
 - 14: Compute gradient $\nabla_\theta \hat{\mathcal{L}}_{\text{Energy-DDM}}$.
 - 15: Update parameters: $\theta \leftarrow \theta - \eta_{\text{DDM}} \nabla_\theta \hat{\mathcal{L}}_{\text{Energy-DDM}}$.
 - 16: **until** convergence criteria met
 - 17: **return** Trained DDM generator G_θ .
-

Remark D.4. The DDM sampling procedure (Algorithm 3) illustrates one way to use the stochastically generated $\hat{\mathbf{x}}_0$ from G_θ (due to the random ξ) to inform a standard diffusion reverse step (like DDPM). Each reverse step thus samples from the learned posterior $p_\theta(\mathbf{x}_0 | \mathbf{x}_t, t)$. Other samplers, such as DDIM Song et al. [2020], could also be adapted to use $\hat{\mathbf{x}}_0$ from G_θ . The specific integration of G_θ 's output into the sampling process can be flexible. The crucial aspect is that G_θ aims to provide samples from the full posterior distribution, not just its mean.

Algorithm 3 DDM Sampling

Require: Trained DDM Generator G_θ , number of steps N_{steps} , time discretization $T = t_{N_{\text{steps}}} > \dots > t_1 > t_0 = 0$, noise schedule parameters $(\bar{\alpha}_t, \sigma_t, \alpha_t^{\text{step}})$, latent distribution $p_\xi(\xi)$. (Here α_t^{step} refers to $\bar{\alpha}_t / \bar{\alpha}_{t_s}$ where t_s is the previous time step).

- 1: Sample initial noise $\mathbf{x}_{t_{N_{\text{steps}}}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
- 2: **for** $j = N_{\text{steps}}$ down to 1 **do**
- 3: Let $t \leftarrow t_j$ and $s \leftarrow t_{j-1}$. Current noisy sample is $\mathbf{x}_t \leftarrow \mathbf{x}_{t_j}$.
- 4: Sample Gaussian noise $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $j > 1$, else $\mathbf{z} = \mathbf{0}$.
- 5: Sample DDM latent noise $\xi \sim p_\xi(\xi)$.
- 6: Predict a clean sample using the DDM: $\hat{\mathbf{x}}_0 \leftarrow G_\theta(\mathbf{x}_t, t, \xi)$. \triangleright Core DDM step, produces one sample from posterior
- 7: Predict noise $\hat{\epsilon}$ consistent with this $\hat{\mathbf{x}}_0$: $\hat{\epsilon} \leftarrow (\mathbf{x}_t - \sqrt{\bar{\alpha}_t} \hat{\mathbf{x}}_0) / \sigma_t$.
- 8: Compute DDPM posterior mean (using predicted $\hat{\epsilon}$): $\mu_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t^{\text{step}}}} \left(\mathbf{x}_t - \frac{1 - \alpha_t^{\text{step}}}{\sqrt{1 - \bar{\alpha}_t}} \hat{\epsilon} \right)$, where $\alpha_t^{\text{step}} = \bar{\alpha}_t / \bar{\alpha}_s$.
- 9: Compute DDPM posterior variance: $\tilde{\sigma}_j^2 = \frac{1 - \bar{\alpha}_s}{1 - \bar{\alpha}_t} (1 - \alpha_t^{\text{step}})$.
- 10: Sample next state: $\mathbf{x}_s \leftarrow \mu_\theta(\mathbf{x}_t, t) + \sqrt{\tilde{\sigma}_j^2} \mathbf{z}$. Set $\mathbf{x}_{t_{j-1}} \leftarrow \mathbf{x}_s$.
- 11: **end for**
- 12: **return** \mathbf{x}_{t_0} (which should approximate $\mathbf{x}_0 \sim q_{\text{data}}$).

