BEYOND RANDOM COUPLINGS: SPHERICAL NOISE ALIGNMENT IN GENERATIVE FLOWS

Anonymous authors
Paper under double-blind review

ABSTRACT

Flow matching models are typically trained using independent noise-data couplings, forcing the network to resolve conflicting conditional fields by learning highly curved velocity trajectories. This leads to slow iterative sampling, as numerical ODE solvers require many Neural Function Evaluations (NFEs) to track these paths accurately. While Minibatch Optimal Transport (OT) straightens the vector field by pairing noise and data samples to minimize transport cost, its reliance on discrete matching within small batches leaves it fundamentally bottlenecked by high-dimensional estimation variance. To overcome this limitation, we introduce continuous coupling via noise optimization, a framework that extends beyond discrete index matching towards continuously shaping noise endpoints. As a concrete realization, we present *Spherical Noise Alignment (SNA)*, an online algorithm that rotates source samples through balanced attractive and repulsive forces. The framework aligns noise samples with target data points during training while preserving prior fidelity. This allows seamless sampling from standard Gaussian noise at inference. Experimental results show that our approach improves source-target alignment, significantly reduces path curvature, and achieves superior sample fidelity in few-step (1–4 NFE) sampling regimes.

1 INTRODUCTION

Modern flow-matching pipelines (Lipman et al., 2023; Albergo et al., 2025; Liu et al., 2022) are commonly trained by coupling data samples with independently drawn Gaussian noise. While such couplings preserve the Gaussian prior, they disregard the geometry of the underlying distributions: nearby data samples can be paired with widely separated noise endpoints, inducing disparate transport directions within the same data neighborhood. This places the burden of discovering useful transport structure entirely on the network, which must reconcile arbitrary noise–data assignments into a coherent velocity field. The resulting learning problem can exhibit higher gradient variance and yield curved sampling trajectories, increasing the computational cost of generation.

To straighten transport trajectories, recent methods restructure training couplings. Minibatch Optimal Transport (OT) (Pooladian et al., 2023; Tong et al., 2024) computes linear assignment permutations within local training minibatches. However, discrete matching operates on static noise draws; it cannot adjust vector coordinates to fill coverage voids, leaving it constrained by high-dimensional estimation variance. Alternatively, global OT, semi-discrete formulations (Mousavi-Hosseini et al., 2026; Kong et al., 2025), and normalizing-flow couplings (Berthelot et al., 2026) improve path alignment, but require costly offline pre-computations or auxiliary network training.

To overcome these constraints, we introduce *Spherical Noise Alignment (SNA)*, a training-time framework that constructs flow-matching couplings by continuously optimizing noise endpoints toward data targets. By optimizing an objective that simultaneously enforces noise–data alignment and adherence to the Gaussian prior, SNA produces couplings that flatten the learned velocity field. To support this optimization, we derive a tractable regularizer that efficiently preserves the Gaussianity of the noise endpoints. Our core insight lies in decomposing the Gaussian KL divergence into three terms: angular mismatch, radial mismatch, and radius–direction dependence. Satisfying the radial and independence constraints by construction leaves angular uniformity as the sole distributional objective. We show that this objective can be efficiently optimized by a noise-noise contrastive loss.

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.

Our results demonstrate that SNA drastically improves sampling efficiency across pixel-space (CIFAR-10, ImageNet32) and latent-space (FFHQ256, ImageNet256) benchmarks. In few-step regimes (2–4 NFEs), SNA achieves up to 50% FID reduction compared to independent flow matching and up to 38% improvement over Minibatch OT baselines, with minimal training overhead and no additional computation at inference.

In summary, our key contributions are:

- We propose continuous noise coupling, a general formulation that optimizes noise endpoints toward data targets while preserving Gaussian prior fidelity.
- We derive a tractable Gaussian regularizer based on contrastive noise-noise repulsion, leading to Spherical Noise Alignment (SNA)—a practical realization of continuous noise couplings that requires fewer sampling steps to generate high-quality samples.

2 Related Work

Efforts to improve inference efficiency in flow-based generative models can be broadly organized into four distinct lines of research: optimizing noise–data couplings, modifying target training objectives, distilling flow trajectories, or optimizing the source prior distribution. While approaching the problem from different angles, all of these paradigms share a core underlying objective: to minimize the curvature of the generative trajectories. Since inference requires numerical integration, straighter trajectories are crucial for maintaining accuracy when integrating in very few steps. Because these approaches modify different stages of the generative pipeline, they are largely complementary.

Flow Matching and Optimized Couplings. Continuous normalizing flows and diffusion models have driven recent advances in generative modeling (Ho et al., 2020; Song et al., 2021; Lipman et al., 2023). Flow matching simplifies training by regressing vector fields along linear probability paths (Liu et al., 2022). Under independent couplings, random noise–data pairings can produce conflicting conditional velocity targets in nearby regions, complicating velocity field regression. Minibatch OT (Tong et al., 2024; Pooladian et al., 2023; Davtyan et al., 2025) addresses this issue by permuting noise–data assignments within sampled batches to reduce transport cost. Chiang et al. (2026) incorporate semantic structure by restricting these assignments to clusters, while Lin et al. (2025) construct couplings aligned with the network's preferred transport directions. Recent semi-discrete formulations (Kong et al., 2025; Mousavi-Hosseini et al., 2026) attempt global alignments between the continuous prior and discrete data points. However, these methods remain biased by finite sample sizes, or, as in semi-discrete formulations, incur heavy offline pre-computation.

