DISTILLING DRIFTING TRANSFORMERS WITH REPRESENTATION AUTOENCODERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Despite the significant training acceleration and promising performance, effectively distilling models in Representation Autoencoder (RAE) latent spaces remains challenging. In this work, we argue that RAE is competent at high-quality one-step generation. We achieve **1.48 FID with only 16-epoch distillation** on ImageNet 256 dataset, surpassing various state-of-the-art methods. To achieve this, we quantitatively study the geometrical behavior of different underlying latent spaces. We conclude that conventional trajectory-based distillation methods heavily rely on priors of plain teacher trajectories, while RAE incurs more complex trajectories with poor properties due to the highly anisotropic latent space. We introduce the recently proposed drifting field as the distillation methodology, which makes use of semantically rich RAE latents and provides supervision without relying on teacher trajectories. Bridging our `Drift-RAE` with previous generative paradigms, we propose several insightful modifications, including the first extrapolation-based guided sampling method for one-step generation with minimal additional overhead.

1 INTRODUCTION

Diffusion and flow-based models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021a; Lipman et al., 2023; Liu et al., 2023; Peebles & Xie, 2023) have become the most dominant paradigm in generative modeling, achieving remarkable success in image (Esser et al., 2024; Black Forest Labs et al., 2025) and video (Blattmann et al., 2023; Brooks et al., 2024) synthesis. Among the literature, the autoencoders play a central role, which compress the data with low information density into compact latent spaces (Rombach et al., 2022; Zheng et al., 2026). This technique allows stable and efficient training in large-dimensional spaces such as high-resolution images and video. Representation Autoencoder (RAE) (Zheng et al., 2026; Tong et al., 2026b; Singh et al., 2026), one of the most representative frameworks, replaces the conventional VAE with pre-trained representation modules and corresponding decoders. The resulting latent spaces allocate much richer semantics and strongly label-wise clustered features, thus improving training efficiency and generation quality.

Despite their great potential, efficient distillation in RAE latent spaces remains challenging. Like other diffusion and flow-based models, RAE also relies on the iterative sampling steps along the denoising trajectories by probability-flow ODE. Conventional methods distill the pre-trained models using the intrinsic priors of the denoising process (Salimans & Ho, 2022; Song et al., 2023; Wang et al., 2024; Sauer et al., 2024a;b; Lin et al., 2024; Yin et al., 2024a;b; Zhou et al., 2024), *i.e.*, compressing the lengthy trajectory consisting of hundreds of steps into one-step or few-step generation. However, prior attempts suggest that distilling models in RAE latent spaces can suffer from unstable optimization and limited generation quality (Hu et al., 2026). These challenges motivate more effective and efficient distillation strategies tailored to RAE latent spaces.

In this work, we introduce an efficient and effective pipeline for RAE distillation. We first claim in Section 3.2 that the challenges of distillation in RAE latent space stem from the severe anisotropy of the patch-wise token distributions. Consequently, the mismatch for RAE to transport isotropic Gaussian noise to the underlying anisotropic latent space would enforce extremely curved denoising trajectories. On the other hand, conventional trajectory-based distillation methods implicitly rely on plain or almost linear teacher trajectories. Therefore, the large curvatures of trajectories in RAE greatly increase the difficulty of distillation. To this end, we propose to distill using Drifting Models (Deng et al., 2026). Drifting Models are a one-step generative paradigm, which directly

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

compute a field to evaluate the discrepancy between generated and real distributions. The drifting field enforces the generated distribution to coincide with the real one and involves no reliance on teacher trajectories, and thus could serve as a competent technique for RAE distillation.

However, the original Drifting Model heavily relies on an MAE module as an auxiliary extractor to filter and concentrate features in the underlying space, incurring additional training overhead. Our theoretical analysis of an idealized spherical sampling model illustrates that overly dispersed distributions in high-dimensional spaces can lead to a loss of spatial selectivity in the positive component of the drifting field. In contrast, the semantically rich RAE latent space naturally exhibits well-clustered patch embeddings, providing a favorable geometry for effective positive-field supervision. Meanwhile, the analogous behavior of the negative drifting field suggests that concentrated initial predictions are also desirable, aligning naturally with pretrained initialization in the distillation setting. To this end, we propose our RAE distillation method, namely `Drift-RAE`, with high efficiency and great performance. To take a further step, we bridge Drifting Models with other generative paradigms, which motivates several practical modifications with high empirical efficacy. Moreover, we introduce Internal Drifting Guidance (IDG), an extrapolation-based guided sampling pipeline for one-step generation with negligible additional overhead.

We evaluate the proposed `Drift-RAE` on ImageNet 256×256. Benefiting from the proposed techniques, we achieve **1.48 FID within 16-epoch distillation**, surpassing or appearing competitive with state-of-the-art one-step or even few-step distillation methods beyond RAE latent spaces (Kim et al., 2024; Chen et al., 2026; Tong et al., 2026a; Liu & Yue, 2025; Peng et al., 2026), without requiring an auxiliary MAE feature extractor. These results demonstrate the effectiveness of drifting-based distillation in RAE latent spaces for efficient and high-quality one-step generation.

2 Related Work

2.1 Flow-Based Models and Distillation

Flow-based Models, including Diffusion Models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021a;b; Rombach et al., 2022) and Flow Matching (Liu et al., 2023; Lipman et al., 2023), are designed to formulate the relation between data and noise distributions through differential equations. Specifically, the training stage introduces a forward process by corrupting initial data signals with independent noise, while the inference stage numerically solves the corresponding ODE or SDE with a learned score. However, approximating the scores of the whole process in a complex pixel space is extremely challenging. To this end, LDM (Vahdat et al., 2021; Rombach et al., 2022) and RAE (Zheng et al., 2026; Singh et al., 2026) shift flow-based generative modeling from the original pixel space to learned latent spaces. Despite the unprecedented capability, the iterative reverse process hinders the sampling efficiency of flow-based models. To address this issue, many attempts have been made to distill the knowledge from pre-trained models and reduce the denoising steps (Salimans & Ho, 2022; Song et al., 2023; Luo et al., 2023; Yin et al., 2024b; Zhou et al., 2024).

2.2 One-Step Generation Trained From Scratch

Generative Adversarial Network (GAN) is the most representative paradigm to train a one-step generator from scratch (Goodfellow et al., 2014; Brock et al., 2019; Kang et al., 2023), which simultaneously trains a generator and a discriminator via adversarial training. However, GAN training can suffer from unstable performance due to mode collapse (Arjovsky & Bottou, 2017). Another family of methods directly realizes the one-step generation by incorporating the prior SDE or ODE dynamics and approximating the corresponding trajectories (Song et al., 2023; Song & Dhariwal, 2024; Geng et al., 2025; 2026). Drifting Model (Deng et al., 2026) is a novel framework, which proposes to progressively evolve the generated distribution towards the real one with a specially designed drifting field. Concretely, the drifting field characterizes the discrepancy between two distributions via pairwise distances and a contrastive-style attraction–repulsion mechanism.

2.3 Guided Sampling

Modern diffusion and flow-based models apply various guidance methods to improve sampling quality. Classifier Guidance (Dhariwal & Nichol, 2021) and Classifier-Free Guidance (Ho & Salimans, 2021)

propose to adjust the conditional score with classifier gradient and jointly learned unconditional score, respectively, which increases the conditional fidelity. AutoGuidance (Karras et al., 2024a) generalizes the idea beyond conditional posterior, and introduces extrapolation between a "bad version" and the model itself, serving as a new paradigm of guided sampling. More recently, Zhou et al. (2026) demonstrate the internal representations of the diffusion transformer are feasible to provide guidance for the backbone output. Singh et al. (2026) further show that the REPA head (Yu et al., 2025) can be viewed as a "weak model" for AutoGuidance. Such internal guidance methods avoid auxiliary models or additional forward passes and therefore enjoy superior computational efficiency.

3 METHOD

3.1 PREREQUISITES

Let q denote the real data distribution and let $\mathbf{y} \sim q$. Flow matching (Liu et al., 2023), one of the most representative flow-based models, defines a forward process by linear interpolation, *i.e.*,

$$\mathbf{y}_t = (1-t)\mathbf{y} + t\boldsymbol{\epsilon}, \tag{1}$$

in which $t \in [0,1]$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$. Then the flow matching starts the generation process at $t = 1$ from pure Gaussian noise with an underlying velocity term $\mathbf{v}(\mathbf{y}_t, t)$:

$$\mathrm{d}\mathbf{y}_t = \mathbf{v}(\mathbf{y}_t, t)\mathrm{d}t, \tag{2}$$

in which the velocity term $\mathbf{v}(\mathbf{y}_t, t)$ can be expressed as

$$\mathbf{v}(\mathbf{y}_t, t) = \mathbb{E}[\dot{\mathbf{y}}_t | \mathbf{y}_t] = \mathbb{E}[\boldsymbol{\epsilon} - \mathbf{y} | \mathbf{y}_t]. \tag{3}$$

Therefore, flow matching employs a model $\mathbf{v}_\theta(\mathbf{y}_t, t)$ to approximate the unknown velocity field $\mathbf{v}(\mathbf{y}_t, t)$ by minimizing the following regression objective:

$$\mathcal{L}(\theta) = \int_0^1 \mathbb{E}_{\mathbf{y},\boldsymbol{\epsilon}} \|\mathbf{v}_\theta(\mathbf{y}_t, t) - (\boldsymbol{\epsilon} - \mathbf{y})\|^2 \mathrm{d}t. \tag{4}$$

Drifting Model (Deng et al., 2026) trains a one-step generator from scratch by computing the drifting field between real data samples $\{\mathbf{y}_i\}$ and synthesized samples $\{\mathbf{x}_j\}$. Notably, the drifting field enforces each $\mathbf{x}_j$ to move away from other $\{\mathbf{x}_k\}_{k \neq j}$ (negative samples) and towards $\{\mathbf{y}_i\}$ (positive samples). Formally, the drifting field $\mathbf{V}_j$ for each $\mathbf{x}_j$ could be formulated as below:

$$\mathbf{V}_j = \sum_i \frac{e^{-\frac{1}{\tau}\|\mathbf{y}_i - \mathbf{x}_j\|}}{\sum_l e^{-\frac{1}{\tau}\|\mathbf{y}_l - \mathbf{x}_j\|}} (\mathbf{y}_i - \mathbf{x}_j) - \sum_{k \neq j} \frac{e^{-\frac{1}{\tau}\|\mathbf{x}_k - \mathbf{x}_j\|}}{\sum_{m \neq j} e^{-\frac{1}{\tau}\|\mathbf{x}_m - \mathbf{x}_j\|}} (\mathbf{x}_k - \mathbf{x}_j). \tag{5}$$

Deng et al. (2026) claim that, if the drifting fields are annihilated throughout the entire space, the synthesized distribution coincides with the real data distribution.