D.5 Proofs for DDM Theoretical Results

D.5.1 Proof of Theorem D.1 (Consistency for DDM Learning)

Let \mathcal{S} be a strictly proper scoring rule. The DDM loss is designed to minimize the expected negative score, defined as

$$-\mathcal{S}(P_{\theta, \mathbf{x}_t, t}, Q_{\mathbf{x}_t, t}),$$

where $P_{\theta, \mathbf{x}_t, t}$ denotes the learned posterior distribution $p_\theta(\cdot | \mathbf{x}_t, t)(\cdot | \mathbf{x}_t, t)$ (induced by $G_\theta(\mathbf{x}_t, t, \xi)$ over ξ), and $Q_{\mathbf{x}_t, t}$ denotes the true posterior distribution $q(\cdot | \mathbf{x}_t, t)$.

Step 1: Relating the Loss to a Divergence This minimization is equivalent to minimizing the score-induced divergence

$$D_{\mathcal{S}}(P_{\theta, \mathbf{x}_t, t} \| Q_{\mathbf{x}_t, t}) = \mathcal{S}(Q_{\mathbf{x}_t, t}, Q_{\mathbf{x}_t, t}) - \mathcal{S}(P_{\theta, \mathbf{x}_t, t}, Q_{\mathbf{x}_t, t}),$$

because the term $\mathcal{S}(Q_{\mathbf{x}_t, t}, Q_{\mathbf{x}_t, t})$ is constant with respect to the parameters θ of $P_{\theta, \mathbf{x}_t, t}$.

Step 2: Properties of Strictly Proper Scoring Rules By the definition of a strictly proper scoring rule, the divergence $D_{\mathcal{S}}(P \| Q)$ satisfies

$$D_{\mathcal{S}}(P \| Q) \geq 0 \quad \text{for any distributions } P, Q,$$

and

$$D_{\mathcal{S}}(P \| Q) = 0 \quad \text{if and only if } P = Q \text{ almost everywhere (a.e.).}$$

Step 3: Expressing the DDM Loss The DDM loss can be written as the expectation of these non-negative divergence terms:

$$\mathcal{L}_{\text{DDM}}(\theta) = \mathbb{E}_{t \sim \text{Unif}[0, T]} \mathbb{E}_{(\mathbf{x}_0, \mathbf{x}_t) \sim q(\mathbf{x}_0, \mathbf{x}_t)} [w_D(t) (-\mathcal{S}(P_{\theta, \mathbf{x}_t, t}, \mathbf{x}_0))].$$

Since the expectation over \mathbf{x}_0 is with respect to $Q_{\mathbf{x}_t, t}$, this can be rewritten as

$$\mathcal{L}_{\text{DDM}}(\theta) = \mathbb{E}_{t \sim \text{Unif}[0, T]} \mathbb{E}_{\mathbf{x}_t \sim q_t(\mathbf{x}_t)} [w_D(t) (-\mathcal{S}(P_{\theta, \mathbf{x}_t, t}, Q_{\mathbf{x}_t, t}))].$$

Step 4: Substituting the Divergence Recall that

$$-\mathcal{S}(P, Q) = D_{\mathcal{S}}(P \| Q) - \mathcal{S}(Q, Q).$$

Substituting this into the loss, we obtain

$$\mathcal{L}_{\text{DDM}}(\theta) = \mathbb{E}_{t \sim \text{Unif}[0, T]} \mathbb{E}_{\mathbf{x}_t \sim q_t(\mathbf{x}_t)} [w_D(t) (D_{\mathcal{S}}(P_{\theta, \mathbf{x}_t, t} \| Q_{\mathbf{x}_t, t}) - \mathcal{S}(Q_{\mathbf{x}_t, t}, Q_{\mathbf{x}_t, t}))].$$

Step 5: Minimization with Respect to Parameters Minimizing $\mathcal{L}_{\text{DDM}}(\theta)$ with respect to θ is therefore equivalent to minimizing

$$\mathbb{E}_{t, \mathbf{x}_t} [w_D(t) D_S(P_{\theta, \mathbf{x}_t, t} \parallel Q_{\mathbf{x}_t, t})],$$

since the term $S(Q_{\mathbf{x}_t, t}, Q_{\mathbf{x}_t, t})$ does not depend on θ .

