Path-Guided Flow Matching for Dataset Distillation

Xuhui Li[1*] **Zixu Wu**[1*] **Zhengquan Luo**[1] **Xiwei Liu**[1]
Yongqiang Yu[1] **Zixiang Hong**[2] **Zhiqiang Xu**[1✉]

[1]Mohamed bin Zayed University of Artificial Intelligence
Abu Dhabi, United Arab Emirates
[2]South China University of Technology
Guangzhou, China

{xuhui.li,zixu.wu,zhengquan.luo}@mbzuai.ac.ae
{xiwei.liu,yongqiang.yu,zhiqiang.xu}@mbzuai.ac.ae
202320116196@mail.scut.edu.cn

Abstract

Dataset distillation compresses large datasets into compact synthetic sets with comparable performance in training models. Despite recent progress on diffusion-based distillation, such methods typically rely on heuristic guidance or prototype assignment over long denoising chains, which increases sampling cost and makes prototype-consistent control harder under strong guidance or low IPC. We propose **Path-Guided Flow Matching (PGFM)**, the first flow matching-based framework for generative distillation, which enables deterministic synthesis by solving an ODE in a few steps. In particular, we introduce a retrieval-based prototype inversion stage that identifies prototype-consistent initial noises for class prototypes on the frozen flow manifold, and further develop an anchor-guided residual correction strategy for bounded stage-wise control. This design follows a controlled-transport principle: retrieval reduces initialization mismatch, while anchor guidance provides a bounded residual correction along the flow trajectory. Extensive experiments across high-resolution benchmarks demonstrate that PGFM matches or surpasses prior diffusion-based distillation approaches with fewer sampling steps while delivering competitive performance with improved efficiency.

1 Introduction

The rapid progress of machine learning has been driven by larger datasets and models, but this trend also makes training increasingly expensive in compute and storage. Beyond model-side efficiency techniques such as pruning [Liu et al., 2017, Ding et al., 2019] and quantization [Wu et al., 2016, Chen et al., 2021, Chauhan et al., 2023], an important alternative is to reduce the data required for training. Dataset distillation [Wang et al., 2018, Liu et al., 2022] does it by compressing a large training dataset into only a few synthetic samples per class (IPC), such that a model trained on the distilled dataset approaches the performance of the model trained on the full dataset.

Among this, diffusion-based generative models have emerged as a dominant approach for dataset distillation, with methods such as DiT [Peebles and Xie, 2023], MinimaxDiff [Gu et al., 2024], and MGD^3 [Chan-Santiago et al., 2025] achieving strong performance especially in the low-IPC regime. However, when used for prototype-guided distillation, diffusion-style sampling typically relies on a long denoising chain, where prototype consistency must be preserved across multiple iterative updates. This makes it harder to control how prototype guidance affects the final sample, since its influence is distributed across many denoising steps. Also, it often increases sampling cost due

*These authors contributed equally to this work ✉ Corresponding author

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

to repeated denoising steps and additional guidance heuristics. This raises a natural question: *Is there a generative paradigm that allows prototype guidance to be incorporated more directly and controllably, while also reducing the cost of synthesizing distilled datasets?*

Flow matching [Lipman et al., 2022, 2024, Liu, 2022] generates samples by integrating a learned velocity field from noise to data. This gives a direct way to study how an initial latent and an added guidance term affect the generated endpoint. For prototype-guided distillation, the two choices are closely related: an unguided endpoint far from its assigned prototype requires a larger correction to reach it. Section 5 bounds the endpoint displacement caused by the scaled guidance and gives a necessary condition for reaching a neighborhood of the assigned prototype.

To this end, we propose the first flow-matching method for dataset distillation, i.e., Path-Guided Flow Matching or **PGFM** for short. PGFM builds on a pretrained **G**aussian **M**ixture **Flow** matching (GM-Flow) generator [Chen et al., 2025a] and synthesizes samples through standard flow sampling. To cover class-specific regions under a limited IPC budget, PGFM clusters the real latents and assigns one prototype to each synthetic sample. It searches a noise bank for an initialization whose unguided endpoint is close to the assigned prototype. Slots that fail the retrieval threshold use warm-start fallback. For accepted retrievals, the stored trajectory supplies intermediate guidance targets before the final prototype target is used. Control scaling limits the guidance relative to the base velocity, and early stopping returns the later sampling stages to the base flow.

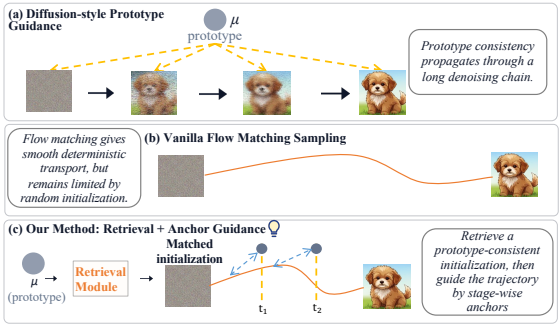


Figure 1: Motivation of our method from comparing different generative distillation paradigms.

Empirically, PGFM delivers substantial improvements over strong generative distillation baselines. On ImageNette, PGFM significantly outperforms the strongest baseline, MGD^3, with gains of $+5.1$, $+5.6$, and $+3.3$ points on ConvNet-6 at IPC $= 10, 20, 50$, respectively, while still achieving clear improvements on ResNet-18 and ResNet-AP. On ImageIDC, PGFM also yields consistent gains over the strongest baseline across all evaluation backbones. On the more challenging ImageNet100 benchmark, we find that unguided flow-matching sampling is already a surprisingly strong baseline, and PGFM further improves upon it and achieves the best performance across target backbones. At the same time, under matched performance, PGFM requires fewer FLOPs than both MinimaxDiffusion and MGD^3. Overall, these results show that flow matching, combined with retrieval-based initialization, warm start fallback and anchor-guided residual correction, can produce distilled datasets with stronger performance and better mode coverage at lower computational cost, especially in tight-budget settings.

We make the following key contributions in this work:

- We are the first to propose flow matching for dataset distillation as an efficient alternative to existing generative distillation paradigms.
- We combine retrieval-based initialization, warm-start fallback, and anchor guidance on a frozen flow generator, with control scaling and early stopping to limit the applied guidance. We derive an endpoint displacement bound and a necessary condition for reaching a prototype neighborhood to explain these design choices.
- Extensive experiments across high-resolution benchmarks show that PGFM improves the accuracy over state-of-the-art methods while substantially reducing computational costs.

2 Related Work

Optimization-based dataset distillation. Early dataset distillation methods optimize a small synthetic set by matching training signals from real data. Gradient matching [Zhao et al., 2021] aligns gradients between real and synthetic batches, but relies on expensive bi-level optimization and

scales poorly. To improve efficiency, Distribution Matching (DM) [Zhao and Bilen, 2023] matches feature statistics under randomly initialized networks, avoiding inner-loop training. Trajectory-based approaches further strengthen the signal by matching multi-step training dynamics, e.g., MTT [Cazenavette et al., 2022]. M3D [Zhang et al., 2024] formulates condensation into a min–max problem that reduces worst-case mismatch, while DSDM [Li et al., 2024] matches semantic distributions with explicit diversity regularization. More recently, NCFM [Wang et al., 2025] proposes a neural characteristic-function discrepancy from a min-max perspective to improve robustness.

Generative dataset distillation. Generative dataset distillation trains a generator to synthesize the entire distilled dataset, rather than optimizing each synthetic image individually [Zhang et al., 2023, Gu et al., 2024, Su et al., 2024]. Early work [Zhang et al., 2023] explored class-conditional GANs with learnable discrete latent representations. More recent methods largely adopt diffusion models as the backbone due to their high-fidelity synthesis. Representative diffusion-based approaches include MinimaxDiffusion [Gu et al., 2024], which fine-tunes a pretrained model with distribution-level objectives; D^4M [Su et al., 2024], which performs mode-aware sampling by injecting structured noisy modes; Influence-guided Diffusion (IGD) [Chen et al., 2025b], which uses influence estimates to guide sampling toward more training-beneficial samples; and MGD^3 [Chan-Santiago et al., 2025], which discovers class-conditional modes (e.g., via latent clustering) and applies mode guidance to improve coverage under tight budgets. However, diffusion-based distillation typically starts from arbitrary noise and only imposes prototype information during denoising, leaving the initialization–target coupling implicit. Flow matching instead provides deterministic noise-to-data transport, which enable us to achieve prototype consistent initialization and lightweight residual correction.

3 Preliminaries

3.1 Dataset Distillation

Given a large-scale dataset with training set $\mathcal{U} = \{(x_i, y_i)\}_{i=1}^{N_\mathcal{U}}$, the goal of dataset distillation (DD) is to build a smaller synthetic dataset $\mathcal{S} = \{(\tilde{x}_i, \tilde{y}_i)\}_{i=1}^{N_\mathcal{S}}$, where $N_\mathcal{S} \ll N_\mathcal{U}$ and $x_i, \tilde{x}_i$ are the original and synthetic samples[2] with corresponding class labels $y_i, \tilde{y}_i$. The essential requirement of DD lies in that the model $\phi_\mathcal{S}$ trained on the smaller synthetic dataset $\mathcal{S}$ should achieve similar test performance to that of a model $\phi_\mathcal{U}$ trained on the original set $\mathcal{U}$. That is, if $\mathcal{A}$ is the accuracy of a model on the test set $\mathcal{U}_e$, then $\mathcal{A}(\phi_\mathcal{U}) \sim \mathcal{A}(\phi_\mathcal{S})$. The size of the distilled dataset $N_\mathcal{S}$ is defined by the distillation budget, denoted as IPC (images per class). Prior generative models [Gu et al., 2024, Su et al., 2024, Zhang et al., 2023] conduct dataset distillation by approximating the dataset distribution through sampling diverse and representative instances. This line of research on *generative dataset distillation* tries to minimize the distribution gap which is often characterized by the following loss matching criterion:

$$\left\| \mathbb{E}_{x \sim P_{data}} \left[\ell(\phi_\mathcal{U}(x), y) \right] - \mathbb{E}_{x \sim P_{syn}} \left[\ell(\phi_\mathcal{S}(\tilde{x}), y) \right] \right\| < \epsilon,$$

where $P_{[\cdot]}$ is the underlying data distribution, ϵ is the error bound constant, and ℓ is a loss function. The use of generative models provides flexibility and privacy protection by not being limited to selecting original samples.