Target Objectives. One prominent approach to reducing coupling curvature is to guide the network to learn a smoother target vector field directly through modified training objectives. Frameworks such as Consistency Models (Song et al., 2023; Yang et al., 2024; Geng et al., 2024) and Shortcut Models (Frans et al., 2025) enforce trajectory self-consistency, whereas MeanFlow (Geng et al., 2025) introduces the concept of *average velocity* to serve as the training target. Alongside these loss modifications, distillation techniques aim to compress complex integration paths into highly efficient few-step samplers (Yin et al., 2024; Luo et al., 2025). Because these techniques modify the network's regression objective, they leave the foundational noise-data geometry unchanged.

Model Distillation. Distillation techniques aim to compress complex integration paths into highly efficient few-step samplers (Salimans & Ho, 2022; Yin et al., 2024; Luo et al., 2025; Boffi et al., 2025; Dao et al., 2026). As a notable example, Rectified Flow (ReFlow) straightens trajectories by retraining on its own generated pairs (Liu et al., 2022). Complementarily, Berthelot et al. (2026) use an invertible normalizing flow model to provide stronger coupling supervision. However, distillation requires pre-trained models whose own training is dependent on prior coupling assumptions.

Optimized Source Distributions. Recent methods also explore adapting the initial source distribution to reduce path curvature. Building upon Lee et al. (2023), Nayal et al. (2026) propose *MixFlow*, which straightens trajectories by mixing conditioned and unconditional sources, while Kim et al. (2026) learn a prior that adapts to specific conditioning variables. In another approach, Malnick et al. (2026) redesign the prior distribution using low-frequency image projections.

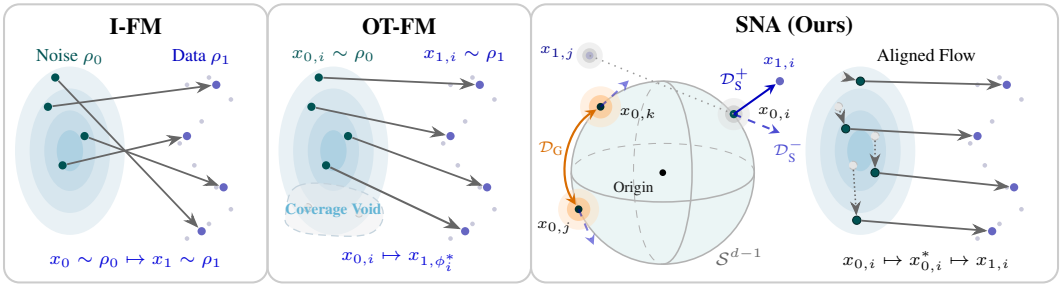


Figure 1: **Coupling Comparison in Flow Matching.** Flow Matching with Independent Couplings **(I-FM)** *(left)* randomly pairs noise $x_0 \sim \rho_0$ with data $x_1 \sim \rho_1$, producing heavily intersecting paths. Flow Matching with Minibatch Optimal Transport **(OT-FM)** (*middle*) aligns noise samples via discrete permutations, but remains bottlenecked as finite points fail to represent the entire noise distribution (*Coverage Void*). **SNA** (*right*) solves this issue: it continuously optimizes noise samples using attractive ($\mathcal{D}_S^+$) and repulsive ($\mathcal{D}_S^-, \mathcal{D}_G$) forces. This results in much better aligned couplings.

3 Preliminaries

3.1 Conditional Flow Matching and Linear Interpolants

Conditional Flow Matching (Lipman et al., 2023; Albergo et al., 2025; Liu et al., 2022; Tong et al., 2024) provides a simulation-free approach to train continuous normalizing flows by constructing a time-dependent probability path between a standard Gaussian noise prior $\rho_0 = \mathcal{N}(0, \mathbf{I}) \in \mathcal{P}(\mathbb{R}^d)$ and a data distribution $\rho_1 = \rho_{\text{data}} \in \mathcal{P}(\mathbb{R}^d)$. Formally, let $\Pi(\rho_0, \rho_1) \subset \mathcal{P}(\mathbb{R}^d \times \mathbb{R}^d)$ denote the set of joint distributions (couplings) that have ρ_0 and ρ_1 as their marginals. Given a paired sample $(x_0, x_1) \sim \pi$, we define the *linear interpolant map* $\phi_t : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ as:

$$x_t := \phi_t(x_0, x_1) = (1-t)x_0 + tx_1, \quad t \in [0,1]. \tag{1}$$

The continuous probability path $\rho_t \in \mathcal{P}(\mathbb{R}^d)$ is then defined via the pushforward operation $\rho_t = (\phi_t)_{\#}\pi$, which satisfies boundary conditions $\rho_t|_{t=0} = \rho_0$ and $\rho_t|_{t=1} = \rho_1$ by construction.