3.2 RETHINKING THE DYNAMICS OF RAE AND DRIFTING MODEL

Trajectory-based distillation methods typically rely on the implicit assumption that the underlying latent space is approximately isotropic, such that the induced flow ODE trajectories remain sufficiently smooth and have moderate curvature (Wu et al., 2025). Motivated by this, we compare in Figure 1 the curvatures of 32 flow ODE trajectories in RAE and traditional SD-VAE latent spaces, following the analysis protocol of Chen et al. (2024). The results show that trajectories in the RAE latent space have curvature values approximately **two orders of magnitude larger** than those of SD-VAE, indicating substantially more complex and sharply varying transport dynamics. In addition, Table 1 reports the average participation ratio (PR) and spectral entropy (SE) per token, further revealing that the RAE latent space is substantially more anisotropic. These observations suggest that conventional trajectory-based distillation methods can become ineffective when directly applied to RAEs, motivating the need for alternative approaches that explicitly account for the geometry of the RAE latent space.

To address this issue, we claim that the recently proposed Drifting Models (Deng et al., 2026) are well suited for flow distillation in RAE latent spaces. Drifting Models are designed to narrow the gap between distributions by directly computing the drifting field with two batches of samples instead of

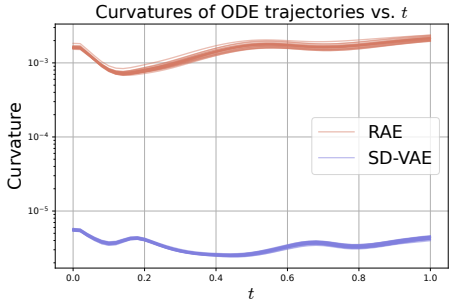

Figure 1: **Curvatures** of ODE trajectories in RAE and SD-VAE latent spaces.

Table 1: **Isotropy** statistics of latent features.

Latent space	PR (↑)	SE (↑)
SD-VAE	**0.2068**	**0.3781**
RAE	0.0630	0.2110

Table 2: **Dispersion** statistics of latent features.

Latent space	NN-d (↓)	S-MMD^2 (↑)
SD-VAE	4.3083	0.00564
RAE	**1.0664**	**0.07913**

matching ODE trajectories. Therefore, unlike conventional trajectory-based distillation methods, the negative effects of highly curved trajectories in RAE latent spaces are mostly alleviated.

Despite the straightforward methodology, we further argue that RAE latent spaces could conversely complement the training dynamics of Drifting Models. Recall that in the original Drifting Model, empirically it is necessary to involve an auxiliary MAE as the feature extractor. We below provide a theoretical explanation for the potential role of the auxiliary MAE under an idealized spherical sampling model. The corresponding proof is deferred to Appendix A.1.

Theorem 1. *For each $d \geq 2$, let $\{\mathbf{y}_i\}_{i=1}^d$ be independent samples drawn uniformly from the unit sphere $\mathbb{S}^{d-1} \subset \mathbb{R}^d$. Let $\mathbf{x}_d \in \mathbb{R}^d$ be a possibly random query independent of these samples. For a fixed temperature $\tau > 0$ that does not depend on d, define*

$$a_{i,d}(\mathbf{x}_d) = \frac{\exp(-\|\mathbf{y}_i - \mathbf{x}_d\|_2/\tau)}{\sum_{\ell=1}^d \exp(-\|\mathbf{y}_\ell - \mathbf{x}_d\|_2/\tau)}, \qquad \mathbf{V}_d^+(\mathbf{x}_d) = \sum_{i=1}^d a_{i,d}(\mathbf{x}_d)\mathbf{y}_i. \tag{6}$$

Then, as $d \to \infty$,

$$\max_{1\leq i\leq d} |d\, a_{i,d}(\mathbf{x}_d) - 1| \xrightarrow{\mathbb{P}} 0, \qquad d\left\|\mathbf{V}_d^+(\mathbf{x}_d)\right\|_2^2 \xrightarrow{\mathbb{P}} 1. \tag{7}$$

In particular, $\|\mathbf{V}_d^+(\mathbf{x}_d)\|_2 \xrightarrow{\mathbb{P}} 0$.

Theorem 1 shows that, under the stated spherical sampling model and a fixed temperature, the positive weights become asymptotically uniform, and their weighted centroid approaches the origin. This illustrates a potential loss of spatial selectivity in dispersed high-dimensional representations, which may hinder drifting-based training by reducing locally informative, query-specific supervision. Further empirical evidence is reported in Table 2, in which NN-d reports the average nearest-neighbor distance and S-MMD^2 evaluates the maximum mean discrepancy between the sample distribution and a spherical distribution. It is noteworthy that SD-VAE suggests a severely dispersed latent space. This suggests that the auxiliary MAE, especially one fine-tuned with classification loss, may benefit drifting-based training by providing more concentrated semantic features.

In contrast, RAE enjoys substantially more concentrated latent spaces, suggesting that RAE could serve as a more favorable underlying latent space for Drifting Models and relieve the redundant module. Furthermore, by analogy with the idealized analysis in Theorem 1, overly dispersed initial predictions may similarly weaken the spatial selectivity of the negative drifting field. This intuition favors concentrated initialization and naturally motivates pretrained flow models, which typically produce more structured and concentrated predictions than random initialization. Accordingly, in the sequel, we focus on distillation in RAE latent spaces under the drifting dynamics. More discussion on training Drifting Models from scratch in RAE latent space is provided in Appendix C.

3.3 Distillation via Drifting in RAE Latent Spaces

We now introduce our method for distilling flow-based models in RAE latent spaces using Drifting Models. Let $\mathbf{v}_\theta(\mathbf{y}, t)$ denote a pretrained flow model. We form a one-step generator to distill as:

$$\mathbf{G}_\theta(\mathbf{z}) = \mathbf{z} - \mathbf{v}_\theta(\mathbf{z}, 1), \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{8}$$

where θ is initialized from the pretrained flow model. Given a batch of latents $\{\mathbf{y}_i\}_{i=1}^{N_{\text{pos}}}$ sampled from the real distribution, we write

$$\mathbf{y}_i = (\mathbf{y}_i^1, \ldots, \mathbf{y}_i^c, \ldots, \mathbf{y}_i^C), \tag{9}$$

where each $\mathbf{y}_i^c \in \mathbb{R}^D$ denotes the c-th patch token, C is the number of patch tokens, and D is the hidden size. The token-wise output of the generator is defined analogously as $\mathbf{G}_\theta^c(\mathbf{z})$.

Motivated by the analysis in Section 3.2, we use the semantically rich RAE latents themselves as the feature space for drifting-based distillation, eliminating the need for an auxiliary MAE in our method. Specifically, we directly define the drifting objective on each RAE latent token as

$$L(\theta) = \sum_{j=1}^{N_{\text{neg}}} \sum_{c=1}^{C} \left\| \mathbf{G}_\theta^c(\mathbf{z}_j) - \text{sg}\left[\mathbf{G}_\theta^c(\mathbf{z}_j) + \tilde{\mathbf{V}}_j^c\right] \right\|^2, \qquad \mathbf{z}_j \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{10}$$

where N_{neg} is the number of generated samples within a minibatch, $\text{sg}[\cdot]$ denotes the stop-gradient operation, and $\tilde{\mathbf{V}}_j^c = \mathbf{V}_j^c / \mathbb{E}_{i,c}[\|\mathbf{y}_i^c\|]$ is the drifting field normalized by the mean ℓ_2 norm across real samples and tokens, with $\mathbf{V}_j^c$ computed by

$$\mathbf{V}_j^c = \sum_{i=1}^{N_{\text{pos}}} \frac{e^{-\frac{1}{\tau}\|\mathbf{y}_i^c - \mathbf{G}_\theta^c(\mathbf{z}_j)\|} \left(\mathbf{y}_i^c - \mathbf{G}_\theta^c(\mathbf{z}_j)\right)}{\sum_{l=1}^{N_{\text{pos}}} e^{-\frac{1}{\tau}\|\mathbf{y}_l^c - \mathbf{G}_\theta^c(\mathbf{z}_j)\|}} - \sum_{k \neq j} \frac{e^{-\frac{1}{\tau}\|\mathbf{G}_\theta^c(\mathbf{z}_k) - \mathbf{G}_\theta^c(\mathbf{z}_j)\|} \left(\mathbf{G}_\theta^c(\mathbf{z}_k) - \mathbf{G}_\theta^c(\mathbf{z}_j)\right)}{\sum_{l \neq j} e^{-\frac{1}{\tau}\|\mathbf{G}_\theta^c(\mathbf{z}_l) - \mathbf{G}_\theta^c(\mathbf{z}_j)\|}}. \tag{11}$$

Beyond the objective in Eq. (10), we subsequently introduce four pillars of modification to further improve the distillation dynamics. Among them, three insights are motivated by theoretical connections between Drifting Models and previous generative paradigms, especially Diffusion-GAN (Wang et al., 2023). Concretely, the drifting field shares the single-softmax weighting and attraction-repulsion structure of the optimal-discriminator logit gradient between kernel-smoothed empirical distributions. Detailed descriptions and discussions are postponed to Appendix A.2.