Step 6: Characterizing the Optimum Assume that $w_D(t) > 0$ almost everywhere and that the expectations cover the relevant supports. Then, the overall loss is minimized if and only if

$$D_S(P_{\theta, \mathbf{x}_t, t} \parallel Q_{\mathbf{x}_t, t}) = 0$$

for q_t -almost every \mathbf{x}_t and for almost every $t \in [0, T]$.

Step 7: Conclusion By the property of strictly proper scoring rules, this implies

$$P_{\theta, \mathbf{x}_t, t} = Q_{\mathbf{x}_t, t} \quad \text{almost everywhere,}$$

under these conditions, assuming sufficient model capacity for G_θ and the necessary regularity conditions for the distributions and the scoring rule.

D.5.2 Proof of Theorem D.2 (DDM Recovery of MSE Loss)

Consider the Energy Diffusion Loss $\mathcal{L}_{\text{Energy-DDM}}^{(\beta)}(\theta)$ (Equation (18)) with exponent $\beta = 2$. If the generator $G_\theta(\mathbf{x}_t, t, \xi) = f_\theta(\mathbf{x}_t, t)$ is deterministic (i.e., it does not depend on the auxiliary noise ξ), then: The first term within the expectation over ξ, ξ' in Equation (18) becomes:

$$\mathbb{E}_{\xi \sim p_\xi(\xi)} [\|f_\theta(\mathbf{x}_t, t) - \mathbf{x}_0\|^2] = \|f_\theta(\mathbf{x}_t, t) - \mathbf{x}_0\|^2,$$

as $f_\theta(\mathbf{x}_t, t)$ is constant with respect to ξ . The second term becomes:

$$-\frac{1}{2} \mathbb{E}_{\xi, \xi' \sim p_\xi(\xi)} [\|f_\theta(\mathbf{x}_t, t) - f_\theta(\mathbf{x}_t, t)\|^2] = -\frac{1}{2} \mathbb{E}_{\xi, \xi'} [\|\mathbf{0}\|^2] = 0.$$

Therefore, the expression within the large parentheses in Equation (18) simplifies to $\|f_\theta(\mathbf{x}_t, t) - \mathbf{x}_0\|^2$. The total DDM loss then becomes:

$$\mathcal{L}_{\text{Energy-DDM}}^{(2)}(\theta) = \mathbb{E}_{t, (\mathbf{x}_0, \mathbf{x}_t) \sim q(\mathbf{x}_0, \mathbf{x}_t)} [w_D(t) \|f_\theta(\mathbf{x}_t, t) - \mathbf{x}_0\|^2],$$

which is the standard Mean Squared Error (MSE) objective for predicting \mathbf{x}_0 from \mathbf{x}_t at time t , weighted by $w_D(t)$.

D.5.3 Proof of Theorem D.3 (DDM Connection to Integrated Energy Distance)

Let $P_{\mathbf{x}_t, t}$ denote the learned posterior distribution $p_\theta(\cdot | \mathbf{x}_t, t)$ and $Q_{\mathbf{x}_t, t}$ denote the true posterior distribution $q(\cdot | \mathbf{x}_t, t)$. The term inside the expectation $\mathbb{E}_{t, (\mathbf{x}_0, \mathbf{x}_t) \sim q(\mathbf{x}_0, \mathbf{x}_t)}$ in the DDM loss definition, Equation (18), is $w_D(t)$ times the negative expected energy score where the forecast is $P_{\mathbf{x}_t, t}$ and observations \mathbf{x}_0 are drawn from $Q_{\mathbf{x}_t, t}$. That is, $w_D(t) \times (-\mathcal{S}_{\text{Energy}}^{(\beta)}(P_{\mathbf{x}_t, t}, Q_{\mathbf{x}_t, t}))$. Explicitly, from Equation (17):