The dynamics of this interpolant are governed by a probability flow ODE driven by a time-varying marginal vector field v_t (Albergo et al., 2025):

$$dx_t = v_t(x_t)dt, \quad x_0 \sim \rho_0. \tag{2}$$

Directly learning v_t is intractable because conditioning on the intermediate state x_t via Bayes' rule requires dividing by the unknown marginal density ρ_t. Instead, Flow Matching relies on constructing tractable *conditional* vector fields $u_t(x_t \mid x_0, x_1)$, which for the linear path simplifies to a constant velocity $x_1 - x_0$. The marginal vector field is then recovered via marginalization: $v_t(x) = \mathbb{E}_\pi[x_1 - x_0 \mid x_t = x]$.

To learn this target vector field, a neural network $v_\theta(x_t, t)$ is optimized to predict the flow by minimizing the conditional flow matching objective:

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{\substack{t\sim\mathcal{U}[0,1] \\ (x_0,x_1)\sim\pi}} \left[\|v_\theta(x_t,t) - (x_1 - x_0)\|_2^2 \right]. \tag{3}$$

Once v_θ accurately models the true field v_t, we can map pure noise into the target distribution by evolving Equation 2 using numerical integration.

3.2 Marginal Field Curvature and Independent Couplings

Although the conditional trajectory for any specific pair (x_0, x_1) is a straight line with constant velocity $x_1 - x_0$, the resulting *marginal* vector field v_t induces highly curved sampling trajectories (Liu et al., 2022; Albergo et al., 2025). This phenomenon arises directly from the marginalization process $v_t(x) = \mathbb{E}_\pi[x_1 - x_0 \mid x_t = x]$. Because the probability flow ODE dictates a deterministic vector field where paths cannot cross, any intermediate spatial point x_t where multiple independent

linear paths intersect forces the marginal vector field to compute a weighted average of their conflicting conditional velocities. This averaging of competing directions causes the overall marginal vector field to curve.

The severity of this curvature is highly dictated by the choice of coupling π. An independent coupling, $\pi(x_0, x_1) = \rho_0(x_0)\rho_1(x_1)$, pairs samples entirely at random (I-FM). This stochastic construction maximizes the number of conflicting paths, forcing the network to average out opposing directions, and resulting in highly curved trajectories.

3.3 Minibatch Optimal Transport and its Limitations

To mitigate this, previous works (Tong et al., 2024; Pooladian et al., 2023) utilize *Optimal Transport* (OT) to replace independent pairings with a coupling π_{OT} that minimizes the expected transport distance between the noise prior ρ_0 and the target data distribution ρ_1. To make this tractable, standard practice relies on approximating the OT map discretely per minibatch (OT-FM). Given a sampled noise batch $\mathbf{x}_0 = \{x_{0,i}\}_{i=1}^N \sim \rho_0^{\otimes N}$ and a data batch $\mathbf{x}_1 = \{x_{1,i}\}_{i=1}^N \sim \rho_1^{\otimes N}$, where $\rho^{\otimes N}$ denotes the joint distribution of N independent samples drawn from ρ, Minibatch OT finds the optimal permutation $\phi^* \in \mathfrak{S}_N$ (where $\mathfrak{S}_N$ is the symmetric group of all permutations over N elements) such that the total squared Euclidean distance is minimized:

$$\phi^* = \arg\min_{\phi \in \mathfrak{S}_N} \sum_{i=1}^{N} \|x_{0,i} - x_{1,\phi_i}\|_2^2 . \tag{4}$$

The model then trains on the paired trajectories $x_{0,i} \to x_{1,\phi_i^*}$. Although this empirically straightens the marginal vector field, it is fundamentally limited by both the statistical bias of minibatch pairings (Fatras et al., 2021) and the high sample complexity of empirical OT (Fournier & Guillin, 2013; Hütter & Rigollet, 2020). Specifically, the rate at which empirical OT estimates converge to the continuous target scales as $\mathcal{O}(N^{-1/d})$ (Chewi et al., 2024). Recovering the true global OT map would therefore require a sample size that scales exponentially with dimension d. This is a direct consequence of the *curse of dimensionality*.

Unfortunately, resolving this statistical issue simply by increasing batch size N is computationally prohibitive. Exact discrete OT algorithms require $\mathcal{O}(N^3)$ time, whereas entropy-regularized solvers such as the Sinkhorn algorithm still scale quadratically as $\mathcal{O}(N^2/\epsilon^2)$ (Sinkhorn, 1964). Discrete index matching is thus fundamentally trapped between statistical variance and computational cost.

4 Spherical Noise Alignment

4.1 Continuous Coupling via Noise Optimization

Our first core insight is to reformulate flow-matching couplings as a continuous optimization task over a batch. Specifically, we seek optimal noise endpoints $\mathbf{x}_0^*$ that maximize alignment with a target data batch $\mathbf{x}_1$ while preserving Gaussianity by solving

$$\mathbf{x}_0^* = \arg\min_{\mathbf{x}_0} \lambda_S \mathcal{D}_{\text{S}}(\mathbf{x}_0, \mathbf{x}_1) + \mathcal{D}_{\text{G}}(\mathbf{x}_0), \tag{5}$$

where $\mathcal{D}_{\text{S}}$ quantifies alignment between the noise and data batches (weighted by λ_S), and $\mathcal{D}_{\text{G}}$ acts as a regularizer ensuring the optimized noise batch remains faithful to the Gaussian prior.