Softmax axes. Recall that the original Drifting Model proposed a bi-directional softmax trick over both the query and reference axes, which is claimed to improve training stability. Here, queries are generated samples at which the drifting field is evaluated and through which gradients are propagated, while references are the positive and negative samples used to compute it. However, the additional query-axis softmax is not directly motivated by the kernel-score interpretation. We therefore retain only the reference-axis softmax over sample indices. This naturally arises from differentiating a log-sum-exp potential and preserves the weighting structure identified by our theoretical analysis.

Perturbing inputs with noise. Original Drifting Models evaluate the drifting field at clean generated samples. Inspired by the connection to Diffusion-GAN (Wang et al., 2023), we instead evaluate the field at noisy queries, providing supervision in neighborhoods of the generated samples:

$$\bar{\mathbf{G}}_\theta^c(\mathbf{z}_j) = \mathbf{G}_\theta^c(\mathbf{z}_j) + \tau \mathbf{n}_j^c, \tag{12}$$

where $\mathbf{n}_j^c \in \mathbb{R}^D$ is a random-direction vector whose norm is sampled from an exponential distribution with unit rate, i.e., $\|\mathbf{n}_j^c\| \sim \text{Exp}(1)$, and τ is the same temperature hyperparameter used in the drifting field. The perturbation is applied only to the queries at which the drifting field is evaluated.

Partially detaching negative samples. Recall that Deng et al. (2026) conclude that enlarging N_{pos} and N_{neg} is beneficial. However, even when each GPU processes only one class, the per-GPU memory budget limits the number of gradient-carrying negative samples to at most $N_{\text{neg}} = 64$ without gradient checkpointing. To further increase the number of samples with negligible additional GPU memory overhead, we propose to partially detach the generated samples. These auxiliary samples aim to more accurately approximate the generated distribution, which improves the stability of distillation. To summarize, by fixing $N_{\text{pos}} = 256$ and 64 negative samples to backpropagate, we employ $N_{\text{extra_neg}} = 192$ negative samples to detach and equivalently achieve $N_{\text{total_neg}} = 256$.

CLS token and registers. We follow Darcet et al. (2024) to concatenate a CLS token and four extra register tokens with the patchified sequence. Specifically, the CLS token is taken to compute the drifting loss, while the extra registers have no target and only provide redundancy.

3.4 Further Improvements via Internal Drifting Guidance

Conventional score-based guidance does not directly carry over to one-step generators, which directly predict denoised samples rather than the intermediate score estimate typically required for guidance. Nevertheless, we propose Internal Drifting Guidance (IDG), which, to the best of our knowledge, constitutes the first attempt to enable extrapolation-based guided sampling for one-step generation.

We design the guidance with internal representations from middle Transformer blocks following (Zhou et al., 2026; Singh et al., 2026). IG (Zhou et al., 2026) trains intermediate predictions with the same denoising objective as the final output and uses them as weaker references for extrapolation, which can be technically viewed as some variant of AutoGuidance (Karras et al., 2024a). Inspired by this design, we train an intermediate projection head $\mathbf{P}_\psi$ to provide a reference prediction for one-step guidance. Then at inference time, we apply guidance with guidance strength ω as

$$\mathbf{G}_\theta^{\text{IDG}}(\mathbf{z}, \omega) = \omega \mathbf{G}_\theta(\mathbf{z}) - (\omega - 1)\mathbf{P}_\psi(\phi(\mathbf{z})), \tag{13}$$

where $\phi(\mathbf{z})$ denotes the intermediate features from the backbone. To this end, we propose to optimize $\mathbf{P}_\psi$ through the drifting loss, in which $\mathbf{G}_\theta(\mathbf{z})$ serves as positive samples. This encourages the head to match the backbone's prediction distribution through attraction and repulsion, rather than explicitly enforcing pointwise agreement for each input noise. We further detach the intermediate features and backbone references when training the head, so that the head loss does not update the backbone.

It is noteworthy that our IDG involves neither auxiliary noise-data pairs nor an additional distillation stage, and could be implemented without requiring an additional forward pass. Most notably, benefiting from the drifting objective on $\mathbf{P}_\psi$, the extrapolation in Eq. (13) could be interpreted as a one-negative drifting from a theoretical perspective, *i.e.*, $\mathbf{G}_\theta(\mathbf{z})$ is drifted away from the negative $\mathbf{P}_\psi(\phi(\mathbf{z}))$. More ablations and discussions on the design of IDG are addressed in Section B.3.

4 Experiments

We evaluate the proposed method on class-conditional image generation with representation autoencoders. Section 4.1 describes the experimental setup, with detailed hyperparameter settings provided in Appendix B.2. Section 4.2 presents the main results, and Section 4.3 provides ablation studies on the modifications introduced in Section 3.3.

4.1 Experimental Setup

Dataset and pretrained checkpoints. We evaluate our method on ImageNet 256×256 (Deng et al., 2009) using DiT^{DH}-XL and DiT^{DH}-L from the RAE codebase[1] (Zheng et al., 2026). Specifically, we use the released DiT^{DH}-XL checkpoint, and train DiT^{DH}-L ourselves following the official code.

Evaluation metrics. We report FID (Heusel et al., 2017) to evaluate generation quality. In addition, we use Precision and Recall (Kynkäänniemi et al., 2019) to measure sample fidelity and diversity, respectively. All metrics are computed using $50{,}000$ generated samples. We adopt class-balanced sampling (Zheng et al., 2026), *i.e.*, generating 50 images for each of the $1{,}000$ ImageNet classes.

Training configurations. Following Deng et al. (2026), the Drifting field is computed at three temperature values, $\{0.02, 0.05, 0.2\}$, and the final objective is obtained by averaging the corresponding losses. For each class and token position, real and generated latents are normalized by the mean ℓ_2 norm of real latents across samples. We fix $N_{\text{class}} = 32$ and train for $10{,}000$ steps, corresponding to approximately 16 epochs, where training epochs are counted using only gradient-carrying samples following Deng et al. (2026). AdamW is used with $\beta_1 = 0.9$, $\beta_2 = 0.95$, and weight decay is set to 0. For stable distillation in RAE latent spaces, we linearly decay the learning rate from 3×10^{-5} to 3×10^{-7} over training and apply gradient clipping with a threshold of 1.0. We do not use exponential moving average (EMA) during distillation. For IDG, we concatenate the features from layers 8 and 14, and then apply a projection head consisting of a three-layer MLP with SiLU activations. We use IDG with $\omega = 1.4$ for the main results, while the training ablations in Tables 3, 5, 6 and 7 are evaluated without IDG. We retain the CLS tokens produced by the DINO encoder as positive samples for the generated CLS tokens. Further details are available in Appendix B.2.

[1] https://github.com/bytetriper/RAE

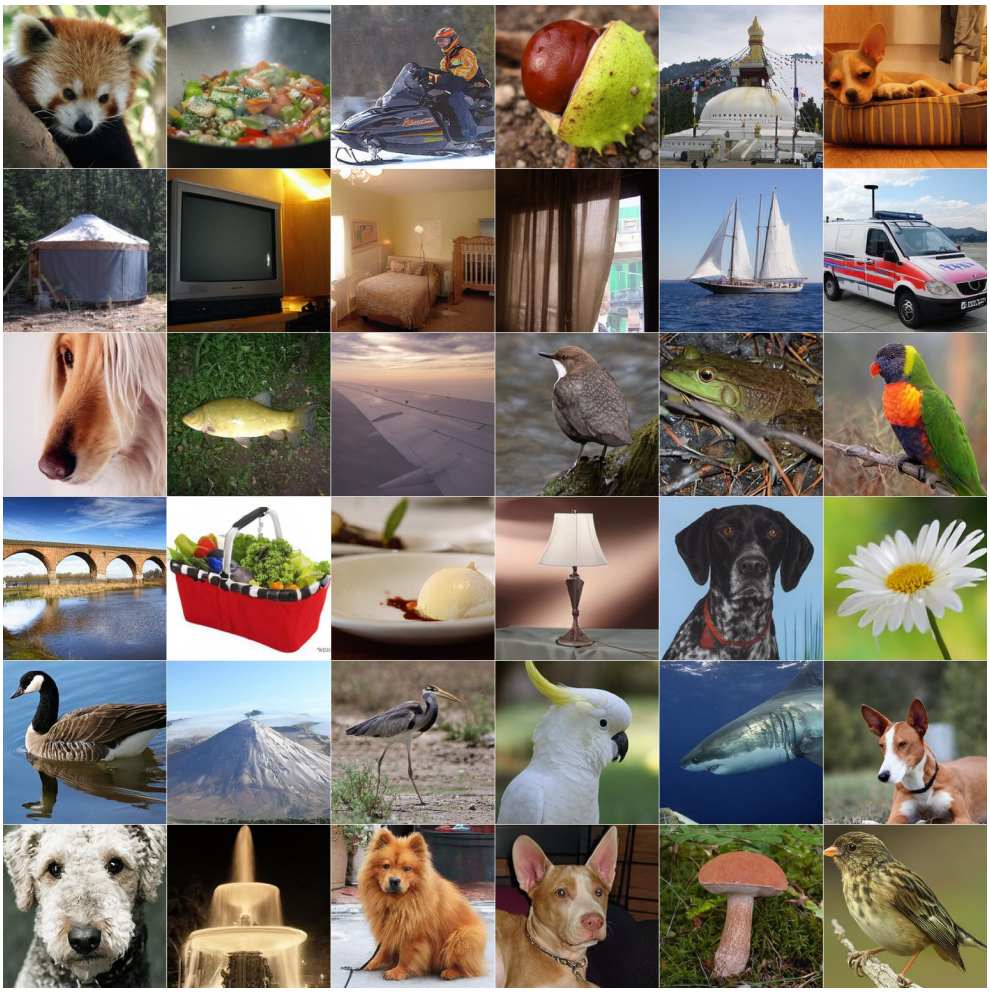

Figure 2: **Visualizations** of generated samples from distilled DiT^{DH}-XL (FID = 1.48).