3.2 Latent Space Generative Modeling

To enable dataset distillation at higher resolutions, it is common to operate in the latent space of a pre-trained Variational Autoencoder (VAE) with a frozen encoder E and a frozen decoder D (e.g., adopted from Stable Diffusion). The encoder induces a perceptual latent space $\mathcal{Z}$ by mapping each image x to a posterior distribution $q_E(z \mid x)$. This defines an aggregated latent distribution over the dataset, as well as its class-conditional variant:

$$q_E(z) = \mathbb{E}_{x \sim P_{\text{data}}} [q_E(z \mid x)] ,$$
$$q_E(z \mid y) = \mathbb{E}_{x \sim P_{\text{data}}(\cdot \mid y)} [q_E(z \mid x)] .$$

In practice, the posterior mean is often used as a deterministic latent representation,

$$\bar{z}(x) = \mathbb{E}_{q_E(z|x)}[z],$$

[2]Throughout the paper, samples are images.

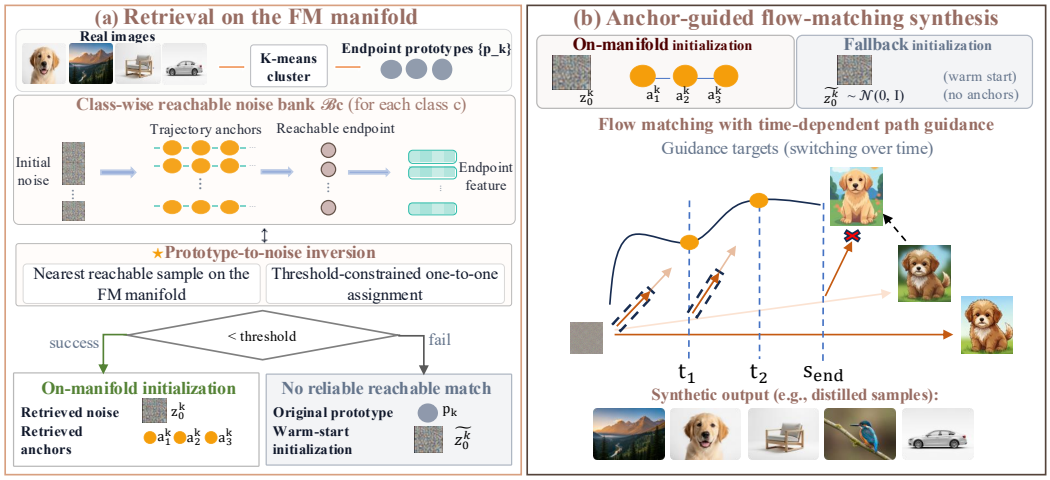


Figure 2: **Overview of PGFM:** (a) Retrieval stage, real images are clustered into endpoint prototypes, and a class-wise reachable noise bank is built via FM trajectories; a prototype-to-noise inversion retrieves the nearest reachable sample under a threshold constraint, yielding either on-manifold initialization (noise + anchors) or a fallback warm start. (b) Sampling stage, samples are generated by flow matching with time-dependent path guidance, where trajectories are sequentially steered by anchors and the final prototype, with additional stabilization (e.g., warm start and early stopping).

which yields, for each class y, an empirical latent set

$$\mathcal{Z}_y = \{\bar{z}(x_i) \,:\, y_i = y\},$$

providing a compact description of the class-conditional data distribution in latent space and allowing reconstruction back to the image space via the frozen decoder D.

3.3 Conditional Flow Matching (CFM) and ODE Sampling

Flow matching trains a continuous generative model by learning a conditional velocity field $u_\phi(t, z, y)$ that transports a simple distribution to the class-conditional data distribution along a continuous time path $t \in [0, 1]$. A common training objective is to regress the model velocity to a target velocity field $u^\star$ constructed from a coupling between $z_0 \sim p_0$ and $z_1 \sim p_1(\cdot \mid y)$:

$$\mathcal{L}(\phi) = \mathbb{E}_{y,\, t,\, z_t}\left[\lambda(t)\left\|u_\phi(t, z_t, y) - u^\star(t, z_t, y)\right\|_2^2\right].$$

After training, sampling boils down to solving an ordinary differential equation (ODE) from noise to data.

In this work, we do not train a flow model from scratch. Instead, we adopt a pre-trained flow-matching generator, i.e., GMFlow, to parameterize the class-conditional velocity field $u_\phi(t, z, y)$ in the VAE latent space. Given a class label y, we start from a base latent $z(0) = z_0$, where $z_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and generate a synthetic latent by integrating Eq. 8 to obtain $\tilde{z} = z(1)$. The synthetic image is then recovered via the frozen VAE decoder, $\tilde{X} = D(\tilde{z})$. In practice, we solve the ODE with efficient numerical solvers (e.g., Heun's method) [Hairer et al., 1993] using a small number of steps. Our proposed guidance is only applied during this sampling process, without modifying the pre-trained generator.

4 Proposed Method: PGFM

Given a training set $\mathcal{U} = \{(x_i, y_i)\}_{i=1}^{N_\mathcal{U}}$, our goal is to synthesize a compact distilled set with IPC samples per class. Our proposed Path-Guided Flow Matching (PGFM) performs this synthesis in the latent space of a frozen VAE using a pretrained conditional flow generator, enabling efficient generation of diverse and representative synthetic samples. The key design principle of PGFM is to select a prototype-specific initialization before applying guidance with controlled strength and

duration. Retrieval selects initial noises through their generated endpoints, with warm-start fallback for unsuccessful matches. Guidance then uses retrieved anchors or the endpoint prototype, with control scaling and early stopping.

The overall PGFM pipeline consists of i) latent-space preprocessing, ii) prototype retrieval on the frozen flow manifold, and iii) flow-matching sampling in latent space, to obtain the final synthetic dataset. Figure 2 illustrates the overall PGFM framework. Here, the frozen flow manifold refers informally to the reference trajectories and endpoints of the fixed, unguided flow model. PGFM keeps the generator and VAE parameters fixed and trains no additional network during image synthesis.

4.1 Latent Space Preprocessing

We start by moving dataset distillation from pixel space to a compact VAE latent space, which is more efficient and perceptually aligned. We therefore use a frozen, pre-trained VAE [Doersch, 2016] with encoder E and decoder D. Given the training dataset $\mathcal{U} = \{(x_i, y_i)\}_{i=1}^{N_\mathcal{U}}$, we encode each image into a latent representation $z_i = \mathrm{mean}(E(x_i))$, and obtain a latent dataset $\mathcal{V} = \{(z_i, y_i)\}_{i=1}^{N_\mathcal{U}}$. For each class y, we collect the empirical latent pool $\mathcal{Z}_y = \{z_i : y_i = y\}$, which serves as a discrete approximation to the class-conditional target distribution $p_1(z \mid y)$. In practice, we keep E and D frozen throughout and perform all prototype construction and flow sampling in this latent space. For numerical stability, we standardize latent data using their statistics: given all encoded latent data $\{z_i\}$, we compute the global mean μ and positive per-dimension standard deviations σ, and apply $\tilde{z} = (z - \mu)/\sigma$. Clustering and endpoint distances use these standardized coordinates. ODE states and guidance targets use the generator's coordinates.

To encourage broad coverage of the class-conditional latent space under a tight budget, we further perform K-means++ clustering [Bahmani et al., 2012] on each standardized pool $\{(z_i - \mu)/\sigma : y_i = y\}$ with $K = \mathrm{IPC}$ and use the resulting cluster centers $\{\mu_{y,k}\}_{k=1}^{K}$ as class-specific prototypes, which serve as target modes for the guided sampling stage described next.

For a fixed cluster, its centroid minimizes the sum of squared Euclidean distances to the cluster members. We use this representative as the target for the corresponding synthesis slot and examine the choice against the closest real latent in Table 11.

4.2 Prototype Retrieval

Cluster prototypes provide targets for generation, but an arbitrary initial noise may produce an endpoint far from its assigned prototype. We therefore search a finite bank of initial noises using their endpoints under the frozen generator.

For each class y, we first build a class-wise noise bank by sampling N_{bank} Gaussian noises $\{\epsilon_y^{(j)}\}_{j=1}^{N_{bank}}$, $\epsilon_y^{(j)} \sim \mathcal{N}(0, I)$, and rolling them forward through the frozen conditional flow model. Let $G_\phi(\epsilon, y)$ denote the numerical endpoint of the fixed unguided sampler from noise ϵ under condition y, expressed in standardized coordinates. We then obtain a bank of reachable latent endpoints

$$\mathcal{B}_y = \{(\epsilon_y^{(j)}, \hat{z}_y^{(j)})\}_{j=1}^{N_{bank}}, \qquad \hat{z}_y^{(j)} = G_\phi(\epsilon_y^{(j)}, y). \tag{1}$$

Given a prototype $\mu_{y,k}$, we retrieve its nearest reachable endpoint from the corresponding class-wise bank by

$$j^\star(y, k) = \arg \min_{1 \le j \le N_{bank}} d\Big(\hat{z}_y^{(j)}, \mu_{y,k}\Big), \tag{2}$$

where $d(\cdot, \cdot)$ denotes the latent-space distance metric. In practice, we use Euclidean distance in the standardized latent space. The retrieved initial noise for prototype $\mu_{y,k}$ is then

$$\epsilon_{y,k}^\star = \epsilon_y^{(j^\star(y,k))}, \qquad \hat{\mu}_{y,k} = \hat{z}_y^{(j^\star(y,k))}. \tag{3}$$

However, not every prototype admits a reliable nearby point on the frozen flow manifold. The selected bank endpoint may still be far from the assigned prototype. We accept it only when its discrepancy is below a threshold and use warm-start fallback otherwise. Specifically, let

$$\delta_{y,k} = d(\hat{\mu}_{y,k}, \mu_{y,k}) \tag{4}$$

denote the retrieval discrepancy for prototype $\mu_{y,k}$. We accept the retrieved noise only when

$$\delta_{y,k} \le \tau, \tag{5}$$

where τ is the acceptance threshold for the retrieved endpoint's distance to the assigned prototype. If (5) holds, we use $\epsilon^{\star}_{y,k}$ as the flow-consistent initialization associated with prototype $\mu_{y,k}$, and later we reuse the corresponding retrieved FM trajectory and record its intermediate latent states at predefined normalized progress values. These intermediate states are stored as trajectory anchors. Otherwise, we treat the retrieval as unsuccessful and fall back to the warm-start initialization.