In practice, this changes the forward process of a training flow-matching model by computing the optimized noise batch $\mathbf{x}_0^* = \mathbf{x}_0^{(K)}$ online via K inner-loop gradient steps with step size η:

$$\mathbf{x}_0^{(k+1)} = \mathbf{x}_0^{(k)} - \eta \nabla_{\mathbf{x}_0^{(k)}} \left[\lambda_S \mathcal{D}_{\text{S}}(\mathbf{x}_0^{(k)}, \mathbf{x}_1) + \mathcal{D}_{\text{G}}(\mathbf{x}_0^{(k)}) \right], \quad \text{where } \mathbf{x}_0^{(0)} \sim \rho_0. \tag{6}$$

Given the optimized endpoints $\mathbf{x}_0^*$, the flow network v_θ is subsequently trained on the resulting straightened conditional vector field $u_t(x_t \mid x_{0,i}^*, x_{1,i}) = x_{1,i} - x_{0,i}^*$. By continuously adjusting initial noise positions during training, this framework bypasses the empirical sample complexity limit of discrete OT. At test time, the model samples directly from standard Gaussian draws $x_0 \sim \mathcal{N}(0, \mathbf{I})$ at zero additional computational cost. The alignment term $\mathcal{D}_{\text{S}}$ can be specified directly through similarities between paired samples, as discussed in Section 4.3. However, defining $\mathcal{D}_{\text{G}}$ is more challenging.

4.2 SPHERICAL GAUSSIANIZATION

Gaussianity is a distributional property: a single noise vector does not reveal the distribution that produced it. Inferring a distributional property from a single realization lies at the heart of the *hard problem of Gaussianity* (Chang et al., 2026). To make this criterion tractable, previous work associates each sample x_i with a surrogate distribution $\hat{Q}_{x_i}$ computed from its spatial statistics. This approximates Gaussianity via a single-sample divergence estimate, which we denote as $\tilde{\mathcal{D}}_{\mathrm{G}}(x_i)$ to distinguish it from our later batch-level objective:

$$\tilde{\mathcal{D}}_{\mathrm{G}}(x_i) \approx D_{\mathrm{KL}}\left(\hat{Q}_{x_i} \,\|\, G\right) = \int_{y\sim x_i} \hat{Q}_{x_i}(y) \log \frac{\hat{Q}_{x_i}(y)}{G(y)}\, dy, \tag{7}$$

where $y = (y_0, y_1)$ contains the values of two adjacent pixels, and $G(y)$ is the standard bivariate Gaussian density. Under an ergodic Markov random field model with local cliques, the method approximates the divergence using KDE estimates of unary and neighboring-pixel pair statistics across multiple spatial scales.

Directly applying this approach to flow matching coupling introduces two limitations. First, estimating and differentiating multiscale pixel-level densities for every sample is very inefficient and adds significant computational overhead to each inner-loop update during training. Second, the regularizer evaluates spatial statistics within individual samples, without explicitly controlling the collective arrangement of the noise endpoints. Consequently, endpoints can satisfy the samplewise statistical constraints while concentrating in similar regions of noise space. In the extreme case, repeating a single admissible noise vector across the batch leaves the average samplewise penalty unchanged, despite eliminating all directional diversity. Our batch-level objective (Equation (5)) therefore calls for a regularizer that accounts for interactions between endpoints and discourages such concentration.

These limitations motivate a Gaussianity measure $\mathcal{D}_{\mathrm{G}}$ that captures statistics across whole noise samples. Our core intuition is to address these requirements by replacing local pixel-statistics estimation with density estimation over a batch of noise directions on the unit hypersphere. By assigning independent χ-distributed radii, Gaussianization then reduces to making the directional distribution uniform. Specifically, for $x \sim Q$, we write $x = ru$, where $r = \|x\|_2$ and $u = x/\|x\|_2$. Let $Q_{u,r}$ denote their joint distribution, with angular and radial marginals Q_u and Q_r, respectively. By exploiting invariance under coordinate transformations (Appendix A), the KL divergence decomposes as

$$D_{\mathrm{KL}}(Q \,\|\, \mathcal{N}(0, I_d)) = \underbrace{D_{\mathrm{KL}}(Q_u \,\|\, \sigma)}_{\text{Angular mismatch}} + \underbrace{D_{\mathrm{KL}}(Q_r \,\|\, \chi_d)}_{\text{Radial mismatch}} + \underbrace{D_{\mathrm{KL}}(Q_{u,r} \,\|\, Q_u \otimes Q_r)}_{\text{Radius–direction dependence}}, \tag{8}$$

where $\sigma = \mathrm{Unif}(\mathbb{S}^{d-1})$ is the uniform probability measure on the unit hypersphere; χ_d is the distribution that models the norm of a standard Gaussian vector, and $Q_u \otimes Q_r$ measures the product of the marginals. The final term is exactly the mutual information between direction and radius.