4.2 Main Results

Table 3 reports the practical effects of the modifications introduced in Section 3.3 with DiT^{DH}-XL. Here, the baseline uses the additional query-axis softmax and sets $N_{\text{pos}} = N_{\text{neg}} = 64$ following the original Drifting Model (Deng et al., 2026). Perturbing inputs with noise and removing the additional query-axis softmax make the implementation more consistent with our theoretical analysis, while also improving empirical generation performance. Further increasing both the numbers of positive and negative samples to 256 yields the best result. Applying IDG to configuration C provides an additional improvement of FID from 1.74 to 1.48.

Table 3: Effect of modifications in Section 3.3.

Config	Modifications	FID (↓)
A	Baseline	2.00
B	+ Input perturb., − query-softmax	1.86
C	+ 192 pos., + 192 detached neg.	**1.74**

Final generation results are reported in Table 4. Our proposed `Drift-RAE` achieves an FID of 1.82 with DiT^{DH}-L and 1.48 with DiT^{DH}-XL using a single generation step, outperforming the previous distillation method in RAE latent spaces, while remaining competitive with distilled models in other spaces. Moreover, `Drift-RAE` reaches an FID of 1.48 within only 16 epochs, demonstrating favorable distillation efficiency. These results suggest that Drifting provides an effective framework for distilling flow-based models trained in representation spaces.

Compared with the original Drifting Model, `Drift-RAE` achieves comparable FID while eliminating the need for an auxiliary MAE. These results support the hypothesis motivated by Theorem 1 and our

Table 4: Main results on ImageNet 256×256. † indicates distillation methods. For each group separately, i.e., vanilla diffusion models, few-step models, and *one-step* models, **bold** indicates the best result, and underlining indicates the second-best result.

Method	Space	NFE	Epochs	FID (↓)	Prec. (↑)	Rec. (↑)
ADM-G (Dhariwal & Nichol, 2021)	Pixel	250	400	4.59	0.82	0.52
DiT-XL/2 (Peebles & Xie, 2023)	VAE	250	1400	2.27	**0.83**	0.57
SiT-XL/2 (Ma et al., 2024)	VAE	250	1400	2.06	0.82	0.59
LightningDiT-XL (Yao & Wang, 2025)	VAE	250	800	1.35	0.79	0.65
DiT^{DH}-XL (Zheng et al., 2026)	RAE	50	800	**1.13**	0.78	**0.67**
IMM-XL/2 ($\omega = 1.5$) (Zhou et al., 2025)	VAE	8	3837	1.99	–	–
STEI† (Liu & Yue, 2025)	VAE	8	20	1.96	–	–
MeanFlow-XL/2+ (Geng et al., 2025)	VAE	2	1000	2.20	–	–
iMF-XL/2 (Geng et al., 2026)	VAE	2	800	1.61	**0.79**	**0.63**
π-Flow† (Chen et al., 2026)	VAE	2	76	1.97	–	–
FACM† (Peng et al., 2026)	VAE	2	60	**1.32**	–	–
MF-RAE† (Hu et al., 2026)	RAE	2	41	1.89	–	–
BigGAN (Brock et al., 2019)	Pixel	1	–	6.95	**0.89**	0.38
StyleGAN-XL (Sauer et al., 2022)	Pixel	1	–	2.30	0.78	0.53
Drifting Model, L/16 (Deng et al., 2026)	Pixel	1	640	1.61	0.81	0.60
MeanFlow-H/16 (Lu et al., 2026)	Pixel	1	320	2.29	0.80	0.59
PaGoDA† (Kim et al., 2024)	Pixel	1	–	1.56	–	0.59
IMM-XL/2 ($\omega = 1.5$) (Zhou et al., 2025)	VAE	1	3837	8.05	–	–
STEI† (Liu & Yue, 2025)	VAE	1	20	7.12	–	–
MeanFlow-XL/2 (Geng et al., 2025)	VAE	1	240	3.43	–	–
iMF-XL/2 (Geng et al., 2026)	VAE	1	800	1.82	0.78	0.63
Drifting Model, L/2 (Deng et al., 2026)	VAE	1	1280	1.54	0.79	0.63
π-Flow† (Chen et al., 2026)	VAE	1	448	2.85	–	–
FreeFlow-XL/2† (Tong et al., 2026a)	VAE	1	300	**1.45**	–	–
MF-RAE† (Hu et al., 2026)	RAE	1	41	2.03	–	–
`Drift-RAE` (DiT^{DH}-L)†	RAE	1	16	1.82	0.77	0.64
`Drift-RAE` (DiT^{DH}-XL)†	RAE	1	16	1.48	0.77	**0.66**

representation statistics: the semantic structure of RAE latents helps preserve informative drifting supervision, allowing effective distillation without an additional MAE feature extractor. Qualitative results are shown in Figure 2, with more visualizations provided in Appendix D.

4.3 Ablation Studies

Input perturbation and softmax axes. As shown in Table 5, removing the query-axis softmax and including input perturbations both lead to improvement over the original drifting baseline. Simultaneously applying both modifications yields similar performance to solely removing the query-axis softmax, while we observe **more stable distillation** with input perturbation. We therefore adopt both modifications as the default configuration throughout all subsequent experiments.

Increasing the number of samples. Table 7 studies the effect of increasing the number of positive and negative samples used to estimate the Drifting field. Increasing only the number of positive samples or only the number of negative samples leads to substantial performance degradation, while the best performance is achieved when positive samples and negative samples are balanced. This observation suggests that an imbalanced estimation of the attractive and repulsive components can systematically bias the resulting Drifting direction and consequently degrade the quality of distillation.

Increasing the number of gradient-carrying negative samples. With $N_{\text{total_neg}} = 256$ fixed, we further enable gradient checkpointing and vary the number of gradient-carrying negatives as $N_{\text{neg}} \in \{64, 128, 192, 256\}$. As shown in Table 6, increasing N_{neg} does not bring performance gains. This indicates that the main benefit of using more negative samples comes from improving the

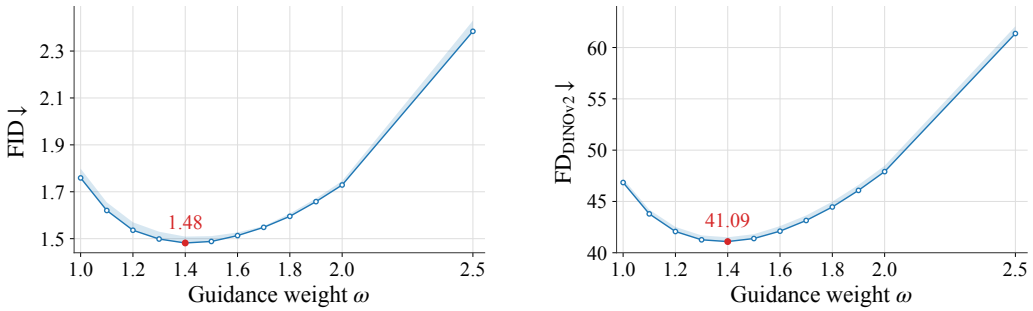

Figure 3: FID and FD_{DINOv2} (Stein et al., 2023) vs. guidance weight ω of IDG. The shaded regions indicate the min/max values over 3 evaluations, following Karras et al. (2024b).

Table 5: Ablation on input perturbation and removing query-axis softmax.

Input Perturb.	Remove query-softmax	FID (↓)
✗	✗	2.00
✗	✓	1.90
✓	✗	1.88
✓	✓	**1.86**

Table 6: Ablation on $N_{\text{extra_neg}}$.

N_{pos}	$N_{\text{total_neg}}$	$N_{\text{extra_neg}}$	FID (↓)
256	256	192	1.74
		128	1.77
		64	1.75
		0	1.74

Table 7: Ablation on different combinations of N_{pos}, $N_{\text{total_neg}}$, and $N_{\text{extra_neg}}$. We highlight the **balanced** configurations with $N_{\text{pos}} = N_{\text{total_neg}}$ in color gray.

N_{pos}	$N_{\text{total_neg}}$	$N_{\text{extra_neg}}$	FID (↓)
64	64	0	1.86
	128	64	2.74
	256	192	4.66
128	64	0	2.20
	128	64	1.82
256	64	0	4.49
	256	192	**1.74**

empirical approximation of the generated distribution, rather than from applying gradients to more generated samples. Such a conclusion is also favorable from an implementation perspective, since detached negative samples require no gradient storage and introduce lower memory overhead.

Guidance weight. Figure 3 reports FID and FD_{DINOv2} (Stein et al., 2023) under different guidance weights. Similar to other guidance methods, both FID and FD_{DINOv2} first decrease and then increase as the guidance weight increases, with the optimal point for both metrics occurring at $\omega = 1.4$.