Warm start. For slot (y, k), let $\xi_{y,k} \sim \mathcal{N}(0, I)$ be a fresh Gaussian draw. Write $\nu_{y,k} = \mu + \sigma \odot \mu_{y,k}$ for the assigned prototype in the generator's coordinates, where $\odot$ denotes elementwise multiplication. We define the fallback state by

$$z^{(y,k)}_{0,\mathrm{ws}} = (1 - \eta_{\mathrm{init}})\xi_{y,k} + \eta_{\mathrm{init}}\nu_{y,k}, \qquad 0 \le \eta_{\mathrm{init}} < 1. \tag{6}$$

The coefficient η_{init} controls the interpolation toward the assigned prototype. It is used in the fallback branch, with the dataset-specific values given in Appendix B. The actual initial condition is

$$z^{(y,k)}_0 = \begin{cases} \epsilon^{\star}_{y,k}, & \delta_{y,k} \le \tau, \\ z^{(y,k)}_{0,\mathrm{ws}}, & \delta_{y,k} > \tau. \end{cases} \tag{7}$$

4.3 Flow-Matching Sampling

PGFM synthesizes samples by integrating the pretrained GMFlow velocity field in the VAE latent space. For the base FM sampler, an initial latent $z_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is evolved by

$$\frac{dz(t)}{dt} = u_\phi\big(t, z(t), y\big), \qquad t \in [0, 1], \tag{8}$$

where u_ϕ is the class-conditional velocity field parameterized by GMFlow. In PGFM, sampling starts from the prototype-specific initialization $z^{(y,k)}_0$ in Eq. (7), replacing the random initialization used by the base FM model. In practice, we apply an explicit ODE solver (e.g., Euler/Heun) to obtain the final latent $\tilde{z} = z(1)$, which is decoded into the synthetic image $\tilde{x} = D(\tilde{z})$ using the frozen VAE decoder.

Trajectory anchor guidance. Instead of guiding the trajectory toward a single endpoint prototype during the whole ODE integration, we use *stage-wise trajectory anchor guidance*. For each assigned prototype $\mu_{y,k}$, if retrieval on the frozen flow manifold succeeds, we obtain a prototype-consistent initial latent and a sequence of trajectory anchors

$$\mathcal{A}_{y,k} = \{a^{(1)}_{y,k}, \ldots, a^{(A)}_{y,k}\},$$

extracted from the retrieved teacher trajectory at predefined progress stages. These anchors serve as intermediate targets before switching to the endpoint prototype. If retrieval fails, guidance uses the endpoint target $\nu_{y,k}$.

During sampling, we augment the pretrained flow field by

$$\frac{dz(t)}{dt} = u_\phi\big(t, z(t), y\big) \;+\; g(t)\, \tilde{u}_{\mathrm{guide}}\big(t, z(t), y, k\big), \tag{9}$$

where $g(t)$ is an early-stop gate and the guidance is computed from the predicted clean latent $\hat{z}_1(t)$:

$$\tilde{u}_{\mathrm{guide}}\big(t, z(t), y, k\big) = \alpha(t)\, u_{\mathrm{guide}}\big(t, z(t), y, k\big), \quad u_{\mathrm{guide}}\big(t, z(t), y, k\big) = \lambda\Big(T_{y,k}(t) - \hat{z}_1(t)\Big), \tag{10}$$

where $T_{y,k}(t)$ is a time-dependent target. The anchors and $\hat{z}_1(t)$ use the generator's coordinates. Let $s(t) \in [0, 1]$ denote the normalized sampling progress. For an accepted retrieval, define

$$T_{y,k}(t) = \begin{cases} a^{(m)}_{y,k}, & s(t) \in [\gamma_{m-1}, \gamma_m), \quad m = 1, \ldots, A, \\ \nu_{y,k}, & s(t) \in [\gamma_A, 1], \end{cases} \qquad \text{with } \gamma_0 = 0. \tag{11}$$

where $\gamma_1 < \cdots < \gamma_A \le s_{\mathrm{end}}$ are predefined stage boundaries. The gate sets the applied guidance to zero after s_{end}.

In our prototype guidance, we use an adaptive trust-region scaling

$$\alpha(t) = \min\left(1,\ \rho(t)\frac{\left\|u_\phi(t, z(t), y)\right\|_2}{\left\|u_{\text{guide}}(t, z(t), y, k)\right\|_2 + c}\right), \tag{12}$$

where $c > 0$ is a numerical stabilizer and $\rho(t) \geq 0$. This scaling bounds the guidance magnitude by $\rho(t)\|u_\phi(t, z(t), y)\|_2$. We linearly decay $\rho(t)$ from ρ_0 to $\rho_{\min}$ until s_{end}, and disable guidance afterwards using

$$g(t) = \mathbb{I}[s(t) \leq s_{\text{end}}]\,. \tag{13}$$

The target switches from retrieved anchors to the assigned prototype, and the gate turns off guidance in the later sampling stages. Together with control scaling, these choices set the direction, relative magnitude, and duration of the correction. Section 5 bounds the resulting endpoint displacement and gives a necessary condition for reaching the assigned prototype neighborhood.

5 Theoretical Analysis

PGFM selects an initial latent before applying guidance with limited magnitude and duration. We study how far this guidance can move the generated endpoint and what this implies for reaching the assigned prototype. For the k-th sample of class y, let $z(t)$ and $\bar{z}(t)$ be the continuous guided and unguided trajectories in Eqs. (9) and (8), both starting from $z_0^{(y,k)}$ in Eq. (7).

Proposition 1. *Suppose both trajectories are absolutely continuous on $[0, 1]$ and $u_\phi(t, \cdot, y)$ is L_u-Lipschitz, uniformly in t, on a region containing both trajectories. If $g(t)\rho(t)\|u_\phi(t, z(t), y)\|_2$ is integrable, then*

$$\|z(1) - \bar{z}(1)\|_2 \leq B_{y,k} := \int_0^1 e^{L_u(1-t)} g(t)\rho(t)\|u_\phi(t, z(t), y)\|_2\, dt. \tag{14}$$

The proof in Appendix A combines the control scaling rule with Grönwall's inequality. The scaling limits the applied guidance relative to the base velocity, and the gate sets it to zero when $s(t) > s_{\text{end}}$. The bound $B_{y,k}$ is evaluated along the guided trajectory $z(t)$.

Combining this bound with the triangle inequality gives, for any $r \geq 0$,

$$\|z(1) - \nu_{y,k}\|_2 \leq r \quad \Longrightarrow \quad \|\bar{z}(1) - \nu_{y,k}\|_2 \leq r + B_{y,k}. \tag{15}$$

Entering the prototype neighborhood requires enough displacement to cover the distance left by the unguided endpoint. The condition above compares that requirement with the displacement bound for the corresponding guided trajectory. This motivates searching generated endpoints for an initial noise before applying guidance.

The sampling ablation in Table 16 reports higher mean accuracy after adding early stopping and then trust-region scaling. Table 6 examines initialization with endpoint guidance fixed: F+E has higher mean accuracy than WS+E in all six settings. These comparisons support the use of initialization selection together with limited guidance.

6 Experiments

We evaluate the accuracy and computational cost of the distilled datasets under the stated training protocol. The component ablations compare initialization and guidance choices, while the supplementary evaluations examine prototype assignment.

6.1 Experimental Setup

Datasets and evaluation. We evaluate PGFM on two standard high-resolution dataset distillation benchmarks, ImageNette and ImageIDC, both at resolution 256×256. Following prior work, we consider IPC budgets $m \in \{10, 20, 50\}$. For each method, we synthesize a distilled set of m images per class, train target student networks from scratch on the distilled set, and report Top-1 test accuracy on the original benchmark test set.

Table 1: Performance comparison on ImageNette and ImageIDC datasets

Dataset	Backbone	IPC	Random	DM	DiT	Minimax	MGD^3	FM	PGFM
Nette	ConvNet-6	10	42.1 ± 0.5	48.5 ± 1.4	55.2 ± 2.3	56.8 ± 1.9	58.5 ± 2.1	60.5 ± 1.1	$\mathbf{63.6 \pm 1.3}$
		20	52.6 ± 0.9	55.2 ± 0.7	62.3 ± 1.0	63.9 ± 0.5	64.3 ± 1.1	65.3 ± 2.1	$\mathbf{69.9 \pm 1.1}$
		50	65.2 ± 1.2	70.4 ± 0.8	73.8 ± 0.6	75.4 ± 0.6	78.1 ± 2.1	77.7 ± 1.1	$\mathbf{81.4 \pm 1.0}$
	ResNet-18	10	48.4 ± 1.0	41.2 ± 1.0	59.5 ± 0.9	61.2 ± 0.4	62.8 ± 2.1	61.9 ± 0.7	$\mathbf{67.3 \pm 1.1}$
		20	58.7 ± 1.1	51.2 ± 0.6	63.0 ± 0.9	65.8 ± 1.6	68.1 ± 2.1	67.5 ± 1.1	$\mathbf{71.3 \pm 1.7}$
		50	71.3 ± 2.1	71.2 ± 0.8	72.7 ± 0.9	74.5 ± 0.6	79.1 ± 1.3	75.1 ± 1.1	$\mathbf{81.2 \pm 1.3}$
	ResNet-AP	10	48.2 ± 1.2	40.5 ± 0.9	56.1 ± 0.8	58.8 ± 1.0	65.8 ± 2.4	62.4 ± 1.3	$\mathbf{68.2 \pm 1.1}$
		20	55.7 ± 1.1	52.8 ± 0.4	63.2 ± 0.7	65.1 ± 0.9	70.4 ± 0.8	65.7 ± 1.1	$\mathbf{72.6 \pm 1.1}$
		50	68.4 ± 1.0	70.8 ± 0.2	72.3 ± 0.5	74.0 ± 0.7	79.1 ± 1.5	73.6 ± 0.1	$\mathbf{79.9 \pm 1.5}$
IDC	ConvNet-6	10	41.7 ± 1.1	46.7 ± 0.8	47.7 ± 1.2	47.2 ± 0.9	52.8 ± 1.5	53.1 ± 1.2	$\mathbf{53.9 \pm 1.1}$
		20	47.0 ± 1.0	52.6 ± 0.9	53.2 ± 1.1	53.2 ± 0.8	54.0 ± 1.2	57.2 ± 1.1	$\mathbf{58.8 \pm 1.2}$
		50	64.7 ± 1.3	68.2 ± 1.0	63.3 ± 0.9	68.2 ± 0.7	71.1 ± 1.5	67.4 ± 1.3	$\mathbf{75.1 \pm 1.2}$
	ResNet-18	10	44.9 ± 1.2	52.4 ± 0.9	51.3 ± 1.1	50.7 ± 0.8	51.6 ± 1.8	52.4 ± 0.7	$\mathbf{54.7 \pm 1.2}$
		20	49.8 ± 1.1	55.7 ± 1.0	56.3 ± 1.2	56.3 ± 0.7	58.7 ± 1.5	57.7 ± 1.4	$\mathbf{61.1 \pm 1.0}$
		50	65.5 ± 1.4	69.1 ± 1.1	64.1 ± 0.9	69.1 ± 0.8	71.0 ± 1.3	66.7 ± 1.0	$\mathbf{72.3 \pm 1.2}$
	ResNet-AP	10	47.1 ± 1.3	52.4 ± 0.5	54.1 ± 0.4	53.1 ± 0.2	54.6 ± 2.1	52.3 ± 0.5	$\mathbf{55.2 \pm 0.6}$
		20	52.1 ± 0.9	58.5 ± 0.4	58.9 ± 0.2	59.0 ± 0.4	61.9 ± 0.9	58.3 ± 1.3	$\mathbf{62.3 \pm 1.2}$
		50	66.1 ± 1.8	69.1 ± 0.8	64.3 ± 0.6	69.6 ± 0.2	72.0 ± 0.8	67.7 ± 0.9	$\mathbf{72.3 \pm 1.0}$

Table 2: Performance comparison on ImageNet100.