$$-\mathcal{S}_{\text{Energy}}^{(\beta)}(P_{\mathbf{x}_t, t}, Q_{\mathbf{x}_t, t}) = \mathbb{E}_{\substack{\hat{\mathbf{x}}_0 \sim P_{\mathbf{x}_t, t} \\ \mathbf{x}_0 \sim Q_{\mathbf{x}_t, t}}} [\|\hat{\mathbf{x}}_0 - \mathbf{x}_0\|^\beta] - \frac{1}{2} \mathbb{E}_{\hat{\mathbf{x}}_0, \hat{\mathbf{x}}'_0 \sim P_{\mathbf{x}_t, t}} [\|\hat{\mathbf{x}}_0 - \hat{\mathbf{x}}'_0\|^\beta].$$

The squared energy distance $D_{\text{Energy}}^{(\beta)}(P, Q)^2$ is defined in Equation (4):

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

Comparing these, we see that:

$$D_{\text{Energy}}^{(\beta)}(P_{\mathbf{x}_t, t}, Q_{\mathbf{x}_t, t})^2 = 2 \left(\mathbb{E}_{\substack{\hat{\mathbf{x}}_0 \sim P_{\mathbf{x}_t, t} \\ \mathbf{x}_0 \sim Q_{\mathbf{x}_t, t}}} [\|\hat{\mathbf{x}}_0 - \mathbf{x}_0\|^\beta] - \frac{1}{2} \mathbb{E}_{\hat{\mathbf{x}}_0, \hat{\mathbf{x}}'_0 \sim P_{\mathbf{x}_t, t}} [\|\hat{\mathbf{x}}_0 - \hat{\mathbf{x}}'_0\|^\beta] \right) - \mathbb{E}_{\mathbf{x}_0, \mathbf{x}'_0 \sim Q_{\mathbf{x}_t, t}} [\|\mathbf{x}_0 - \mathbf{x}'_0\|^\beta].$$

So, $D_{\text{Energy}}^{(\beta)}(P_{\mathbf{x}_t,t}, Q_{\mathbf{x}_t,t})^2 = 2(-\mathcal{S}_{\text{Energy}}^{(\beta)}(P_{\mathbf{x}_t,t}, Q_{\mathbf{x}_t,t})) - \mathbb{E}_{\mathbf{x}_0, \mathbf{x}'_0 \sim Q_{\mathbf{x}_t,t}} [\|\mathbf{x}_0 - \mathbf{x}'_0\|^\beta]$. Rearranging this gives:

$$-\mathcal{S}_{\text{Energy}}^{(\beta)}(P_{\mathbf{x}_t,t}, Q_{\mathbf{x}_t,t}) = \frac{1}{2} D_{\text{Energy}}^{(\beta)}(P_{\mathbf{x}_t,t}, Q_{\mathbf{x}_t,t})^2 + \frac{1}{2} \mathbb{E}_{\mathbf{x}_0, \mathbf{x}'_0 \sim Q_{\mathbf{x}_t,t}} [\|\mathbf{x}_0 - \mathbf{x}'_0\|^\beta].$$

The DDM loss $\mathcal{L}_{\text{Energy-DDM}}^{(\beta)}(\boldsymbol{\theta})$ is therefore:

$$\mathbb{E}_{t \sim \text{Unif}[0,T]} \mathbb{E}_{\mathbf{x}_t \sim q_t(\mathbf{x}_t)} \left[w_D(t) \left(\frac{1}{2} D_{\text{Energy}}^{(\beta)}(P_{\mathbf{x}_t,t}, Q_{\mathbf{x}_t,t})^2 + \frac{1}{2} \mathbb{E}_{\mathbf{x}_0, \mathbf{x}'_0 \sim Q_{\mathbf{x}_t,t}} [\|\mathbf{x}_0 - \mathbf{x}'_0\|^\beta] \right) \right].$$