5 Conclusion

In this paper, we propose Drifting-based distillation for flow-based models in RAE latent spaces. We first quantitatively analyze the geometry of RAE latent spaces and theoretically study the dynamics of Drifting, showing that the highly curved ODE trajectories in RAE latent spaces make trajectory-based distillation challenging, while their semantically concentrated representations allow Drifting to operate without an auxiliary MAE. Motivated by a connection between Drifting Models and the Diffusion-GAN framework, we introduce several practical modifications that improve distillation performance. We further propose an extrapolation-based guided sampling method for one-step generation with negligible additional cost. Experiments on ImageNet 256×256 demonstrate that our method outperforms previous distillation methods in RAE latent spaces and achieves performance comparable to the original Drifting Model and distilled models in other spaces.

Limitations and Future Work. Although this work eliminates the need for an additional MAE for Drifting-based distillation in RAE spaces, achieving competitive generation quality when training Drifting Models from scratch without an auxiliary MAE remains an open challenge. Developing native Drifting training methods that do not rely on auxiliary modules is still an important direction. Moreover, the need for many same-class positive samples at each Drifting update may hinder scalability, especially in text-to-image generation settings. Reducing this dependence on abundant positive samples is also a promising direction for future research.

AI USE STATEMENT

We used generative AI tools for language polishing and formatting checks. The authors reviewed all AI-assisted revisions. The authors take full responsibility for the final content of this paper, including the accuracy of its technical claims, its experimental results, and references.

ETHICS STATEMENT

Though this work primarily focuses on methodological advances in efficient one-step image generation, we acknowledge the potential ethical risks associated with generative models. The resulting models may inherit biases present in the training data and pretrained components. Moreover, reducing the computational cost of image generation may facilitate misuse, including the creation of misleading or deceptive visual content. These risks should be considered in downstream applications, with appropriate safeguards and clear disclosure of AI-generated content.

REPRODUCIBILITY STATEMENT

We describe our method in Section 3 and the experimental setup and evaluation protocols in Section 4.1. Proofs and additional theoretical derivations are provided in Appendix A. The procedures for our statistical analyses are detailed in Appendix B.1, while the training hyperparameters and additional implementation details are provided in Appendix B.2.

REFERENCES

Martín Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. In *International Conference on Learning Representations*, 2017.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. FLUX.1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. OpenAI technical report, 2024. URL `https://openai.com/index/video-generation-models-as-world-simulators/`.

Defang Chen, Zhenyu Zhou, Can Wang, Chunhua Shen, and Siwei Lyu. On the trajectory regularity of ODE-based diffusion sampling. In *International Conference on Machine Learning*, 2024.

Hansheng Chen, Kai Zhang, Hao Tan, Leonidas Guibas, Gordon Wetzstein, and Sai Bi. π-Flow: Policy-based few-step generation via imitation distillation. In *International Conference on Learning Representations*, 2026.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *International Conference on Learning Representations*, 2024.

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

Mingyang Deng, He Li, Tianhong Li, Yilun Du, and Kaiming He. Generative modeling via drifting. *arXiv preprint arXiv:2602.04770*, 2026.

Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, 2021.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning*, 2024.

Zixuan Fu, Chong Wang, Lanqing Guo, Kailai Zhou, Jiahao Nie, and Bihan Wen. A frozen pixel-space diffusion model can guide itself with its own samples. *arXiv preprint arXiv:2607.29122*, 2026.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *Advances in Neural Information Processing Systems*, 2025.

Zhengyang Geng, Yiyang Lu, Zongze Wu, Eli Shechtman, J Zico Kolter, and Kaiming He. Improved mean flows: On the challenges of fastforward generative models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *Advances in Neural Information Processing Systems Workshop*, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020.

Zheyuan Hu, Chieh-Hsin Lai, Ge Wu, Yuki Mitsufuji, and Stefano Ermon. MeanFlow transformers with representation autoencoders. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up GANs for text-to-image synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. In *Advances in Neural Information Processing Systems*, 2024a.

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

Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Yuhta Takida, Naoki Murata, Toshimitsu Uesaka, Yuki Mitsufuji, and Stefano Ermon. PaGoDA: Progressive growing of a one-step generator from a low-resolution diffusion teacher. In *Advances in Neural Information Processing Systems*, 2024.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems*, 2019.

Chieh-Hsin Lai, Bac Nguyen, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yuki Mitsufuji, Stefano Ermon, and Molei Tao. A unified view of score-based and drifting models. *arXiv preprint arXiv:2603.07514*, 2026.

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

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Wenze Liu and Xiangyu Yue. Learning to integrate diffusion ODEs by averaging the derivatives. In *Advances in Neural Information Processing Systems*, 2025.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023.

Yiyang Lu, Susie Lu, Qiao Sun, Hanhong Zhao, Zhicheng Jiang, Xianbang Wang, Tianhong Li, Zhengyang Geng, and Kaiming He. One-step latent-free image generation with pixel mean flows. In *International Conference on Machine Learning*, 2026.

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

Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. SiT: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *International Conference on Computer Vision*, 2023.

Yansong Peng, Kai Zhu, Yu Liu, Pingyu Wu, Hebei Li, Xiaoyan Sun, and Feng Wu. FACM: Flow-anchored consistency models. In *International Conference on Learning Representations*, 2026.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

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

Axel Sauer, Katja Schwarz, and Andreas Geiger. StyleGAN-XL: Scaling stylegan to large diverse datasets. In *SIGGRAPH*, 2022.

Axel Sauer, Frederic Boesel, Tim Dockhorn, Andreas Blattmann, Patrick Esser, and Robin Rombach. Fast high-resolution image synthesis with latent adversarial diffusion distillation. In *SIGGRAPH Asia*, 2024a.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, 2024b.

Jaskirat Singh, Boyang Zheng, Zongze Wu, Richard Zhang, Eli Shechtman, and Saining Xie. Improved baselines with representation autoencoders. *arXiv preprint arXiv:2605.18324*, 2026.

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

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a.

Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. In *International Conference on Learning Representations*, 2024.

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

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, 2023.

George Stein, Jesse C. Cresswell, Rasa Hosseinzadeh, Yi Sui, Brendan Leigh Ross, Valentin Villecroze, Zhaoyan Liu, Anthony L. Caterini, J. Eric T. Taylor, and Gabriel Loaiza-Ganem. Exposing flaws of generative model evaluation metrics and their unfair treatment of diffusion models. In *Advances in Neural Information Processing Systems*, 2023.

Shangyuan Tong, Nanye Ma, Saining Xie, and Tommi Jaakkola. Flow map distillation without data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026a.

Shengbang Tong, Boyang Zheng, Ziteng Wang, Bingda Tang, Nanye Ma, Ellis Brown, Jihan Yang, Rob Fergus, Yann LeCun, and Saining Xie. Scaling text-to-image diffusion transformers with representation autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, 2026b.

Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. In *Advances in Neural Information Processing Systems*, 2021.

Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*, volume 47 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, 2018. doi: 10.1017/9781108231596.

Fu-Yun Wang, Zhaoyang Huang, Alexander W Bergman, Dazhong Shen, Peng Gao, Michael Lingelbach, Keqiang Sun, Weikang Bian, Guanglu Song, Yu Liu, Xiaogang Wang, and Hongsheng Li. Phased consistency models. In *Advances in Neural Information Processing Systems*, 2024.

Zhendong Wang, Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. Diffusion-GAN: Training GANs with diffusion. In *International Conference on Learning Representations*, 2023.

Zhangkai Wu, Xuhui Fan, Hongyu Wu, and Longbing Cao. SCoT: Unifying consistency models and rectified flows via straight-consistent trajectories. In *Advances in Neural Information Processing Systems*, 2025.

Jingfeng Yao and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Frédo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. In *Advances in Neural Information Processing Systems*, 2024a.

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

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *International Conference on Learning Representations*, 2025.

Le Zhang, Ning Mang, and Aishwarya Agrawal. RiT: Vanilla diffusion transformers suffice in representation space. *arXiv preprint arXiv:2605.21981*, 2026.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *International Conference on Learning Representations*, 2026.

Linqi Zhou, Stefano Ermon, and Jiaming Song. Inductive moment matching. In *International Conference on Machine Learning*, 2025.

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

Xingyu Zhou, Qifan Li, Xiaobin Hu, Hai Chen, and Shuhang Gu. Guiding a diffusion transformer with the internal dynamics of itself. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

APPENDIX

In this appendix, we provide additional technical details and discussions omitted from the main text. Appendix A presents the detailed proof of Theorem 1 in Section 3.2, and establishes the connection between Drifting Models and Diffusion-GAN. Appendix B provides further implementation details and more ablation studies. Appendix C presents additional attempts and discussions on training Drifting Models from scratch. Appendix D presents additional qualitative generation results.

A PROOFS AND DERIVATIONS

A.1 PROOF OF THEOREM 1

Proof. By the concentration of spherical marginals (Vershynin, 2018, Theorem 3.4.6), there exists an absolute constant $c_0 > 0$ independent of d, such that, for any deterministic unit vector $\mathbf{u} \in \mathbb{S}^{d-1}$ and $\mathbf{Y} \sim \mathrm{Unif}(\mathbb{S}^{d-1})$,

$$\mathbb{P}(|\langle \mathbf{u}, \mathbf{Y} \rangle| > t) \le 2e^{-c_0 d t^2}, \qquad t > 0. \tag{14}$$

Let $\mathbf{u}_d = \mathbf{x}_d / \|\mathbf{x}_d\|$ when $\mathbf{x}_d \neq \mathbf{0}$, and let $\mathbf{u}_d$ be a fixed unit vector otherwise. By independence, the same concentration bound applies to $\langle \mathbf{u}_d, \mathbf{y}_i \rangle$ conditionally on $\mathbf{x}_d$, and also to $\langle \mathbf{y}_i, \mathbf{y}_j \rangle$ for $i \neq j$.