	IPC = 10 (0.8%)			IPC = 20 (1.6%)		
	ConvNet-6	ResNet-18	ResNetAP-10	ConvNet-6	ResNet-18	ResNetAP-10
Random	17.0 ± 0.3	19.1 ± 0.4	17.5 ± 0.5	24.8 ± 0.2	26.7 ± 0.5	25.5 ± 0.3
Herding	17.2 ± 0.3	19.8 ± 0.3	16.1 ± 0.2	24.3 ± 0.4	27.6 ± 0.1	24.7 ± 0.1
IDC-1	24.3 ± 0.5	25.7 ± 0.1	25.1 ± 0.2	28.8 ± 0.3	29.9 ± 0.2	30.2 ± 0.2
MinMaxDiff	22.3 ± 0.5	24.8 ± 0.2	22.5 ± 0.3	29.3 ± 0.4	32.3 ± 0.1	31.2 ± 0.1
MGD^3	23.4 ± 0.9	23.8 ± 0.5	25.7 ± 0.7	30.2 ± 0.5	32.0 ± 1.4	33.1 ± 0.5
FM	24.9 ± 0.4	24.8 ± 0.7	25.5 ± 0.4	30.6 ± 0.2	32.0 ± 0.4	31.5 ± 0.6
PGFM (Ours)	$\mathbf{25.9} \pm 0.4$	$\mathbf{26.8} \pm 0.4$	$\mathbf{27.6} \pm 0.3$	$\mathbf{31.7} \pm 0.2$	$\mathbf{32.9} \pm 0.5$	$\mathbf{34.3} \pm 0.5$
Full	79.9 ± 0.4	80.3 ± 0.2	81.8 ± 0.7	79.9 ± 0.4	80.3 ± 0.2	81.8 ± 0.7

Baselines. We compare PGFM with several representative dataset distillation baselines, including DM [Zhao and Bilen, 2023], DiT [Peebles and Xie, 2023], MinimaxDiff [Gu et al., 2024], and MGD^3 [Chan-Santiago et al., 2025]. As the flow-matching baseline, we use the same pretrained GMFlow [Chen et al., 2025a] backbone as in our method and perform vanilla flow matching sampling without retrieval or anchor guidance. We evaluate all methods on three target student architectures, namely ConvNet-6, ResNet-18, and ResNet-AP, to assess the generalization ability of the distilled datasets across different model capacities. Results are reported as mean $\pm$ standard deviation over repeated training runs under the same evaluation protocol.

Implementation details. Our pretrained generator $\mathcal{G}$ is GMFlow trained on ImageNet, and all experiments are conducted at resolution 256×256. Unless otherwise specified, we use 48 flow-matching inference steps with 4 substeps per step. For PGFM, the default retrieval setting uses a class-wise noise bank of size 64, one anchor at normalized progress 0.4, and early-stopped guided sampling with $s_{\text{end}} = 0.6$. The trust-region coefficient decays linearly from $\rho_0 = 0.5$ to $\rho_{\min} = 0.1$. All experiments are conducted on NVIDIA GeForce RTX 4090 GPUs (24GB memory each, CUDA 12.9). Additional implementation details and hyperparameter settings are provided in Appendix B.

6.2 Result Analysis

As shown in Table 1, PGFM consistently outperforms prior diffusion-based distillation baselines on both ImageNette and ImageIDC across backbones, with the clearest gains under tight budgets, i.e., low IPC. On ImageNette, PGFM improves over the strongest diffusion baseline MGD^3 by up to $+5.1$ points on ConvNet-6 at IPC $= 10$ and $+3.2$ points on ResNet-18 at IPC $= 20$, while remaining competitive and often superior at IPC $= 50$. On ImageIDC, the same pattern holds: FM sampling is already strong, and PGFM further improves over MGD^3 in most settings, with gains of $+1.1/+3.1/+0.6$ points at IPC $= 10$ and $+4.8/+2.4/+0.4$ points at IPC $= 20$ on

Table 3: Model size comparison and end-to-end matched performance comparison on ImageNette. ↓ means less is better.

Metric	MinimaxDiff	MGD^3	Ours
Params (M) ↓	674.93	675.24	**542.77**
IPC	20	19	**11**
FLOPs (T) ↓	4784	3114	**2201**
Efficiency Gain	1.0×	1.5×	**2.2**×

Table 4: Computational complexity comparison on sampling stage, with ImageNette dataset. NFE: Number of Function Evaluations.

Metric	MiniMaxDiff	MGD^3	PGFM
Time / image (s) ↓	1.019	0.924	**0.45**
Steps (NFE) ↓	50	50	**48**
FLOPs (T) ↓	23.73	17.30	**4.27**
Efficiency Gain	1.0×	1.4×	**5.5**×

Table 5: Backbone generalization study on SelfFlow. We compare the original SelfFlow baseline and SelfFlow equipped with PGFM. Results are reported as Top-1 accuracy (%).

Dataset	Method	IPC=10			IPC=20			IPC=50		
		ConvNet	ResNet18	ResNet-AP	ConvNet	ResNet18	ResNet-AP	ConvNet	ResNet18	ResNet-AP
ImageNette	SelfFlow	49.4±0.9	54.8±0.4	55.5±0.9	55.5±0.9	61.5±0.5	65.6±1.4	72.7±0.7	73.4±1.3	77.5±1.0
	SelfFlow+PGFM	**59.7±0.7**	**64.0±0.3**	**63.7±1.0**	**65.5±0.7**	**67.9±0.1**	**70.7±0.1**	**77.2±1.4**	**75.4±0.6**	**77.5±0.7**
ImageIDC	SelfFlow	43.1±0.4	44.9±0.4	47.6±1.1	45.8±2.0	47.5±1.1	52.7±2.1	63.6±0.7	67.6±0.6	65.8±0.9
	SelfFlow+PGFM	**49.7±1.5**	**49.7±0.3**	**50.1±0.7**	**53.2±0.6**	**59.1±0.7**	**57.3±0.1**	**68.2±1.4**	**68.9±0.4**	**65.8±0.6**

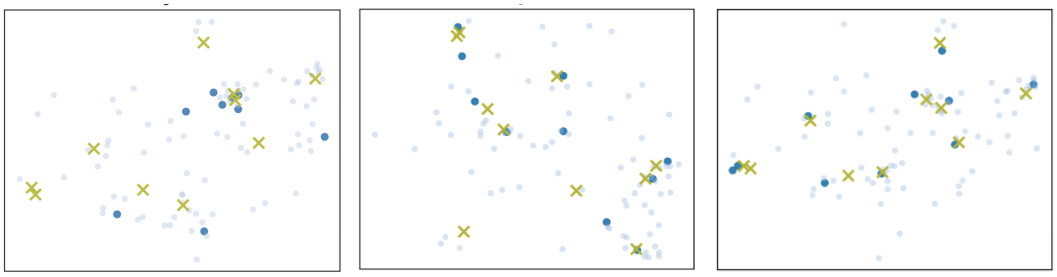
ConvNet-6/ResNet-18/ResNet-AP, respectively. This trend highlights the main advantage of PGFM: by explicitly searching the initial point and guiding flow trajectories toward diverse prototypes, it enables a compact synthetic set to better cover the critical modes of the data distribution.

ImageNet100. Table 2 reports results on the more challenging ImageNet100 benchmark. When IPC $= 10$, pure flow-matching sampling (FM, no guidance) already improves over strong diffusion baselines (e.g., beating MGD^3 by $+1.5$ and $+1.0$ points on ConvNet-6 and ResNet-18, respectively), suggesting that flow-based sampling produces latents that are inherently closer to the class-conditional manifold and thus more "trainable" under scarce budgets. With this strong starting point, PGFM further boosts FM by $+1.0$, $+2.0$, and $+2.1$ points across the three backbones, respectively, supporting our core idea that aligning initialization with the pretrained flow manifold is effective, and trajectory guidance is useful for correcting residual retrieval mismatch. At IPC $= 20$, PGFM remains consistently better than both FM and MGD^3, which achieves the largest improvement $+2.8$ on ResNetAP-10 (see Appendix C.1 for imagenet1k experiments).

Computational cost Table 3 reports both the model size and the matched-performance complexity on ImageNette. Our model is more compact, using 542.77M parameters compared with about 675M for the diffusion-based baselines. To further compare efficiency under similar performance levels, we report IPC and end-to-end computational cost. Our method at IPC$= 11$ already achieves accuracy comparable to MinimaxDiff at IPC$= 20$ and MGD^3 at IPC$= 19$. Under this setting, our method requires the lowest end-to-end FLOPs, reducing the total cost from 4784 and 3114 to 2201. These results suggest that PGFM offers a more efficient solution overall.

Sampling stage cost Table 4 reports a direct comparison of the sampling-stage cost in PGFM against state-of-the-art diffusion-based distillation methods. PGFM achieves the lowest wall-clock time per image (0.45s vs. 1.019s and 0.924s) while using slightly fewer function evaluations. It also reduces the sampling cost to only 4.27 TFLOPs per image, compared with 23.73 for MiniMaxDiff and 17.30 for MGD^3. This results in a substantial $5.5\times$ efficiency gain relative to MiniMaxDiff, demonstrating that flow matching provides a substantially lighter sampling interface than diffusion-based methods for generative distillation.

Different backbones To verify the generalization of our distillation method, we conduct additional experiments by replacing GMFlow with another pretrained flow matching backbone, SelfFlow [Chefer et al., 2026]. As shown in Table 5, PGFM consistently improves over vanilla SelfFlow across different settings . These results demonstrate that the improvement comes from the proposed retrieval-based initialization and anchor-guided trajectory correction, rather than only from the strength of a specific pretrained generator.



(a) Baseline FM (b) MGD^3 (c) PGFM (Ours)

Figure 3: **t-SNE Analysis.** Visual comparison of latent distributions. (a) Generated by FM baseline. (b) Generated by MGD^3. (c) Generated by our PGFM. (● Synthetic image, ● Real image, × Prototypes)

Table 6: Ablation on retrieval-based initialization and guidance design. FM: vanilla flow matching baseline; WS: warm start only; R: retrieval only; F: Fallback mechanism with retrieval and warm start; WS + E: warm start + endpoint guidance; F + E: Fallback + endpoint guidance; Full: Fallback + anchor guidance.