Since the term $\frac{1}{2} \mathbb{E}_{\mathbf{x}_0, \mathbf{x}'_0 \sim Q_{\mathbf{x}_t,t}} [\|\mathbf{x}_0 - \mathbf{x}'_0\|^\beta]$ depends only on the true posterior $Q_{\mathbf{x}_t,t}$ and not on the learnable parameters $\boldsymbol{\theta}$ (which define $P_{\mathbf{x}_t,t}$), minimizing $\mathcal{L}_{\text{Energy-DDM}}^{(\beta)}(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ is equivalent to minimizing:

$$\mathbb{E}_{t \sim \text{Unif}[0,T]} \mathbb{E}_{\mathbf{x}_t \sim q_t(\mathbf{x}_t)} \left[w_D(t) \frac{1}{2} D_{\text{Energy}}^{(\beta)}(P_{\mathbf{x}_t,t} \parallel Q_{\mathbf{x}_t,t})^2 \right].$$

This establishes the connection that DDM using the energy score minimizes an expected energy distance between the learned and true conditional posterior distributions.

E Proofs of Theoretical Results for VarFlow

E.1 Proof of Theorem 4.1 (Consistency of VarFlow Objective)

The VarFlow loss $\mathcal{L}_{\text{VarFlow}}^{(\beta)}(\phi)$ is defined in Equation (8) as:

$$\mathcal{L}_{\text{VarFlow}}^{(\beta)}(\phi) = \mathbb{E}_{t \sim \text{Unif}[0,T]} \left[\tilde{w}(t) \underbrace{\left(\mathbb{E}_{\substack{\mathbf{x}_t^S \sim p_{\phi,t} \\ \mathbf{x}_t^T \sim q_t}} [\|\mathbf{x}_t^S - \mathbf{x}_t^T\|^\beta] - \frac{1}{2} \mathbb{E}_{\substack{\mathbf{x}_t^S \sim p_{\phi,t} \\ \mathbf{x}_t^{S'} \sim p_{\phi,t}}} [\|\mathbf{x}_t^S - \mathbf{x}_t^{S'}\|^\beta] \right)}_{-\mathcal{S}_{\text{Energy}}^{(\beta)}(p_{\phi,t}, q_t) \text{ (see Equation (17))}} \right].$$

Let $P_t = p_{\phi,t}(\mathbf{x}_t)$ denote the student's noisy marginal distribution at time t , and $Q_t = q_t(\mathbf{x}_t)$ denote the teacher's (true data) noisy marginal distribution at time t . The term inside the expectation over t is $\tilde{w}(t) \left(-\mathcal{S}_{\text{Energy}}^{(\beta)}(P_t, Q_t) \right)$. As established in Section C.3 (and Section D.5.3 by analogy), the negative expected energy score can be related to the squared energy distance:

$$-\mathcal{S}_{\text{Energy}}^{(\beta)}(P_t, Q_t) = \frac{1}{2} D_{\text{Energy}}^{(\beta)}(P_t, Q_t)^2 + \frac{1}{2} \mathbb{E}_{\mathbf{x}_t^T, \mathbf{x}_t^{T'} \sim Q_t} [\|\mathbf{x}_t^T - \mathbf{x}_t^{T'}\|^\beta].$$