We choose $K > 0$ such that $c_0 K^2 > 3$, and define

$$t_d = K \sqrt{\frac{\log d}{d}}. \tag{15}$$

Then consider the event

$$\mathcal{E}_d = \left\{ \max_{1 \le i \le d} |\langle \mathbf{u}_d, \mathbf{y}_i \rangle| \le t_d, \quad \max_{1 \le i < j \le d} |\langle \mathbf{y}_i, \mathbf{y}_j \rangle| \le t_d \right\}. \tag{16}$$

The union bound in Equation (14) gives

$$\mathbb{P}(\mathcal{E}_d^c) \le 2d e^{-c_0 d t_d^2} + 2\binom{d}{2} e^{-c_0 d t_d^2} = (d^2 + d) d^{-c_0 K^2} \to 0, \tag{17}$$

and thus $\mathbb{P}(\mathcal{E}_d) \to 1$. Then the following weights and the remainder bounds are established on $\mathcal{E}_d$.

We first establish the asymptotic uniformity of the positive weights. Let

$$D_{i,d} = \|\mathbf{y}_i - \mathbf{x}_d\|, \qquad R_d = \sqrt{1 + \|\mathbf{x}_d\|^2}. \tag{18}$$

Since $\|\mathbf{y}_i\| = 1$, we have

$$D_{i,d}^2 = R_d^2 - 2\langle \mathbf{x}_d, \mathbf{y}_i \rangle. \tag{19}$$

Consequently, on $\mathcal{E}_d$,

$$|D_{i,d} - R_d| = \frac{2|\langle \mathbf{x}_d, \mathbf{y}_i \rangle|}{D_{i,d} + R_d} \le \frac{2\|\mathbf{x}_d\|}{R_d} |\langle \mathbf{u}_d, \mathbf{y}_i \rangle| \le 2t_d. \tag{20}$$

Define $b_{i,d} = e^{-(D_{i,d} - R_d)/\tau}$, and thus $a_{i,d}(\mathbf{x}_d) = b_{i,d} / \sum_{l=1}^{d} b_{l,d}$. Then on $\mathcal{E}_d$,

$$e^{-2t_d/\tau} \le b_{i,d} \le e^{2t_d/\tau}, \tag{21}$$

and therefore

$$e^{-4t_d/\tau} \le d\, a_{i,d}(\mathbf{x}_d) \le e^{4t_d/\tau}. \tag{22}$$

Let $\eta_d = e^{4t_d/\tau} - 1$. It follows that

$$\max_{1 \le i \le d} |d\, a_{i,d}(\mathbf{x}_d) - 1| \le \eta_d. \tag{23}$$

Since $t_d \to 0$ and τ is fixed, $\eta_d \to 0$. Together with Equation (17), this proves

$$\max_{1\le i\le d} |d\, a_{i,d}(\mathbf{x}_d) - 1| \xrightarrow{\mathbb{P}} 0. \tag{24}$$

We next compare the weighted centroid with the empirical mean. Define

$$\bar{\mathbf{y}}_d = \frac{1}{d}\sum_{i=1}^{d} \mathbf{y}_i, \qquad \delta_{i,d} = a_{i,d}(\mathbf{x}_d) - \frac{1}{d}, \qquad \mathbf{r}_d = \sum_{i=1}^{d} \delta_{i,d}\mathbf{y}_i. \tag{25}$$

Then $\mathbf{V}_d^+(\mathbf{x}_d) = \bar{\mathbf{y}}_d + \mathbf{r}_d$. By Equation (23),

$$|\delta_{i,d}| \le \frac{\eta_d}{d}, \qquad \sum_{i=1}^{d} \delta_{i,d}^2 \le \frac{\eta_d^2}{d}, \qquad \sum_{i=1}^{d} |\delta_{i,d}| \le \eta_d. \tag{26}$$

Using the pairwise inner-product bound in $\mathcal{E}_d$, we obtain

$$\|\mathbf{r}_d\|^2 = \sum_{i=1}^{d} \delta_{i,d}^2 + \sum_{i\ne j} \delta_{i,d}\delta_{j,d}\langle \mathbf{y}_i, \mathbf{y}_j\rangle \le \sum_{i=1}^{d} \delta_{i,d}^2 + t_d \left(\sum_{i=1}^{d} |\delta_{i,d}|\right)^2 \le \eta_d^2 \left(\frac{1}{d} + t_d\right). \tag{27}$$

For sufficiently large d, there exists a constant $C_\tau > 0$, independent of d, such that $\eta_d \le C_\tau t_d$. Thus,

$$d\|\mathbf{r}_d\|^2 \le C_\tau^2 \left(t_d^2 + d t_d^3\right) = C_\tau^2 \left(\frac{K^2 \log d}{d} + \frac{K^3 (\log d)^{3/2}}{\sqrt{d}}\right) \to 0. \tag{28}$$

Since $\mathbb{P}(\mathcal{E}_d) \to 1$, we conclude that

$$\sqrt{d}\,\|\mathbf{r}_d\| \xrightarrow{\mathbb{P}} 0. \tag{29}$$

It remains to characterize the empirical mean. Expanding its squared norm gives

$$\|\bar{\mathbf{y}}_d\|^2 = \frac{1}{d} + \frac{2}{d^2} \sum_{1\le i<j\le d} \langle \mathbf{y}_i, \mathbf{y}_j\rangle. \tag{30}$$

By independence and spherical symmetry, for $i \ne j$,

$$\mathbb{E}\langle \mathbf{y}_i, \mathbf{y}_j\rangle = 0, \qquad \mathbb{E}\langle \mathbf{y}_i, \mathbf{y}_j\rangle^2 = \frac{1}{d}. \tag{31}$$

Moreover, inner products corresponding to distinct unordered index pairs are uncorrelated. Hence,

$$\mathbb{E}\|\bar{\mathbf{y}}_d\|^2 = \frac{1}{d}, \qquad \mathrm{Var}\big(\|\bar{\mathbf{y}}_d\|^2\big) = \frac{4}{d^4}\binom{d}{2}\frac{1}{d} = \frac{2(d-1)}{d^4}. \tag{32}$$

By Chebyshev's inequality, for every $d \ge 2$ and $\varepsilon > 0$,

$$\mathbb{P}\big(\big|d\|\bar{\mathbf{y}}_d\|^2 - 1\big| \ge \varepsilon\big) \le \frac{2(d-1)}{d^2\varepsilon^2}. \tag{33}$$

In particular,

$$d\|\bar{\mathbf{y}}_d\|^2 \xrightarrow{\mathbb{P}} 1, \qquad \sqrt{d}\,\|\bar{\mathbf{y}}_d\| \xrightarrow{\mathbb{P}} 1. \tag{34}$$

Finally, the reverse triangle inequality gives

$$\Big|\sqrt{d}\,\|\mathbf{V}_d^+(\mathbf{x}_d)\| - \sqrt{d}\,\|\bar{\mathbf{y}}_d\|\Big| \le \sqrt{d}\,\|\mathbf{r}_d\| \xrightarrow{\mathbb{P}} 0. \tag{35}$$

Consequently,

$$\sqrt{d}\,\|\mathbf{V}_d^+(\mathbf{x}_d)\| \xrightarrow{\mathbb{P}} 1, \tag{36}$$

which implies

$$d\|\mathbf{V}_d^+(\mathbf{x}_d)\|^2 \xrightarrow{\mathbb{P}} 1. \tag{37}$$

It follows immediately that $\|\mathbf{V}_d^+(\mathbf{x}_d)\| \xrightarrow{\mathbb{P}} 0$. □

A.2 Drifting as Empirical Diffusion-GAN

In this section, we connect Drifting Models to adversarial training, especially the viewpoint of Diffusion-GAN (Wang et al., 2023). The key observation is that kernel smoothing yields an optimal-discriminator logit gradient with a weighted attraction-repulsion structure, providing a theoretical perspective on the practical drifting field.

Let q denote the target distribution and $p = \mathbf{G}_{\theta\#}p_{\mathbf{z}}$ denote the generated distribution, where $p_{\mathbf{z}}$ is the noise distribution. For a fixed generator, the optimal discriminator for the standard GAN objective is

$$D^*_{q,p}(\mathbf{x}) = \frac{q(\mathbf{x})}{q(\mathbf{x}) + p(\mathbf{x})}, \tag{38}$$

whose logit is

$$\operatorname{logit} D^*_{q,p}(\mathbf{x}) = \log \frac{D^*_{q,p}(\mathbf{x})}{1 - D^*_{q,p}(\mathbf{x})} = \log q(\mathbf{x}) - \log p(\mathbf{x}). \tag{39}$$

Therefore, the gradient of the optimal discriminator logit gives the score difference

$$\nabla_{\mathbf{x}} \operatorname{logit} D^*_{q,p}(\mathbf{x}) = \nabla_{\mathbf{x}} \log q(\mathbf{x}) - \nabla_{\mathbf{x}} \log p(\mathbf{x}). \tag{40}$$

The gradient of the non-saturating generator loss $-\log D^*_{q,p}(\mathbf{x})$ is the negative score difference up to a positive scalar factor, since

$$\nabla_{\mathbf{x}} \left[-\log D^*_{q,p}(\mathbf{x}) \right] = \frac{p(\mathbf{x})}{q(\mathbf{x}) + p(\mathbf{x})} \left(\nabla_{\mathbf{x}} \log p(\mathbf{x}) - \nabla_{\mathbf{x}} \log q(\mathbf{x}) \right). \tag{41}$$

We now consider empirical distributions smoothed by a kernel. Given real samples $\{\mathbf{y}_i\}_{i=1}^N \sim q$ and generated samples $\{\mathbf{x}_j\}_{j=1}^M \sim p$, define empirical measures

$$\hat{q} = \frac{1}{N} \sum_{i=1}^{N} \delta_{\mathbf{y}_i}, \qquad \hat{p} = \frac{1}{M} \sum_{j=1}^{M} \delta_{\mathbf{x}_j}. \tag{42}$$

For $l > 0$, consider the exponential kernel

$$k_l(\mathbf{x}; \tau) = \exp\left(-\frac{\|\mathbf{x}\|_2^l}{\tau} \right), \tag{43}$$

and the smoothed empirical densities (up to a common normalizing constant)

$$\hat{q}_l(\mathbf{x}) = (k_l * \hat{q})(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} k_l(\mathbf{x} - \mathbf{y}_i; \tau), \quad \hat{p}_l(\mathbf{x}) = (k_l * \hat{p})(\mathbf{x}) = \frac{1}{M} \sum_{j=1}^{M} k_l(\mathbf{x} - \mathbf{x}_j; \tau). \tag{44}$$

The following proposition connects the kernel-induced attraction-repulsion field and the optimal-discriminator logit gradient, which is also closely related to the results in Lai et al. (2026).