Dataset	IPC	Backbone	FM	WS	R	F	WS + E	F + E	Full
ImageNette	10	ConvNet-6	60.5 ± 1.1	61.6 ± 1.4	63.0 ± 1.2	64.3 ± 0.4	62.1 ± 0.1	$\mathbf{64.4 \pm 0.8}$	63.6 ± 1.1
		ResNet-18	61.9 ± 0.7	61.5 ± 0.7	64.6 ± 1.5	65.9 ± 1.8	64.1 ± 0.1	66.2 ± 1.1	$\mathbf{66.7 \pm 1.1}$
		ResNet-AP	62.4 ± 1.3	62.5 ± 3.0	65.6 ± 1.6	66.3 ± 1.3	63.5 ± 0.4	67.5 ± 1.0	$\mathbf{68.2 \pm 1.1}$
ImageIDC	10	ConvNet-6	53.1 ± 1.2	52.7 ± 1.2	53.4 ± 1.4	53.8 ± 0.6	53.2 ± 1.0	53.7 ± 1.0	$\mathbf{53.9 \pm 1.1}$
		ResNet-18	52.4 ± 0.7	51.9 ± 0.9	52.7 ± 1.3	53.0 ± 1.3	52.9 ± 0.9	53.4 ± 0.3	$\mathbf{54.7 \pm 1.1}$
		ResNet-AP	52.3 ± 0.5	51.8 ± 0.8	52.4 ± 1.7	52.7 ± 0.7	52.5 ± 0.8	52.8 ± 1.4	$\mathbf{55.2 \pm 1.1}$

6.3 Ablation Study

Coverage analysis. Beyond downstream accuracy, we examine the distribution of synthetic samples in latent space. To provide a qualitative visualization of diversity coverage in latent space, we plot the distilled datasets of MGD^3, FM, and PGFM using t-SNE. Figure 3 suggests that the unguided FM baseline covers only part of the real-data manifold, while MGD^3 improves coverage but still shows visible mismatch between some synthetic samples and their intended prototype regions. PGFM, in comparison, yields embeddings that are more consistently distributed around the target prototype regions and better overlap with the real-data features. This suggests that our path-guided control provides a more reliable mechanism for mode assignment and coverage (For more coverage analysis, please refer to Appendix C.1).

Component analysis. Table 6 compares initialization and guidance choices. Warm start gives limited gains over vanilla FM, while retrieval and fallback improve student accuracy. With fallback retained, anchor guidance gives higher mean accuracy than endpoint guidance in five of the six settings, with ImageNette on ConvNet-6 as the exception. Further ablations appear in Appendix C.2.

7 Conclusion

In this paper, we present PGFM, a flow-matching framework for generative dataset distillation. Our method exploits deterministic flow transport as a controllable sampling interface, and combines retrieval-based initialization, warm start fallback mechanism, and anchor-guided residual correction to improve prototype consistency under low IPC settings. Experiments on high-resolution benchmarks show that PGFM consistently outperforms strong diffusion-based baselines. These results suggest that flow matching is not only an efficient alternative to diffusion for generative distillation, but also a promising foundation for prototype-guided sample synthesis.

References

Marcel R Ackermann, Johannes Blömer, Daniel Kuntze, and Christian Sohler. Analysis of agglomerative clustering. *Algorithmica*, 69(1):184–215, 2014.

Bahman Bahmani, Benjamin Moseley, Andrea Vattani, Ravi Kumar, and Sergei Vassilvitskii. Scalable k-means++. *arXiv preprint arXiv:1203.6402*, 2012.

George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A. Efros, and Jun-Yan Zhu. Dataset distillation by matching training trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4750–4759, 2022.

Jeffrey A Chan-Santiago, Praveen Tirupattur, Gaurav Kumar Nayak, Gaowen Liu, and Mubarak Shah. Mgd ^{3}: Mode-guided dataset distillation using diffusion models. *arXiv preprint arXiv:2505.18963*, 2025.

Arun Chauhan, Utsav Tiwari, et al. Post training mixed precision quantization of neural networks using first-order information. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1343–1352, 2023.

Hila Chefer, Patrick Esser, Dominik Lorenz, Dustin Podell, Vikash Raja, Vinh Tong, Antonio Torralba, and Robin Rombach. Self-supervised flow matching for scalable multi-modal synthesis. *arXiv preprint arXiv:2603.06507*, 2026.

Hansheng Chen, Kai Zhang, Hao Tan, Zexiang Xu, Fujun Luan, Leonidas Guibas, Gordon Wetzstein, and Sai Bi. Gaussian mixture flow matching models. *arXiv preprint arXiv:2504.05304*, 2025a.

Mingyang Chen, Jiawei Du, Bo Huang, Yi Wang, Xiaobo Zhang, and Wei Wang. Influence-guided diffusion for dataset distillation. In *The Thirteenth International Conference on Learning Representations*, 2025b.

Weihan Chen, Peisong Wang, and Jian Cheng. Towards mixed-precision quantization of neural networks via constrained optimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5350–5359, 2021.

Xiaohan Ding, Guiguang Ding, Yuchen Guo, and Jungong Han. Centripetal sgd for pruning very deep convolutional networks with complicated structure. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4943–4953, 2019.

Carl Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.

Jianyang Gu, Saeed Vahidian, Vyacheslav Kungurtsev, Haonan Wang, Wei Jiang, Yang You, and Yiran Chen. Efficient dataset distillation via minimax diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Ernst Hairer, Gerhard Wanner, and Syvert P Nørsett. *Solving ordinary differential equations I: Nonstiff problems*. Springer, 1993.

Hongcheng Li, Yucan Zhou, Xiaoyan Gu, Bo Li, and Weiping Wang. Diversified semantic distribution matching for dataset distillation. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 7542–7550, 2024.

Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. The global k-means clustering algorithm. *Pattern recognition*, 36(2):451–461, 2003.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky TQ Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024.

Qiang Liu. Rectified flow: A marginal preserving approach to optimal transport. *arXiv preprint arXiv:2209.14577*, 2022.

Songhua Liu, Kai Wang, Xingyi Yang, Jingwen Ye, and Xinchao Wang. Dataset distillation via factorization. *Advances in Neural Information Processing Systems*, 35:1100–1113, 2022.

Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE international conference on computer vision*, pages 2736–2744, 2017.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

Tian Qin, Zhiwei Deng, and David Alvarez-Melis. A label is worth a thousand images in dataset distillation. *Advances in Neural Information Processing Systems*, 37:131946–131971, 2024.

Carl Rasmussen. The infinite gaussian mixture model. *Advances in neural information processing systems*, 12, 1999.

Duo Su, Junjie Hou, Weizhi Gao, Yingjie Tian, and Bowen Tang. Dˆ 4: Dataset distillation via disentangled diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5809–5818, 2024.

Shaobo Wang, Yicun Yang, Zhiyuan Liu, Chenghao Sun, Xuming Hu, Conghui He, and Linfeng Zhang. Dataset distillation with neural characteristic function: A minmax perspective. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 25570–25580, 2025.

Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018.

Jiaxiang Wu, Cong Leng, Yuhang Wang, Qinghao Hu, and Jian Cheng. Quantized convolutional neural networks for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4820–4828, 2016.

David Junhao Zhang, Heng Wang, Chuhui Xue, Rui Yan, Wenqing Zhang, Song Bai, and Mike Zheng Shou. Dataset condensation via generative model. *arXiv preprint arXiv:2309.07698*, 2023.

Hansong Zhang, Shikun Li, Pengju Wang, Dan Zeng, and Shiming Ge. M3d: Dataset condensation by minimizing maximum mean discrepancy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 9314–9322, 2024.

Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6514–6523, 2023.

Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. In *Ninth International Conference on Learning Representations 2021*, 2021.

Appendix Contents

A Proof of Proposition 1 **14**

B Hyperparameter Setting **14**

C Extensive Experiments **14**

C.1 General Experiments 15

C.2 Further Ablation Study 16

D Visualization **18**

D.1 Effect of Trust Region ρ 18

D.2 Synthetic Images 18

D.3 Flow Matching versus Diffusion 18

E Limitations and Future Work **19**

F GenAI Usage Disclosure **19**

A Proof of Proposition 1

Proof. Equations (12) and (13) give

$$\|g(t)\tilde{u}_{\text{guide}}(t, z(t), y, k)\|_2 \le g(t)\rho(t)\|u_\phi(t, z(t), y)\|_2.$$

The two trajectories have the same initial state. Subtracting their integral equations and using the Lipschitz condition yields

$$\|z(t) - \bar{z}(t)\|_2 \le L_u \int_0^t \|z(s) - \bar{z}(s)\|_2 \, ds + \int_0^t g(s)\rho(s)\|u_\phi(s, z(s), y)\|_2 \, ds.$$

Grönwall's inequality gives

$$\|z(t) - \bar{z}(t)\|_2 \le \int_0^t e^{L_u(t-s)} g(s)\rho(s)\|u_\phi(s, z(s), y)\|_2 \, ds.$$

Setting $t = 1$ proves Eq. (14). If $\|z(1) - \nu_{y,k}\|_2 \le r$, then

$$\|\bar{z}(1) - \nu_{y,k}\|_2 \le \|\bar{z}(1) - z(1)\|_2 + \|z(1) - \nu_{y,k}\|_2 \le B_{y,k} + r,$$

which proves Eq. (15). □

For the decaying schedule in Section 4, $g(t)\rho(t) \le \rho_0 g(t)$, so

$$B_{y,k} \le e^{L_u}\rho_0 \int_0^1 g(t)\|u_\phi(t, z(t), y)\|_2 \, dt.$$

The integral accumulates the base speed along the guided trajectory while guidance is active.

For numerical endpoints in the same coordinates, add the terminal errors of the two solver runs to $B_{y,k}$ in the displacement bound and the necessary condition.

B Hyperparameter Setting

Our pre-trained model $\mathcal{G}$ is `gmflow_imagenet_k8_ema` trained on ImageNet. We set the classifier-free guidance scale to 0.3. For the flow matching ODE solver, $N = 48$ inference steps with 4 intermediate sub-steps are used to ensure high trajectory precision while maintaining computational efficiency. The latent prototypes are initialized using K-means++ clustering by default, where we employ cluster centroids as guidance targets with convergence tolerance 10^{-4} and up to 300 iterations.