Although Equation (8) now has three terms, our algorithm satisfies the radial and independence constraints by construction, leaving only the angular term to optimize. Our algorithm updates unit directions $u_i \in \mathbb{S}^{d-1}$ using projected gradient descent. After optimization, we draw radii $r_i \sim \chi_d$ independently of the optimized directions and reconstruct the endpoints as $x_i = r_i u_i$. This eliminates both the radial mismatch and radius–direction dependence terms. Gaussianization therefore reduces to driving the directional density toward uniformity on the hypersphere; implementation details and proofs are provided in the Appendix B.

To approximate the angular KL divergence, we have to estimate the surrogate distributional density $\hat{Q}_u$ relative to the noise directions in a batch. Each endpoint contributes a smooth kernel on the hypersphere, and averaging these contributions yields a continuous density estimate that captures angular concentration across the batch. We use a von Mises–Fisher (vMF) kernel, whose contribution from a sample direction u to a query direction v with bandwidth parameter γ is

$$\mathcal{K}(u, v, \gamma) = \frac{e^{\gamma^{-1}\langle u, v\rangle}}{c_{\gamma,d}}, \quad \text{with } c_{\gamma,d} = \int_{\mathbb{S}^{d-1}} e^{\gamma^{-1}\langle u, w\rangle}\, d\sigma(w) \tag{9}$$

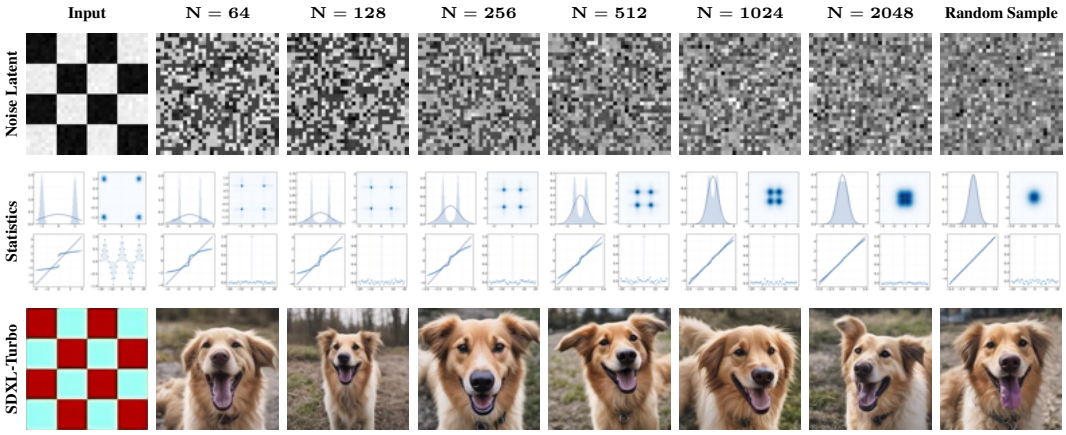


Figure 2: **Recovery of Gaussian noise.** Even when starting from a highly non-Gaussian input, the Gaussian regularizer D_G successfully recovers the isotropic Gaussian distribution as the batch size N increases. Top: optimized latent $\mathbf{x}_0^*$. Middle: Gaussianity diagnostics (*1D Histogram, 2D Density, Q-Q Plot, and Spatial Autocorrelation*). Bottom: corresponding generated samples using SDXL-Turbo. During inference, diffusion models are surprisingly robust to non-Gaussian statistics, provided the noise latent lacks clear structure.

Then, we use the vMF kernel $\mathcal{K}(u, v, \gamma)$ to numerically approximate the spherical density with a leave-one-out summation to avoid self density contributions as

$$\hat{Q}_u(u_i) \approx \frac{1}{N-1} \sum_{j \neq i} \mathcal{K}(u_j, u_i, \gamma). \tag{10}$$

Lastly, we have to evaluate the angular mismatch term from Equation (8) with the estimated per-sample density of Equation (10). The KL divergence can be also written as an expectation over the true distribution Q_u. We approximate this expectation using a standard Monte Carlo estimator over the empirical batch samples $u_i \sim Q_u$:

$$D_{\mathrm{KL}}(\hat{Q}_u \,\|\, \sigma) = \mathbb{E}_{u \sim Q_u} \left[\log \frac{\hat{Q}_u(u)}{\sigma(u)} \right] \approx \frac{1}{N} \sum_{i=1}^{N} \left(\log \hat{Q}_u(u_i) - \log \sigma(u_i) \right). \tag{11}$$

Since the uniform measure $\sigma(u_i)$ is constant everywhere on the sphere, it becomes an additive constant C. Substituting the KDE definition into the remaining term yields a log-sum-exponential objective:

$$D_{\mathrm{KL}}(\hat{Q}_u \,\|\, \sigma) \approx \frac{1}{N} \sum_{i=1}^{N} \log \sum_{j \neq i} \frac{e^{\gamma^{-1} \langle u_j, u_i \rangle}}{c_{\gamma,d}}. \tag{12}$$

Discarding additive constants that do not affect the gradient with respect to u_i, we obtain our empirical objective for the batch $\mathbf{x}_0$:

$$\mathcal{D}_{\mathrm{G}}(\mathbf{x}_0) := \frac{1}{N} \sum_{i=1}^{N} \log \sum_{j \neq i} e^{\gamma^{-1} \langle u_j, u_i \rangle}. \tag{13}$$

The resulting log-sum-exponential objective coincides with the contrastive denominator associated with InfoNCE (Oord et al., 2019; Chen et al., 2020; Wang & Isola, 2022; Betser et al., 2026). Thus, our work efficiently connects contrastive repulsion to KL-based Gaussian prior regularization.