Proposition 1. *For an evaluation point $\mathbf{x} \notin \{\mathbf{y}_i\}_{i=1}^N \cup \{\mathbf{x}_j\}_{j=1}^M$, let*

$$\mathbf{V}_l(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i^+(\mathbf{x}) \|\mathbf{y}_i - \mathbf{x}\|_2^{l-2} (\mathbf{y}_i - \mathbf{x}) - \sum_{j=1}^{M} \alpha_j^-(\mathbf{x}) \|\mathbf{x}_j - \mathbf{x}\|_2^{l-2} (\mathbf{x}_j - \mathbf{x}), \tag{45}$$

where

$$\alpha_i^+(\mathbf{x}) = \frac{k_l(\mathbf{x} - \mathbf{y}_i; \tau)}{\sum_{n=1}^{N} k_l(\mathbf{x} - \mathbf{y}_n; \tau)}, \qquad \alpha_j^-(\mathbf{x}) = \frac{k_l(\mathbf{x} - \mathbf{x}_j; \tau)}{\sum_{m=1}^{M} k_l(\mathbf{x} - \mathbf{x}_m; \tau)}. \tag{46}$$

Then

$$\nabla_{\mathbf{x}} \operatorname{logit} D^*_{\hat{q}_l, \hat{p}_l}(\mathbf{x}) = \frac{l}{\tau} \mathbf{V}_l(\mathbf{x}). \tag{47}$$

Proof. We first compute the score of $\hat{q}_l$. Since

$$\nabla_{\mathbf{x}} k_l(\mathbf{x}-\mathbf{y}_i;\tau) = -\frac{l}{\tau}\|\mathbf{x}-\mathbf{y}_i\|_2^{l-2}(\mathbf{x}-\mathbf{y}_i)k_l(\mathbf{x}-\mathbf{y}_i;\tau), \tag{48}$$

we have

$$\begin{aligned}\nabla_{\mathbf{x}} \log \hat{q}_l(\mathbf{x}) &= \frac{l}{\tau}\sum_{i=1}^{N}\frac{k_l(\mathbf{x}-\mathbf{y}_i;\tau)}{\sum_{n=1}^{N}k_l(\mathbf{x}-\mathbf{y}_n;\tau)}\|\mathbf{y}_i-\mathbf{x}\|_2^{l-2}(\mathbf{y}_i-\mathbf{x}) \\ &= \frac{l}{\tau}\sum_{i=1}^{N}\alpha_i^{+}(\mathbf{x})\|\mathbf{y}_i-\mathbf{x}\|_2^{l-2}(\mathbf{y}_i-\mathbf{x}).\end{aligned} \tag{49}$$

Analogously,

$$\nabla_{\mathbf{x}} \log \hat{p}_l(\mathbf{x}) = \frac{l}{\tau}\sum_{j=1}^{M}\alpha_j^{-}(\mathbf{x})\|\mathbf{x}_j-\mathbf{x}\|_2^{l-2}(\mathbf{x}_j-\mathbf{x}). \tag{50}$$

Substituting Equation (49) and Equation (50) into Equation (40) gives the desired result. □

Proposition 1 identifies the kernel-induced field $\mathbf{V}_l$ with the optimal-discriminator logit gradient up to a constant factor. In particular, when $l = 2$, the norm factor disappears, and $\mathbf{V}_2$ becomes the standard attraction-repulsion field induced by a Gaussian/RBF kernel. When $l = 1$ as used in practice, the same derivation introduces inverse-distance factors, yielding a *normalized* direction for each displacement. In high-dimensional settings where the relevant pairwise distances concentrate around a common scale, these factors are approximately constant across sample pairs. In practical implementations of Drifting, these per-displacement normalization factors are typically omitted. Thus Drifting can be viewed as estimating, up to scale, the optimal discriminator logit gradient between two perturbed distributions, making it related to the Diffusion-GAN (Wang et al., 2023) framework.

This connection provides the motivation for the modifications introduced in Section 3.3:

First, the reference-axis softmax weights arise naturally from differentiating the log-density of an exponential-kernel mixture. The additional query-axis softmax is not induced by this derivation, motivating our choice to retain only the reference-axis softmax.

Second, the connection to discrimination between perturbed distributions motivates evaluating the drifting field at perturbed generated samples, which we implement by injecting noise into queries. However, using the theoretically matched perturbation scale can be inefficient in high-dimensional spaces, as perturbed samples may rarely stay near the clean generated samples and thus provide less direct supervision to the generator. In practice, we therefore use random-direction noise whose norm follows an exponential distribution with unit rate, which provides a practical trade-off between sampling from the perturbed distribution and maintaining sample efficiency.

Finally, Drifting relies on Monte Carlo estimation of the underlying distributions. Using more samples improves the empirical approximation of both real and generated distributions, leading to more accurate estimates of the Drifting direction.

B Implementation Details and Ablation Studies

B.1 Details of Statistical Analysis in Section 3.2

Here we provide additional details for the statistical analyses used in Section 3.2, including trajectory curvature, isotropy, and dispersion statistics.

Trajectory curvature. For the SD-VAE latent space, we use a DiT-XL model from Peebles & Xie (2023); for the RAE latent space, we use the DiT^{DH}-XL model from Zheng et al. (2026). We sample 32 trajectories for each model using the first-order ODE sampler with the sampling parameters following the respective official implementations. The sampled trajectories are directly fed into the official implementation of Chen et al. (2024), following its curvature evaluation protocol without additional preprocessing. We report the curvature values returned by this implementation.

Isotropy statistics. We quantify the isotropy of latents using the participation ratio (PR) and spectral entropy (SE). For *each class* and *each spatial token*, we collect the corresponding latents across samples and compute their covariance matrix. Let $\{\mu_i\}_{i=1}^r$ be the non-negative eigenvalues of this covariance matrix, where r is the number of eigenvalues. We define the normalized spectrum as

$$p_i = \mu_i / \sum_{j=1}^{r} \mu_j. \tag{51}$$

The normalized participation ratio is computed as

$$\mathrm{PR} = \frac{\left(\sum_{i=1}^r \mu_i\right)^2}{r \sum_{i=1}^r \mu_i^2} = \frac{1}{r \sum_{i=1}^r p_i^2}, \tag{52}$$

and the normalized spectral entropy is computed as

$$\mathrm{SE} = \frac{1}{r} \exp\left(-\sum_{i=1}^{r} p_i \log p_i\right). \tag{53}$$

Both metrics are normalized to lie in $[1/r, 1]$, with larger values indicating a more isotropic spectrum. For SD-VAE, we flatten each 2×2 latent patch into a 16-dimensional token. We compute PR and SE separately for each class and each token, and then average over all classes and all tokens. We note that a concurrent work (Zhang et al., 2026) also studies representation-space geometry and reports conclusions that appear different from ours. We suspect that the discrepancy mainly comes from the aggregation protocol. Zhang et al. (2026) measure global statistics after mixing samples from all classes and aggregating token positions, while our analysis is performed per class and per token. For the purpose of this paper, our protocol is preferred since DiTs condition on class embeddings and process latents as token sequences. Therefore, the relevant geometry is the local within-class, within-token geometry rather than the global geometry obtained by aggregating all classes and tokens.

Dispersion statistics. We measure how dispersed samples are in SD-VAE and RAE latent spaces using nearest-neighbor distance (NN-d) and spherical maximum mean discrepancy (S-MMD^2). Since SD-VAE and RAE latents have different dimensionalities, we normalize all Euclidean distances by $\sqrt{d}$ to remove the scaling of Euclidean distance with dimensionality, where d denotes the feature dimension. NN-d is averaged over all samples, while S-MMD^2 is averaged across all 1000 classes.

NN-d is the average distance from each sample to its nearest neighbor within the same class:

$$\text{NN-d} = \frac{1}{\sum_{c=1}^{C} n_c} \sum_{c=1}^{C} \sum_{i=1}^{n_c} \min_{j \neq i} \frac{1}{\sqrt{d}} \|\mathbf{x}_{c,i} - \mathbf{x}_{c,j}\|_2, \tag{54}$$

where C is the total number of classes and n_c is the number of samples within class c. A smaller NN-d indicates closer neighbors and thus more concentrated samples.