For retrieval-based initialization, we construct a class-wise noise bank with 64 candidate noises per class by default, generate their corresponding latent endpoints using the same 48-step FM sampler, and perform nearest-neighbor retrieval in the latent space under a 70% threshold. Retrieval is enforced with unique prototype-to-noise assignment. If no reliable retrieved initialization is found, we fall back to the original warm-start strategy. In the fallback branch, Eq. (6) interpolates the fresh Gaussian draw $\xi_{y,k}$ toward $\nu_{y,k}$. We set η_{init} by dataset and IPC. Specifically, for ImageNette, we set $\eta_{\text{init}} \in \{0.09, 0.05, 0.01\}$ for IPCs $\{10, 20, 50\}$, respectively. For the more challenging ImageIDC and ImageNet-100 datasets, we use smaller bias values of $\{0.04, 0.02, 0.01\}$ for the corresponding IPC settings.

For trajectory anchor guidance, we use one intermediate anchor by default at normalized progress $s = 0.4$, and stop guidance at $s_{\text{end}} = 0.6$. The path guidance strength is set to $\lambda = 0.5$. Finally, the trust-region scaling is controlled by an initial radius $\rho_0 = 0.5$ and a terminal radius $\rho_{\min} = 0.1$, with linear decay over the guided phase.

C Extensive Experiments

In this section, we provide experimental results beyond the main paper to further validate the performance of our method. We organize these experiments into two parts. First, we present a set of general supplementary evaluations. Second, we conduct a comprehensive ablation study to examine the contribution of each design component.

Table 7: ImageNet-1K accuracy (%) under IPC $= 10$.

Method	ConvNet-6	ResNet18	ResNet-AP	ResNet-50	ResNet-101
MiniMaxDiffusion	11.6	17.7	16.6	19.2	18.4
FM	12.9	19.8	19.4	22.5	22.1
MGD3	13.6	21.5	20.2	21.4	20.5
PGFM	**14.1**	**22.8**	**21.3**	**23.8**	**22.3**

Table 8: Per-class Hit Rate (%) comparison on ImageNette (IPC $= 10$). PGFM achieves significantly higher coverage across all distinct semantic classes.

Class ID	**Class Name**	**FM**	**MGD^3**	**PGFM (Ours)**
n01440764	Tench	10.0	10.0	**80.0**
n02102040	English Springer	20.0	30.0	**70.0**
n02979186	Cassette Player	0.0	20.0	**50.0**
n03000684	Chainsaw	10.0	30.0	**70.0**
n03028079	Church	20.0	20.0	**80.0**
n03394916	French Horn	10.0	40.0	**90.0**
n03417042	Garbage Truck	0.0	40.0	**70.0**
n03425413	Gas Pump	10.0	10.0	**90.0**
n03445777	Golf Ball	0.0	60.0	**100.0**
n03888257	Parachute	20.0	40.0	**80.0**
Average	–	**10.0**	**30.0**	**78.0**

C.1 General Experiments

ImageNet1k. Table 7 reports additional ImageNet-1K results at $\mathrm{IPC} = 10$ under an image-to-image setting. Different from prior diffusion-based work that evaluates ImageNet-1K in a text-to-image setting, our pretrained GMFlow backbone only supports image-conditioned generation. We therefore use this experiment to validate the scalability of PGFM on ImageNet-1K.

From the experimental results, we can find that even without diffusion-style sampling, pure flow-matching (FM) consistently outperforms MiniMaxDiffusion across all backbones, with gains of $+1.3$, $+2.1$, and $+2.8$ points, respectively. MGD3 improves over FM on ResNet18 but remains close on ConvNet-6 and ResNet-AP. Our method, PGFM, achieves the best performance on all three architectures, which corresponds to improvements over MGD3 of $+0.5$, $+1.3$, and $+1.1$ points, respectively. These results indicate that path-guided flow sampling is particularly effective in the low-IPC regime on large-scale datasets.

Complex models. Table 7 shows that under IPC $= 10$ on ImageNet-1K, PGFM achieves the best accuracy across ResNet-18/50/101, surpassing MGD3 by a clear margin. Moreover, the unguided FM baseline is already competitive with diffusion baselines (e.g., on ResNet-50/101), suggesting that flow-matching sampling produces more transferable and trainable synthetic data under tight budgets, and PGFM further improves with lightweight prototype-guided coverage.

Details of hit rate Table 8 shows the detailed hit rate of each class when comparing with FM and MGD^3, where hit rate measures the fraction of generated samples that successfully hit their assigned prototypes within a predefined tolerance. Our method, PGFM, can significantly increase the coverage, demonstrating that our method is better aligned with the prototypes to achieve the diversity.

Representativeness and coverage. We visualize the trade-off between representativeness and coverage in Figure 4: Representativeness is measured as the average Euclidean distance to the 50 nearest neighbors in the standardized VAE latent space of the real training set and converting it into a score, while coverage quantifies the diversity. Higher values indicate better performance for both measurements. The region in the top-right corner denotes the Optimal balance. We can see that PGFM achieves significantly superior coverage, consistently populating the upper region of the

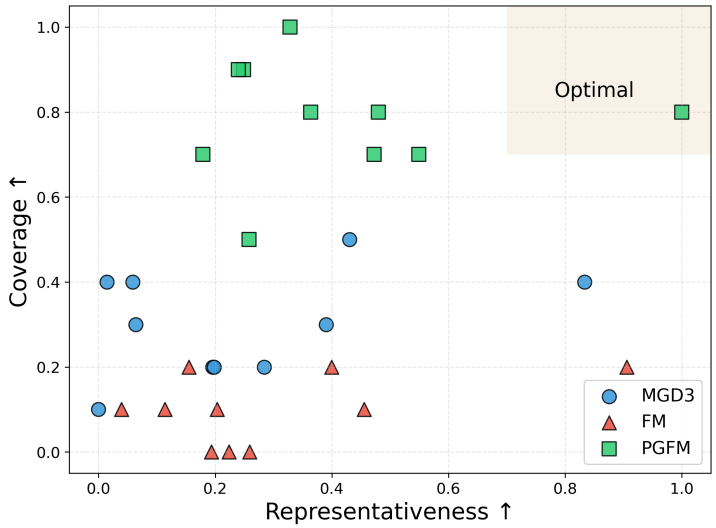


Figure 4: Representativeness and coverage analysis

Table 9: Comparison under soft-label settings. Results are reported as Top-1 accuracy (%).

Dataset	Method	IPC=10			IPC=20			IPC=50		
		ConvNet	ResNet18	ResNet-AP	ConvNet	ResNet18	ResNet-AP	ConvNet	ResNet18	ResNet-AP
ImageIDC	Random	49.2±0.3	49.8±1.7	52.5±0.7	56.2±0.8	52.9±1.0	54.4±1.7	63.9±0.7	61.4±0.6	61.5±0.7
	Baseline	53.4±1.3	51.6±2.0	52.7±1.4	56.6±2.4	55.4±1.8	57.5±1.0	61.3±2.1	59.1±1.7	59.3±0.7
	MGD3	50.8±1.4	51.7±1.0	50.9±1.1	54.4±0.7	53.1±0.9	54.5±1.0	61.4±1.0	61.5±1.1	61.5±0.8
	GMFlow	**53.5±1.5**	**54.4±0.4**	**55.2±0.9**	**57.1±0.7**	**58.8±0.4**	**59.1±0.5**	**63.3±0.7**	**63.8±0.8**	**64.1±1.3**
ImageNette	Random	51.6±1.7	53.0±2.0	54.5±2.1	58.9±2.7	60.2±1.6	60.7±1.5	69.9±2.1	71.5±2.4	69.8±1.6
	Baseline	55.4±1.4	55.1±1.3	53.8±0.8	54.8±2.5	56.8±1.1	57.2±1.3	66.6±1.6	66.6±1.6	63.8±1.1
	MGD3	52.3±1.0	56.4±1.2	56.4±1.3	60.2±0.6	59.4±1.3	59.2±1.5	70.2±0.3	66.6±1.6	62.8±0.7
	GMFlow	**59.6±0.6**	**60.5±1.3**	**58.4±0.3**	**62.0±0.7**	**62.5±0.1**	**61.0±1.4**	**71.8±0.6**	**68.7±0.7**	**67.3±1.0**

plot. Meanwhile, PGFM demonstrates improved representativeness, with several class prototypes successfully landing in or near the optimal region, outperforming MGD^3 and FM.

Soft-label test To better understand the role of soft labels in generative dataset distillation, we conduct two additional sets of experiments following the official [Qin et al., 2024] implementation. Table 9 evaluates student training with entirely soft labels, while table 10 evaluates a mixed setting where 50% hard labels and 50% soft labels are used. All expert models for generating soft labels are trained based on official code repository [Qin et al., 2024].

From these results, we observe that soft labels provide additional supervision and can improve the performance when combined with hard labels. However, training exclusively with soft labels is generally inferior to the hard-label setting, indicating that soft-label information and synthetic image quality are complementary factors. These experiments demonstrate that PGFM focuses on improving the image synthesis axis, while soft-label optimization can serve as an orthogonal enhancement for future generative distillation frameworks.

C.2 Further Ablation Study

We provide further ablation study to thoroughly analyze each components in our method. The analysis will be divided by different process.

C.2.1 Prototype Construction

Cluster closest point Table 11 shows that using closest point instead of the cluster center will significantly reduce the performance. This is because it turns a stable mode representative (the centroid) into a single, noisy sample. The centroid summarizes the cluster's high-density region and provides a smooth, robust target for guidance, whereas the closest point still carries instance-

Table 10: Comparison under 50% hard-label settings. Results are reported as Top-1 accuracy (%).

Dataset	Method	IPC=10			IPC=20			IPC=50		
		ConvNet	ResNet18	ResNet-AP	ConvNet	ResNet18	ResNet-AP	ConvNet	ResNet18	ResNet-AP
ImageIDC	Random	54.0±2.6	51.7±1.9	54.1±2.1	56.0±2.8	57.1±2.3	59.7±1.9	71.9±1.5	72.0±1.6	70.3±1.7
	FM	56.8±1.3	56.0±2.0	57.0±1.3	61.6±1.3	60.2±0.9	61.0±0.8	69.3±1.7	66.6±1.3	67.6±1.7
	MGD3	57.1±0.7	54.6±1.0	58.6±1.1	59.6±1.1	59.1±1.7	62.5±1.5	72.5±0.6	71.9±1.0	72.5±1.1
	GMFlow	**56.2±1.2**	**58.4±0.8**	**58.8±0.4**	**61.4±0.6**	**65.7±0.5**	**63.5±0.3**	**75.5±0.3**	**74.0±0.2**	**73.4±0.6**
ImageNette	Random	56.7±1.9	58.3±1.7	59.9±1.3	61.5±1.3	65.1±1.3	67.7±1.5	77.8±1.5	77.4±1.4	77.0±1.8
	FM	63.8±1.3	61.0±1.1	60.8±0.9	65.2±1.3	66.0±0.7	66.3±1.4	78.9±1.1	74.9±2.1	78.1±2.1
	MGD3	63.4±1.8	63.7±1.9	64.6±1.6	70.3±1.2	68.2±1.4	69.2±1.2	79.0±0.8	77.2±1.7	77.2±0.8
	GMFlow	**68.0±1.1**	**68.5±0.5**	**69.3±1.3**	**70.7±0.4**	**70.8±0.8**	**70.2±0.8**	**79.8±0.6**	**79.8±0.3**	**78.3±1.0**

level details and sample noise. Guiding trajectories toward such samples tends to overfit specific appearances and reduces within-class diversity.