The term $\frac{1}{2} \mathbb{E}_{\mathbf{x}_t^T, \mathbf{x}_t^{T'} \sim Q_t} [\|\mathbf{x}_t^T - \mathbf{x}_t^{T'}\|^\beta]$ depends only on the teacher's distribution Q_t and is therefore constant with respect to the student's parameters ϕ (which define P_t). Thus, the VarFlow loss can be written as:

$$\mathcal{L}_{\text{VarFlow}}^{(\beta)}(\phi) = \mathbb{E}_{t \sim \text{Unif}[0,T]} \left[\tilde{w}(t) \left(\frac{1}{2} D_{\text{Energy}}^{(\beta)}(P_t, Q_t)^2 + \text{const}_t \right) \right],$$

where $\text{const}_t = \frac{1}{2} \mathbb{E}_{\mathbf{x}_t^T, \mathbf{x}_t^{T'} \sim Q_t} [\|\mathbf{x}_t^T - \mathbf{x}_t^{T'}\|^\beta]$ does not depend on ϕ . For $\beta \in (0, 2)$, the energy distance $D_{\text{Energy}}^{(\beta)}(P_t, Q_t)$ is a metric on the space of probability distributions with finite β -moments. Consequently, $D_{\text{Energy}}^{(\beta)}(P_t, Q_t)^2 \geq 0$, and $D_{\text{Energy}}^{(\beta)}(P_t, Q_t)^2 = 0$ if and only if $P_t = Q_t$ almost everywhere. Given the assumptions that $\tilde{w}(t) > 0$ almost everywhere on $[0, T]$ and that the sampling of t has full support on this interval, the loss $\mathcal{L}_{\text{VarFlow}}^{(\beta)}(\phi)$ is an expectation of non-negative terms (namely, $\tilde{w}(t) \frac{1}{2} D_{\text{Energy}}^{(\beta)}(P_t, Q_t)^2$) plus terms that are constant with respect to ϕ . This loss is minimized if and only if $D_{\text{Energy}}^{(\beta)}(P_t, Q_t)^2 = 0$ for almost every $t \in [0, T]$ for which $\tilde{w}(t) > 0$. This implies that $P_t = Q_t$ for almost every $t \in [0, T]$. In terms of the specific distributions, this means $p_{\phi,t}(\mathbf{x}_t) = q_t(\mathbf{x}_t)$ for almost every t , assuming that the student generator g_ϕ has sufficient capacity to model the target distributions and that all relevant distributions possess finite β -moments.

E.2 VarFlow Gradient Computation and Comparison to VSD

This section details why VarFlow does not encounter the same computational issues related to complex gradients as traditional VSD.

VSD Gradient and the Intractable Student Score. The VSD gradient for the student generator parameters ϕ is given by Equation (6):

$$\nabla_{\phi} \mathcal{L}_{\text{VSD-KL}} = \mathbb{E}_{t, \mathbf{z}, \epsilon} \left[\underbrace{\tilde{w}(t) (s_{\phi, t}(\mathbf{x}_t) - s_t^*(\mathbf{x}_t))}_{\text{Score difference}} \cdot \nabla_{\phi} (\sqrt{\alpha_t} g_{\phi}(\mathbf{z})) \right],$$

where $\mathbf{x}_t = \sqrt{\alpha_t} g_{\phi}(\mathbf{z}) + \sigma_t \epsilon$. The teacher's score $s_t^*(\mathbf{x}_t) = \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t)$ is obtained from the pre-trained teacher diffusion model (e.g., $s_t^*(\mathbf{x}_t) \approx -\epsilon_{\text{teacher}}(\mathbf{x}_t, t)/\sigma_t$). The critical term is the student's score $s_{\phi, t}(\mathbf{x}_t) = \nabla_{\mathbf{x}_t} \log p_{\phi, t}(\mathbf{x}_t)$. The student's noisy marginal distribution $p_{\phi, t}(\mathbf{x}_t)$ is defined as:

$$p_{\phi, t}(\mathbf{x}_t) = \int p_{\phi, t}(\mathbf{x}_t | g_{\phi}(\mathbf{z}')) p_{\mathbf{z}}(\mathbf{z}') d\mathbf{z}' = \int \mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t} g_{\phi}(\mathbf{z}'), \sigma_t^2 \mathbf{I}) p_{\mathbf{z}}(\mathbf{z}') d\mathbf{z}'.$$