We demonstrate that minimizing $\mathcal{D}_{\mathrm{G}}$ effectively restores Gaussian statistics by conducting an experiment starting from structured, highly non-Gaussian latents. As illustrated in Figure 2, optimizing $\mathcal{D}_{\mathrm{G}}$ with increasing batch sizes N progressively eliminates spatial autocorrelation and aligns the measured statistics with the target Gaussian distribution.

4.3 Sample and Noise Alignment Term ($\mathcal{D}_S$)

For alignment, we formulate a contrastive loss between the normalized noise directions $u_{0,i} = x_{0,i}/\|x_{0,i}\|_2$ and their corresponding data target directions $u_{1,i} = x_{1,i}/\|x_{1,i}\|_2$. This encourages noise endpoints to rotate toward data directions, with each endpoint's contribution weighted by its angular proximity to the target.

$$\mathcal{D}_S(\mathbf{x}_0, \mathbf{x}_1) := -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{\tau^{-1}\langle u_{0,i}, u_{1,i}\rangle}}{\sum_{j=1}^{N} e^{\tau^{-1}\langle u_{0,i}, u_{1,j}\rangle}}, \tag{14}$$

where $\tau > 0$ is a separate temperature parameter. Because the objective depends exclusively on normalized unit vectors, it acts purely on orientation; our sphere-constrained updates and independent post-hoc radial sampling preserve the exact χ_d radial distribution. The denominator exerts a repulsive force against all non-assigned data targets. This prevents the noise vector from degenerating into the broader image manifold and forces it to isolate features strictly unique to its assigned target.

5 Experiments

5.1 Unconditional Pixel-Space Image Generation with U-Net

We evaluate different coupling strategies on two 32×32 pixel-space benchmarks: CIFAR-10 (50k images) and ImageNet-32 (≈ 1.28M images). Following the evaluation protocols of recent coupling papers (Kong et al., 2025; Mousavi-Hosseini et al., 2026; Malnick et al., 2026), we adopt the U-Net architecture and setup from Tong et al. (2024) for CIFAR-10, while for ImageNet32, we follow Pooladian et al. (2023). Within each data set, I-FM, OT-FM, and SNA training differ exclusively in the way noise-data pairs are sampled, all other aspects of FM training remain identical. Here, SNA + OT denotes initialization via minibatch OT (Eq. 4). All baseline numbers are reproduced. During inference, we sample from the exact Gaussian prior $x_0 \sim \mathcal{N}(0, \mathbf{I})$. We measure sample quality via Fréchet Inception Distance (FID) (Heusel et al., 2018), and path straightness via flow curvature (Lee et al., 2023). To test few-step generation, we report FID using a deterministic Euler solver across varying number of function evaluations (NFEs). Full implementation details are provided in Appendix B.

Our pixel-space results on CIFAR-10 and ImageNet32 are summarized in Table 2. Across both benchmarks, SNA consistently improves sample quality over independent coupling (I-FM) and Minibatch OT (OT-FM) baselines. Performance gains are most pronounced in few-step regimes. On CIFAR-10, SNA+OT (alignment weight $\lambda_S = 1/2$) achieves a 2-step FID of 56.40: an $\sim$68% reduction compared to I-FM (175.91) and an $\sim$38% improvement over OT-FM (91.75). On ImageNet32, SNA similarly delivers strong performance gains on a significantly larger dataset.

Table 1: **Trajectory curvature (↓).** SNA reduces curvature across both datasets.

Method	CIFAR-10	ImageNet32
I-FM	0.0479	0.0600
OT-FM	0.0326	0.0416
SNA (ours, $\lambda_S = 1/2$)	0.0324	0.0412
SNA + OT (ours, $\lambda_S = 1/5$)	**0.0251**	**0.0381**

These improvements are directly explained by the straightened probability flows. As shown in Table 1, SNA consistently achieves the lowest trajectory curvature across both benchmarks, reducing curvature relative to OT-FM by $\sim$23% on CIFAR-10 and $\sim$8% on ImageNet32. Qualitative samples reflect these metrics, producing much sharper images at low NFEs (Figure 5).

5.2 Latent-Space Image Generation using SiT

We next evaluate SNA on 256×256 resolution using Scalable Interpolant Transformers (SiT) (Ma et al., 2024). Following standard practice, images are encoded into $4 \times 32 \times 32$ latents using a pre-trained Stable Diffusion VAE (`sd-vae-ft-ema`). We test on two benchmarks: unconditional FFHQ256 (70k images) and class-conditional ImageNet256 ($\approx$ 1.28M images), keeping training settings identical across baselines. (See Appendix B for model and training details.) To address the *curse of conditions* in class-conditional OT-FM, we follow Cheng & Schwing (2025) and construct minibatch OT couplings strictly within individual classes using enlarged per-class batches. Similarly, we restrict SNA's contrastive loss to samples that share the same class label.