Table 11: Ablation study on using closest point instead of centroid under setting with anchor $0.25, 0.5$, banksize 128, threshold 70%, ImageNette dataset, IPC $= 10$.

Method	**ConvNet**	**ResNet18**	**ResNet-AP**
Closest Point	61.5 ± 1.2	62.4 ± 1.4	63.7 ± 0.9
PGFM (Ours)	$\mathbf{63.4 \pm 1.1}$	$\mathbf{65.2 \pm 1.1}$	$\mathbf{66.4 \pm 0.1}$

Other cluster algorithm We further investigate the impact of different clustering algorithms by comparing our approach against K-means[Likas et al., 2003], Agglomerative Clustering [Ackermann et al., 2014] and Gaussian Mixture Models (GMM) [Rasmussen, 1999]. The results in Table 12 demonstrate that the K-means++ clustering algorithm consistently yields the best performance under setting with anchor $0.25, 0.5$, banksize 128, threshold 70%.

C.2.2 Retrieval Stage

We further examine the retrieval settings and the fallback. The threshold and bank size determine how often PGFM can find reliable reachable endpoints, while the fallback mechanism controls the remaining unmatched prototypes. We focus on several important hyperparameters, including the retrieval threshold, the bank size, and the number of trajectory anchors. We report both downstream accuracy and retrieval success rate, where success rate measures the fraction of prototypes that obtain valid retrieved initializations under the thresholded matching rule. Unless otherwise specified, all experiments in this subsection use the default setting: threshold$= 70\%$, one anchor at $s = 0.4$, and bank size $= 64$.

Threshold analysis We first study the effect of different retrieval thresholds. Table 13 summarizes the corresponding downstream performance and retrieval success rate.

Bank-size analysis Then, we conduct experiments on the bank size. Table 14 summarizes the corresponding downstream performance and retrieval success rate.

Anchors analysis Finally, we conduct experiments on the number of trajectory anchors. Table 15 summarizes the corresponding downstream performance and retrieval success rate.

C.2.3 Sampling Stage Ablation

Next, we conduct further experiments on the sampling stage.

Overall analysis Table 16 reports an ablation on ImageNette with IPC $= 10$ across three evaluation backbones under setting with anchor $0.25, 0.5$, banksize 128, threshold 70%. Starting from unguided FM sampling, adding path control alone already yields consistent gains (up to $+1.1$ points on ConvNet-6). However, path guidance can partially disrupt the underlying flow field, sometimes introducing blur or noise and leading to higher variance in downstream task. To mitigate it, we introduce early stop, which turns off guidance in the later phase so that the base flow can recover

Table 12: Ablation study on different clustering algorithms under setting with anchor $0.25, 0.5$, banksize 128, threshold 70% with ImageNette, IPC $= 10$

Method	**ConvNet**	**ResNet18**	**ResNet-AP**
GMM	61.6 ± 1.2	64.0 ± 0.9	63.6 ± 1.1
Agglomerative	61.4 ± 1.3	63.4 ± 1.0	64.0 ± 1.2
K-means	62.8 ± 1.1	64.0 ± 1.1	64.9 ± 0.1
K-means++	$\mathbf{63.4 \pm 1.1}$	$\mathbf{65.2 \pm 1.1}$	$\mathbf{66.4 \pm 0.1}$

Table 13: Ablation on retrieval threshold percentile.

Threshold (%)	Success	ConvNet-6	ResNet-18	ResNet-AP
0	10/100	63.6 ± 0.6	64.7 ± 1.3	65.9 ± 0.4
30	26/100	62.1 ± 1.8	63.7 ± 1.6	64.9 ± 0.1
50	46/100	64.4 ± 0.6	64.4 ± 0.0	65.4 ± 0.3
70	67/100	$\mathbf{64.7 \pm 1.3}$	$\mathbf{67.0 \pm 0.9}$	66.5 ± 0.9
100	98/100	64.0 ± 1.7	66.5 ± 0.4	$\mathbf{66.9 \pm 0.3}$

fine details. This further improves performance, especially on ResNet-18. Finally, the trust region for guidance stabilizes sampling by bounding the control strength relative to the model dynamics, preventing over-steering into poorly supported regions of the latent space. Combining all three components gives the most stable results, reaching the best performance.

Effect of η_{init} This ablation studies the warm-start fallback. Table 17 shows that the initialization strength η_{init} has a strong impact on prototype coverage. As η_{init} increases, the Hit Rate rises steadily (from 33% at η_{init}=0 to 86% at η_{init}=0.15), indicating that warm-starting closer to the assigned prototype helps the trajectory enter the correct mode basin early. However, larger η_{init} is not always better: overly aggressive initialization can over-constrain samples toward the prototype centers, reducing discriminative details and thereby hurting downstream task. As a result, η_{init} controls a clear trade-off between coverage and sample utility, and we find a moderate value (η_{init}=0.09) which achieves the best overall accuracy while substantially improving coverage on ImageNette for IPC $= 10$. Additional hyperparameter settings are provided in Appendix B.

Coverage analysis on different components Table 18 shows that unguided FM has very poor prototype coverage (10%), while MGD^3 achieves 30% but still mismatches many prototypes. For PGFM, path control alone helps but can perturb the original flow field and drift off-manifold. Although adding the early stop and trust region stabilizes the guidance, coverage remains limited. Warm start is the key to entering the correct prototype basin early, giving rise to a large jump in hit rate (78%) and enabling the flow to match most cluster centers reliably.

D Visualization

D.1 Effect of Trust Region ρ

Figure 5 shows that when we set a high ρ_0 or use a constant ρ throughout sampling, the generated images become noticeably blurred. These examples support using moderate guidance strength and turning off guidance in the later sampling stages.

D.2 Synthetic Images

Figure 6 shows the synthetic images that are generated on ImageNette for IPC $= 10$.

D.3 Flow Matching versus Diffusion

A comparison of the generation trajectories in Figure 7 highlights the superior efficiency of flow matching. We can see that the flow matching process yields a smooth evolution where recognizable

Table 14: Ablation on retrieval bank size.

Bank size	Success Rate	ConvNet-6	ResNet-18	ResNet-AP
32	0.65	59.1 ± 0.8	64.0 ± 0.9	64.8 ± 0.3
64	0.67	63.9 ± 1.9	$\mathbf{66.7 \pm 0.4}$	68.3 ± 0.5
128	0.69	64.2 ± 0.2	65.9 ± 2.0	65.5 ± 1.0
256	0.65	$\mathbf{64.6 \pm 0.9}$	65.7 ± 0.8	66.7 ± 0.6
512	0.68	64.3 ± 1.8	65.7 ± 1.5	$\mathbf{68.9 \pm 1.2}$

Table 15: Ablation on the number of trajectory anchors. The guidance stop time is fixed to $s_{\text{end}} = 0.6$.

Anchors	Anchor positions s	Success Rate	ConvNet-6	ResNet-18	ResNet-AP
1	0.4	0.67	62.1 ± 0.4	$\mathbf{67.8 \pm 0.6}$	$\mathbf{68.2 \pm 0.3}$
2	0.2, 0.4	0.67	62.3 ± 0.7	67.1 ± 0.7	67.9 ± 0.4
3	0.2, 0.4, 0.6	0.67	$\mathbf{62.5 \pm 0.4}$	67.4 ± 0.6	67.4 ± 1.4

structure emerges early in the trajectory, even under path-guided control. In contrast, the diffusion process in Figure 8 remains dominated by noise for the majority of the sampling timeline, with meaningful semantic content only materialized in the final stages. This indicates that diffusion-based methods require significantly more time and sampling steps to converge, whereas flow matching efficiently establishes image structure much earlier.

E Limitations and Future Work

PGFM has several limitations. Since it is built as a control layer on top of a frozen flow-matching generator, its performance is inherently constrained by the quality of the pretrained backbone and the VAE latent space. This issue is especially visible in more challenging settings, where prototype-level control alone may not be sufficient to capture subtle discriminative structure. In addition, although early stopping and trust-region scaling improve stability, the method still depends on several hyperparameters whose optimal values can vary across datasets and budgets. Our current prototype construction is also relatively simple, relying on latent-space clustering that may be suboptimal for long-tailed or highly nonuniform class distributions.

For our future work, it would be valuable to develop more adaptive control schedules, stronger prototype discovery strategies, and guidance mechanisms that incorporate downstream training signals more directly. A broader validation across different flow backbones, together with stronger theoretical analysis of mode coverage and manifold preservation, would further clarify the scope and generality of the framework.

F GenAI Usage Disclosure

In this work, we used the following generative AI tools:

- **Grammarly**: We used Grammarly (an AI-powered writing assistant) to check grammar and improve readability in this paper. All content was ultimately reviewed and approved by the authors.
- **ChatGPT-5.4**:
  - **code**: During the coding stage, ChatGPT-5.4 was used to assist in debugging by providing suggestions for correcting coding errors. All implementations were ultimately written, tested, and verified by the author.
  - **writing**: ChatGPT-5.4 was used exclusively during the final stage of manuscript preparation to assist in polishing the English descriptions within the experimental analysis section. All generated suggestions were carefully reviewed, edited, and adapted by the authors to ensure technical accuracy and align with the original intent. Other sections of the paper were not generated or rewritten using AI tools.

Table 16: Ablation study on ImageNette, IPC $= 10$ with anchor $0.25, 0.5$, banksize 128, threshold 70%. Path C means path control. Early S means early stop. Trust R means trust region.

Model	**Path C.**	**Early S.**	**Trust R.**	**Acc.**
ConvNet-6				60.5 ± 2.1
ResNetAP-10	–	–	–	61.9 ± 1.7
ResNet-18				62.4 ± 2.3
ConvNet-6				61.6 ± 1.2
ResNetAP-10	✓	–	–	62.4 ± 1.7
ResNet-18				63.0 ± 1.5
ConvNet-6				62.1 ± 1.3
ResNetAP-10	✓	✓	–	63.5 ± 1.6
ResNet-18				64.8 ± 2.0
ConvNet-6				$\mathbf{63.4} \pm 1.1$
ResNetAP-10	✓	✓	✓	$\mathbf{64.3} \pm 1.1$
ResNet-18				$\mathbf{66.4} \pm 1.1$

Table 17: Ablation study on initialization parameter η_{init} under setting with anchor $0.25, 0.5$, banksize 128, threshold 70%. We report both accuracy and Hit Rate (coverage) on ImageNette (IPC $= 10$).