Computing $\nabla_{\mathbf{x}_t} \log p_{\phi, t}(\mathbf{x}_t)$ involves taking the gradient of the logarithm of an integral over all possible latent codes \mathbf{z}' . This is generally intractable for complex, high-dimensional generators g_{ϕ} and distributions $p_{\mathbf{z}}$. VSD thus requires approximations for $s_{\phi, t}(\mathbf{x}_t)$, such as:

- Training an auxiliary score network $\epsilon_{\text{aux}}(\mathbf{x}_t, t; \omega)$ (parameterized by ω) to approximate $-\sigma_t s_{\phi, t}(\mathbf{x}_t)$ via denoising score matching on samples from g_{ϕ} . This introduces an additional network to train and potential optimization challenges (e.g., alternating updates).
- Using the score of the conditional distribution $p_{\phi, t}(\mathbf{x}_t | \mathbf{z})$ for the specific \mathbf{z} that generated \mathbf{x}_t (i.e., $\nabla_{\mathbf{x}_t} \log p_{\phi, t}(\mathbf{x}_t | \mathbf{z}) = -\epsilon/\sigma_t$). This is a strong simplification and may not accurately reflect the score of the true marginal $p_{\phi, t}(\mathbf{x}_t)$.

These approximations can lead to biases, training instabilities, or failure to fully match the target distribution.

VarFlow Gradient Computation. The VarFlow loss is given by Equation (8):

$$\begin{aligned} \mathcal{L}_{\text{VarFlow}}^{(\beta)}(\phi) = \mathbb{E}_t \left[\tilde{w}(t) \left(\underbrace{\mathbb{E}_{\substack{\mathbf{z}^S \sim p_{\mathbf{z}}, \epsilon^S \sim \mathcal{N}} \\ \mathbf{x}_0^T \sim q_{\text{data}}, \epsilon^T \sim \mathcal{N}}} [\|\mathbf{x}_t^S(\phi, \mathbf{z}^S, \epsilon^S) - \mathbf{x}_t^T(\mathbf{x}_0^T, \epsilon^T)\|^\beta]}_{\text{Term 1: Cross-term}} \right. \right. \\ \left. \left. - \frac{1}{2} \underbrace{\mathbb{E}_{\substack{\mathbf{z}^{S1} \sim p_{\mathbf{z}}, \epsilon^{S1} \sim \mathcal{N}} \\ \mathbf{z}^{S2} \sim p_{\mathbf{z}}, \epsilon^{S2} \sim \mathcal{N}}} [\|\mathbf{x}_t^{S1}(\phi, \mathbf{z}^{S1}, \epsilon^{S1}) - \mathbf{x}_t^{S2}(\phi, \mathbf{z}^{S2}, \epsilon^{S2})\|^\beta]}_{\text{Term 2: Student-student term}} \right) \right], \end{aligned}$$

where, for a given t : $\mathbf{x}_t^S(\phi, \mathbf{z}, \epsilon) = \sqrt{\alpha_t} g_{\phi}(\mathbf{z}) + \sigma_t \epsilon$, $\mathbf{x}_t^T(\mathbf{x}_0, \epsilon) = \sqrt{\alpha_t} \mathbf{x}_0 + \sigma_t \epsilon$. The parameters ϕ only appear in the student-generated samples \mathbf{x}_t^S . We need to compute ∇_{ϕ} of the terms involving \mathbf{x}_t^S . Let $L_1 = \|\mathbf{x}_t^S - \mathbf{x}_t^T\|^\beta$ and $L_2 = \|\mathbf{x}_t^{S1} - \mathbf{x}_t^{S2}\|^\beta$. The gradient of L_1 with respect to ϕ is (assuming $\beta \neq 0$ and $\mathbf{x}_t^S \neq \mathbf{x}_t^T$):