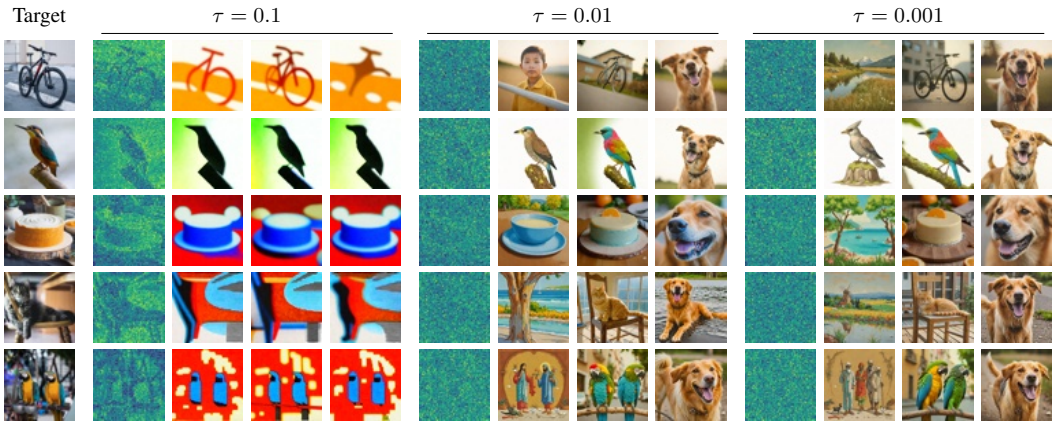


Figure 3: **Noise optimization across temperatures.** Left: target image. Remaining blocks evaluate alignment temperatures τ, showing optimized noise $\mathbf{x}_0^*$ ($K = 100$) alongside SDXL-Turbo generations for empty, matching, and non-matching ("dog") prompts. High temperatures ($\tau = 0.1$) push noise outside the Gaussian typical set, causing generation failure. The optimal regime ($\tau = 0.01$) successfully balances target retention with prompt generalization. While this confirms single-sample typicality, enforcing joint batch normality necessitates our repulsion term $\mathcal{D}_G$.

Table 2: **Main image generation benchmark results (FID-50k ↓).** Combined side-by-side comparison across pixel-space (left) and latent-space (right) benchmarks. SNA consistently improves sample quality over I-FM and OT-FM baselines. For ImageNet256, a guidance scale of 2.0 is used.

Unconditional Pixel-Space Generation						**Latent-Space Generation**					
Method	1-Step	2-Step	4-Step	8-Step	16-Step	Method	1-Step	2-Step	4-Step	8-Step	16-Step
CIFAR-10						**FFHQ256**					
I-FM	346.76	175.91	54.39	18.21	9.59	I-FM	386.29	151.25	81.66	33.65	17.03
OT-FM	229.98	91.75	30.00	14.20	9.08	OT-FM	295.11	118.56	70.38	29.92	15.73
SNA ($\lambda_S = 1/2$)	220.87	87.39	28.87	14.00	8.90	SNA ($\lambda_S = 1/3$)	286.92	119.33	64.35	**27.42**	**15.11**
SNA + OT ($\lambda_S = 1/2$)	**148.54**	**56.40**	**22.64**	**12.28**	**8.23**	SNA + OT ($\lambda_S = 1/5$)	**254.90**	**105.18**	**52.04**	29.14	17.35
ImageNet32						**ImageNet256**					
I-FM	419.92	205.99	69.51	25.05	12.14	I-FM	296.31	167.66	43.02	11.90	7.04
OT-FM	252.88	102.63	38.05	15.89	9.41	OT-FM	273.89	138.43	35.10	10.65	6.72
SNA ($\lambda_S = 1/2$)	250.87	105.67	39.93	17.85	10.89	SNA ($\lambda_S = 1/2$)	270.65	130.05	32.21	10.50	6.81
SNA + OT ($\lambda_S = 1/5$)	**166.07**	**71.78**	**26.86**	15.03	12.82	SNA + OT ($\lambda_S = 1$)	**244.59**	**114.72**	**29.98**	**10.26**	6.85
SNA + OT ($\lambda_S = 1/30$)	230.85	93.33	34.02	**14.78**	**8.96**	SNA + OT ($\lambda_S = 1/5$)	254.94	125.62	31.75	10.28	**6.65**

As summarized in Table 2, SNA scales effectively to latent representations. On FFHQ256, SNA + OT provides the largest boost in few-step regimes, improving 4-step FID by $\sim$26% over OT-FM (52.04 vs. 70.38). Similarly, on class-conditional ImageNet256, SNA + OT uniformly improves on all baselines in every sampling budget. It drops 1-step FID from 273.89 (OT-FM) to 254.94 and maintains its lead through 16 steps (6.65 FID).