S-MMD^2 compares the empirical sample distribution with a spherical reference distribution. For a set of samples $\{\mathbf{x}_{c,i}\}_{i=1}^{n_c}$, we define the reference radius as $\rho_c = \frac{1}{n_c} \sum_{i=1}^{n_c} \|\mathbf{x}_{c,i}\|_2$. Then we use the uniform distribution on the origin-centered sphere of radius ρ_c as the reference:

$$Q_c = \mathrm{Unif}(\rho_c \mathbb{S}^{d-1}). \tag{55}$$

Let $\hat{P}_c = \frac{1}{n_c} \sum_{i=1}^{n_c} \delta_{\mathbf{x}_{c,i}}$ denote the empirical latent distribution of class c. We compute

$$\text{S-MMD}^2(\hat{P}_c, Q_c) = \frac{1}{n_c^2} \sum_{i,j=1}^{n_c} k(\mathbf{x}_{c,i}, \mathbf{x}_{c,j}) + \mathbb{E}_{\mathbf{U},\mathbf{U}' \overset{\text{iid}}{\sim} Q_c} k(\mathbf{U}, \mathbf{U}') - \frac{2}{n_c} \sum_{i=1}^{n_c} \mathbb{E}_{\mathbf{U} \sim Q_c} k(\mathbf{x}_{c,i}, \mathbf{U}),$$

where $k(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|_2}{\sqrt{d}}\right)$. To evaluate the spherical expectations efficiently, we exploit rotational symmetry. For a fixed vector $\mathbf{x}$ with $r = \|\mathbf{x}\|_2$ and $\mathbf{U} \sim Q_c$,

$$\|\mathbf{x} - \mathbf{U}\|_2 \overset{\mathrm{d}}{=} \sqrt{r^2 + \rho_c^2 - 2r\rho_c T}, \qquad \frac{T+1}{2} \sim \mathrm{Beta}\left(\frac{d-1}{2}, \frac{d-1}{2}\right). \tag{56}$$

Here, T denotes the first coordinate of a uniformly distributed point on $\mathbb{S}^{d-1}$. The reference-reference expectation follows from the same expression with $r = \rho_c$. We evaluate these one-dimensional expectations using 64-point Gauss–Jacobi quadrature and compute the empirical pairwise term exactly in blocks. A larger S-MMD^2 indicates that the sample distribution is less sphere-like.

Table 8: Hyperparameter settings for `Drift-RAE`.

Hyperparameter	Value
N_{class}	32
N_{pos}	256
N_{neg}	64
$N_{\text{extra_neg}}$	192
Temperatures τ	$\{0.02,\ 0.05,\ 0.2\}$
Training steps T	10,000
Learning rate schedule	$\max\left\{3\times10^{-7},\ 3\times10^{-5}-\frac{(3\times10^{-5}-3\times10^{-7})}{10000}\cdot t\right\}$
Gradient clipping	1.0
Optimizer	AdamW ($\beta_1=0.9,\ \beta_2=0.95$)
Weight decay	0.0

Table 9: Ablation on the intermediate layers used by IDG at $\omega = 1.4$.

Layers for Internal Drifting Guidance															
8	✓				✓	✓	✓				✓	✓	✓		✓
10		✓			✓			✓	✓		✓	✓		✓	✓
12			✓			✓		✓		✓	✓		✓	✓	✓
14				✓			✓		✓	✓		✓	✓	✓	✓
FID (↓)	1.54	1.56	1.55	1.52	1.49	1.56	**1.48**	1.53	1.54	1.50	1.60	1.56	1.55	1.57	1.52

B.2 Hyperparameter Settings

We summarize the main hyperparameters in Table 8. Positive samples are drawn from per-class FIFO caches, each holding N_{pos} samples and updated with the corresponding examples from each incoming data batch. We omit the exponential moving average (EMA) of model parameters, as it does not yield consistent improvement. All experiments are conducted on 32 NVIDIA H20 GPUs.

For IDG, we use the intermediate features from layers 8 and 14 by default and concatenate them along the feature dimension. The concatenated features are projected by a three-layer MLP with two hidden layers of width 2048, SiLU activations, and an output dimension of 768. The head is trained with the drifting loss that uses the backbone predictions as positive references and the same hyperparameters as the backbone. Both the intermediate features and backbone references are detached, so the head loss updates only the projection head. For results with IDG, we use $\omega = 1.4$ by default.

For the generated CLS tokens, we directly take the CLS tokens of real samples produced by the frozen DINO encoder to compute the drifting field. Following Darcet et al. (2024), we also append four extra register tokens to the Transformer token sequence. These extra tokens have no explicit target or auxiliary loss and simply provide redundancy during training and inference.

B.3 Additional Ablations

Here we further ablate the additional tokens and the implementation details of IDG. All results are evaluated on ImageNet 256×256 with distilled DiT^{DH}-XL.

CLS and registers. Table 10 reports the comparison of FID scores with and without the additional class token and registers. The inclusion of the additional tokens leads to slight improvement across guidance scales.

Table 10: Ablation on CLS and registers.

CLS + registers	**FID** (↓)	
	$\omega = 1.0$	$\omega = 1.4$
✗	1.77	1.56
✓	**1.74**	**1.48**

Intermediate layers for IDG. Table 9 compares all combinations of layers $\{8, 10, 12, 14\}$ for IDG. Using two intermediate layers yields slightly better results than using a single layer, while adding more layers does not yield consistent improvement. Layers $\{8, 14\}$ achieve the best FID.

Table 13: Attempts to train drifting models from scratch without auxiliary MAEs.

Latent space	Model	Strategy	FID (↓)
SD-VAE	DiT-XL	–	220.55
RAE	DiT^{DH}-XL	–	100.31
RAE	DiT^{DH}-XL	+ decode-encode	7.04

Training schedule for IDG. We also evaluate a two-stage training schedule as applied in Fu et al. (2026). Specifically, we first distill the backbone and then train the IDG head in a dedicated stage with the frozen backbone. As shown in Table 11, this two-stage schedule achieves an FID of 1.50 compared to 1.48 of our joint training strategy. Given the comparable performance, we adopt the joint training paradigm as it eliminates the need for an additional training stage.

Table 11: Ablation on the IDG training schedule with $\omega = 1.4$.

Training schedule	FID (↓)
Two-stage	1.50
One-stage	**1.48**

Training loss for guidance head. We use drifting loss to train the guidance head with distribution-level supervision from the backbone predictions by default. To examine this choice, we also train the guidance head with MSE and cosine similarity losses, using the backbone outputs as targets. As shown in Table 12, drifting achieves a marginally lower FID than MSE, whereas cosine similarity alone fails to provide effective supervision to the guidance head, despite its use in REPA (Yu et al., 2025). Since cosine similarity completely ignores magnitude information, we introduce a modified loss as

Table 12: Ablation on the training loss for the guidance head ($\omega = 1.4$).

Training loss	FID (↓)
Drifting	**1.48**
MSE	1.51
Cosine sim.	267.65
Cosine sim. + norm	1.49

$$\mathcal{L}_{\text{Cosine sim.+norm}}(\mathbf{x}, \mathbf{y}) = -\frac{\min\{\|\mathbf{x}\|, \|\mathbf{y}\|\}}{\max\{\|\mathbf{x}\|, \|\mathbf{y}\|\}} \cos\angle(\mathbf{x}, \mathbf{y}). \tag{57}$$

Equation (57) is minimized when the cosine similarity converges to 1 while the norms match. Applying the modified loss leads to FID closer to drifting. These results suggest that the guidance head does not necessarily require the drifting objective, but preserving magnitude information is important. We retain drifting as the default because, in addition to its marginal empirical advantage, it also provides distribution-level supervision consistent with the backbone distillation objective.

C Attempts to Train Drifting Models from Scratch with RAEs

We also explore whether Drifting Models can be trained from scratch without auxiliary MAEs. As shown in Table 13, directly training Drifting Models in either SD-VAE or RAE latent spaces without an additional MAE fails to produce effective results. This difficulty is consistent with our analysis in Section 3.2: dispersed initial predictions can weaken the spatial selectivity of the negative drifting field, making effective supervision harder to establish from scratch.

To alleviate this issue, we further try a simple decode-encode strategy. Specifically, we first decode the generated latents using the RAE decoder, and then re-encode the decoded samples with the RAE encoder to compute the Drifting direction and apply gradient backpropagation. We denote this variant as "+ decode-encode". As shown in Table 13, this strategy enables training from scratch in RAE spaces to a certain extent and achieves an FID of 7.04. We hypothesize that the decode-encode process can effectively project off-manifold generated samples back toward the data manifold, thereby alleviating the excessive dispersion caused by poor initialization. Despite this improvement, training Drifting Models from scratch in RAE spaces without an auxiliary MAE still lags behind state-of-the-art methods. Developing more informative drifting supervision during early training remains an important direction for improving MAE-free generation from scratch.

D More Visualizations

Additional class-wise qualitative results are shown in Figures 4 to 6.



Class 1: goldfish

Class 3: tiger shark



Class 12: house finch



Class 14: indigo bunting



Class 100: black swan



Class 127: white stork



Class 129: spoonbill



Class 141: redshank



Class 153: Maltese dog



Class 222: kuvasz

Figure 4: Additional visualizations of generated samples from distilled DiT^{DH}-XL (FID=1.48).



Class 235: German shepherd



Class 270: white wolf



Class 294: brown bear



Class 324: cabbage butterfly



Class 387: red panda



Class 407: ambulance



Class 425: barn



Class 437: beacon



Class 780: schooner



Class 888: viaduct

Figure 5: Additional visualizations of generated samples from distilled DiT^{DH}-XL (FID=1.48).



Class 908: wing

Class 928: ice cream



Class 930: French loaf



Class 933: cheeseburger



Class 959: carbonara



Class 967: espresso



Class 970: alp



Class 976: promontory, headland



Class 979: valley



Class 985: daisy

Figure 6: Additional visualizations of generated samples from distilled DiT^{DH}-XL (FID=1.48).