$$\begin{aligned} \nabla_{\phi} \|\mathbf{x}_t^S - \mathbf{x}_t^T\|^\beta &= \beta \|\mathbf{x}_t^S - \mathbf{x}_t^T\|^{\beta-2} (\mathbf{x}_t^S - \mathbf{x}_t^T)^\top (\nabla_{\phi} \mathbf{x}_t^S) \\ &= \beta \|\mathbf{x}_t^S - \mathbf{x}_t^T\|^{\beta-2} (\mathbf{x}_t^S - \mathbf{x}_t^T)^\top (\sqrt{\alpha_t} \nabla_{\phi} g_{\phi}(\mathbf{z}^S)). \end{aligned}$$

The gradient of L_2 with respect to ϕ (assuming $\mathbf{x}_t^{S1} \neq \mathbf{x}_t^{S2}$):

$$\begin{aligned} \nabla_{\phi} \|\mathbf{x}_t^{S1} - \mathbf{x}_t^{S2}\|^\beta &= \beta \|\mathbf{x}_t^{S1} - \mathbf{x}_t^{S2}\|^{\beta-2} (\mathbf{x}_t^{S1} - \mathbf{x}_t^{S2})^\top (\nabla_{\phi} (\mathbf{x}_t^{S1} - \mathbf{x}_t^{S2})) \\ &= \beta \|\mathbf{x}_t^{S1} - \mathbf{x}_t^{S2}\|^{\beta-2} (\mathbf{x}_t^{S1} - \mathbf{x}_t^{S2})^\top \\ &\quad (\sqrt{\alpha_t} \nabla_{\phi} g_{\phi}(\mathbf{z}^{S1}) - \sqrt{\alpha_t} \nabla_{\phi} g_{\phi}(\mathbf{z}^{S2})). \end{aligned}$$

In practice, these gradients are computed via automatic differentiation through the Monte Carlo estimators (Equation (9) or Equation (10)). The key observation is that $\nabla_{\phi} \mathcal{L}_{\text{VarFlow}}^{(\beta)}$ depends only on:

1. Samples $\mathbf{x}_t^S, \mathbf{x}_t^{S'}, \mathbf{x}_t^T$.
2. The Jacobian of the student generator, $\nabla_{\phi} g_{\phi}(\mathbf{z})$, which is readily available through back-propagation.

Crucially, the gradient computation for VarFlow does *not* require or involve the term $\nabla_{\mathbf{x}_t} \log p_{\phi,t}(\mathbf{x}_t)$. The VarFlow objective is formulated directly at the level of samples from $p_{\phi,t}(\mathbf{x}_t)$ and $q_t(\mathbf{x}_t)$, and its optimization relies on standard backpropagation through the generator g_{ϕ} that produces these samples. This completely sidesteps the need to estimate or approximate the intractable student marginal score, which is the primary source of complexity and potential issues in VSD’s gradient calculation. VarFlow’s gradient is therefore more direct and less prone to approximation errors inherent in estimating $s_{\phi,t}(\mathbf{x}_t)$.

F Limitations

The VarFlow objective, particularly the U-statistic estimator for the student-student interaction term $\mathbb{E}[\|\mathbf{x}_t^S - \mathbf{x}_t^{S'}\|^{\beta}]$ (Eq. (9)), involves $O(K^2)$ pairwise distance computations within the student batch of size K . Although our ablations demonstrate strong performance with moderate batch sizes (e.g., $K = 16$ or $K = 32$ per GPU) and the paired estimator (Eq. (10)) is used for the cross-term in our final setup, the student self-comparison term remains inherently quadratic in its full U-statistic form. This could present a computational bottleneck when scaling to extremely large batch sizes or in highly resource-constrained training environments, potentially necessitating careful batch size selection or exploration of more computationally efficient estimators for this term without sacrificing performance.