η_{init}	Accuracy (%)			Hit Rate(%)
	ConvNet	ResNet18	ResNet-AP	
0	62.1 ± 1.3	63.5 ± 1.6	64.8 ± 2.0	33
0.05	61.6 ± 1.3	62.3 ± 0.9	63.8 ± 1.2	52
0.09	$\mathbf{63.4 \pm 1.1}$	$\mathbf{64.3 \pm 1.1}$	$\mathbf{66.4 \pm 1.1}$	78
0.15	62.0 ± 1.3	62.8 ± 0.8	64.2 ± 1.2	**86**

No AI tool was used to generate core content or conclusions.

Table 18: Coverage analysis on ImageNette, IPC $= 10$. Warm S means warm start. Hit rate (%) measures prototype coverage, which is the fraction of generated samples that successfully hit their assigned prototypes within a predefined tolerance.

Component	FM	MGD^3	PGFM			
Path C.	–	–	✓	✓	✓	✓
Early S.	–	–	–	✓	✓	✓
Trust R.	–	–	–	–	✓	✓
Warm S.	–	–	–	–	–	✓
Coverage/Hit rate (%)	10	30	18	27	33	**78**



Figure 5: Blurring synthetic images on ImageNette for IPC $= 10$

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: Our abstract and introduction has accurately reflect the paper's contribution and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

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

   Answer: [Yes]

   Justification: The limitation parts and future work is elaborated in the appendix E.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.

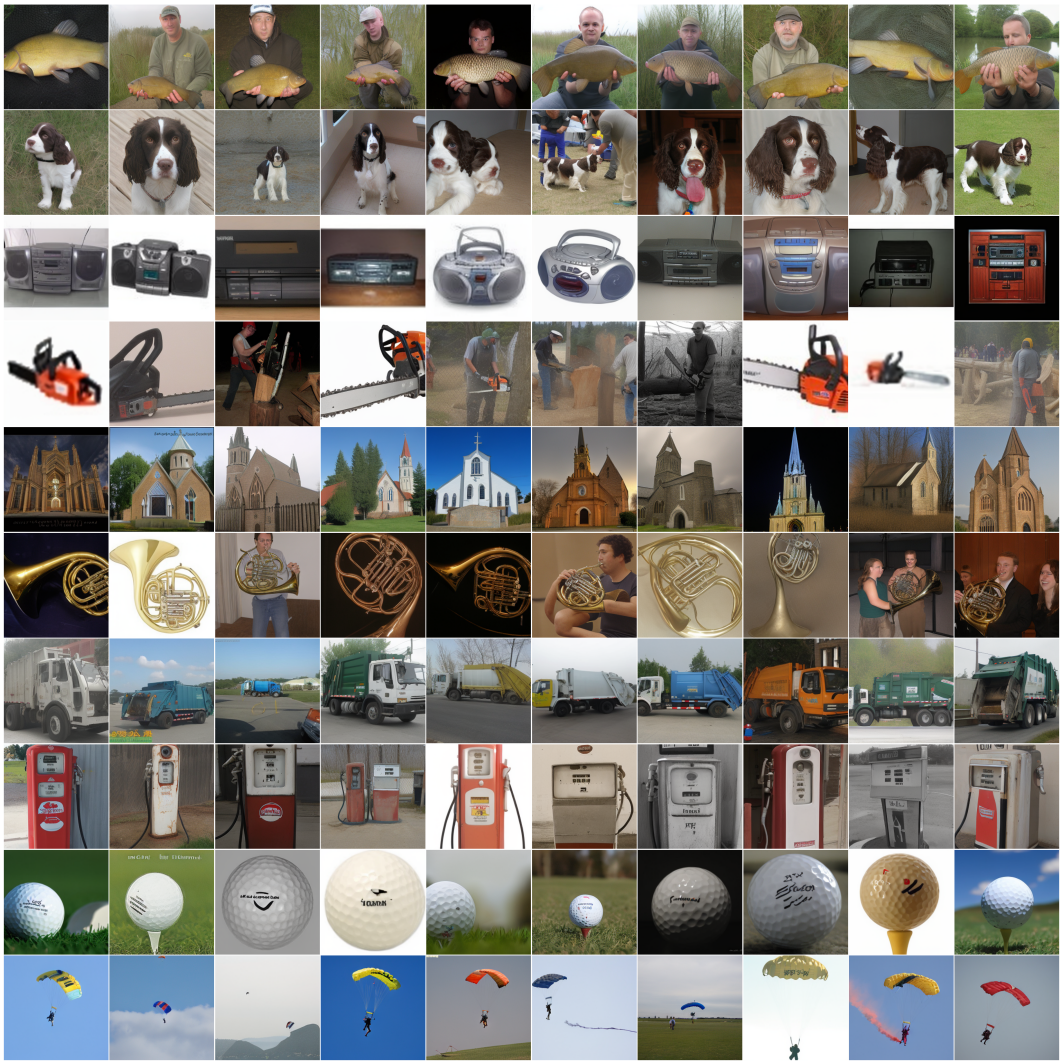


Figure 6: Synthetic images with ImageNette and IPC $= 10$

- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**



Figure 7: Visualization of the flow matching-based sampling step from time 0 to 1.

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

Answer: [Yes]

Justification: The assumptions are stated in Proposition 1. Appendix A proves the proposition and the resulting prototype-neighborhood condition.

Guidelines:

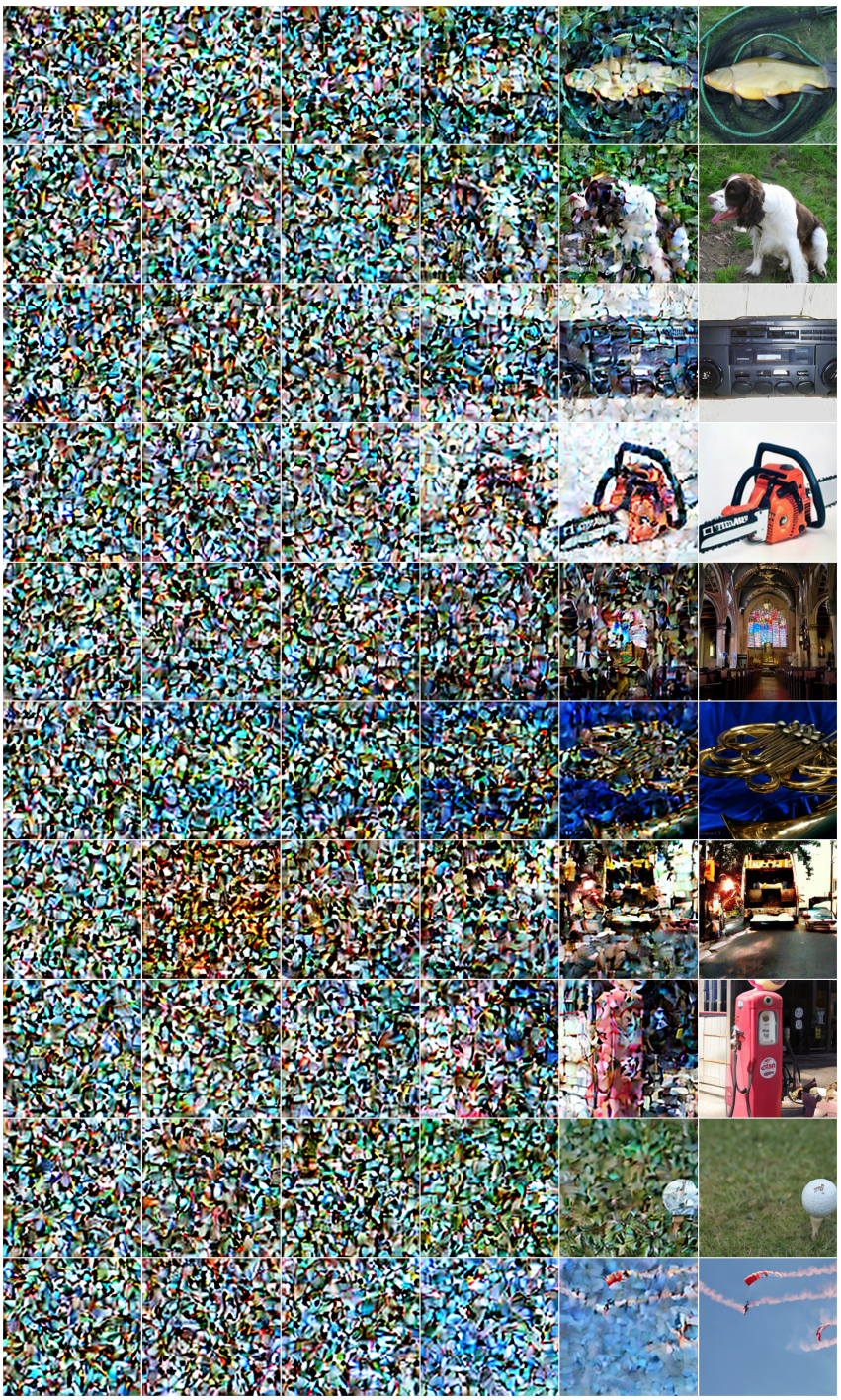


Figure 8: Visualization of the diffusion-based sampling.

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.

- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

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 have included a complete hyperparameter setting in appendix B. And the experiments is carefully designed to claim our contributions and theoretical analysis.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the 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: The code and data will be released once our paper get accepted.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.

- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

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) necessary to understand the results?

   Answer: [Yes]

   Justification: We have included a complete hyperparameter setting in appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

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: [Yes]

   Justification: We report all main experimental results as mean $\pm$ standard deviation over repeated runs under the same evaluation protocol.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).

- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

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: Our implement details has included the details information on the computer resources.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

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: Our research is conducted in the paper conform.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Our paper is a foundational research and not tied to particular applications.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: Our paper is a fundamental research. We do not provide any trained large language models or image generators.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

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 properly credit all external assets used in this work, including datasets, baseline methods, and the pretrained GMFlow model, and we use them in accordance with their original licenses and terms of use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

Answer: [N/A]

Justification: Not applicable for the current submission, since we do not release new assets (e.g., code, models, or datasets) at this stage.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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: [N/A]

Justification: Our experiments used open-source datasets.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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: [N/A]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

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 rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We have included a section to describe the usage of LLMs in appendix F.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.