5.3 Analysis

Component Ablations. We perform a progressive ablation on CIFAR-10 to evaluate the contribution of each component in our objective (Table 3). Applying only the cross-modal alignment force ($\mathcal{D}_S$) shifts noise samples toward the target data topology. The denominator in $\mathcal{D}_S$ provides implicit cross-modal repulsion; without this contrastive balance, a purely attractive pairwise loss would cause the optimization to diverge or collapse into degenerate points (Figure 3). Applying $\mathcal{D}_S$ alone substantially improves few-step generation. However, because $\mathcal{D}_S$ lacks explicit *unimodal* angular repulsion, noise samples still suffer from local clustering (prior hole problem). Incorporating $\mathcal{D}_G$ enforces uniform spherical dispersion across nega-

Table 3: **Progressive ablation** starting from the OT-FM baseline. Measured in FID-50k (↓).

Configuration	Euler-4	Euler-10	Dopri5
OT-FM *(baseline)*	30.00	12.13	3.66
+ $\mathcal{D}_S$ ($\tau = 0.1$)	25.32	15.81	15.84
+ $\mathcal{D}_S$ ($\tau = 0.01$)^{def}	**22.60**	12.89	10.52
SNA + OT ($\lambda_S\mathcal{D}_S + \mathcal{D}_G$, $\lambda_S = 1/5$)	25.24	**11.02**	**3.56**

tive pairs, ensuring the Gaussianity of the batch. Together, these components produce an optimal balance, achieving superior performance in both fixed-step and adaptive ODE solvers (Dopri5).

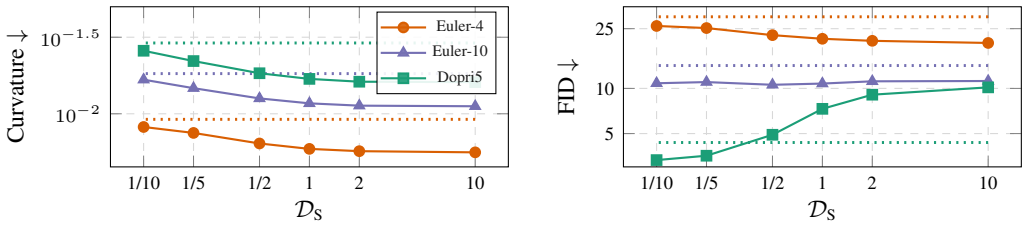


Figure 4: **Analysis on parameter $\mathcal{D}_S$.** Higher $\mathcal{D}_S$ reduce curvature and optimize few-step sampling, while lower $\mathcal{D}_S$ preserve prior fidelity, improving high-NFE sampling. Dotted lines denote OT-FM.

Effect of Structural Alignment Weight $\mathcal{D}_S$. Figure 4 shows how $\mathcal{D}_S$ balances path straightness with prior fidelity. We train different models with various values of $\mathcal{D}_S$. Keeping $\mathcal{D}_S$ large allows the noise samples to align closely with the data. This creates exceptionally straight paths, resulting in peak few-step performance. The downside is that the prior drifts from a true Gaussian, which hurts high-step quality (Dopri5 solver). In contrast, smaller $\mathcal{D}_S$ values reduce path straightness, but benefit Dopri5 performance.

Table 4: Inference profile of the $\mathcal{D}_S$-conditioned model on CIFAR-10. Measured in FID-50k (↓).

Inference $\mathcal{D}_S$	Euler-2	Euler-4	Euler-10	Dopri5
OT-FM	91.75	30.00	12.13	3.66
$\mathcal{D}_S = 2/3$	**55.04**	**22.19**	**10.48**	6.13
$\mathcal{D}_S = 1/3$	62.11	23.83	10.54	4.25
$\mathcal{D}_S = 1/6$	70.75	25.88	11.07	**3.49**

Post-Hoc Path Regulation. Interestingly, the flow matching network can be explicitly conditioned on the alignment parameter $\mathcal{D}_S$ during training. This turns $\mathcal{D}_S$ into a *post-hoc inference parameter*, allowing users to tune the trade-off between path straightness and prior adherence at generation time without retraining the model (for more details, see Table 4 and App. C.3).

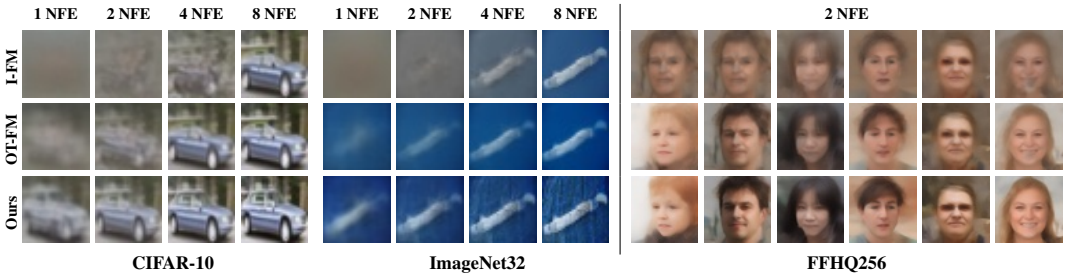


Figure 5: **Qualitative Results across different step counts.** Compared to the baselines, our method (SNA + OT) generates high-quality samples even at low NFEs.

6 Conclusion

We introduced *Spherical Noise Alignment (SNA)*, a coupling method that moves beyond discrete minibatch OT permutations to continuous noise optimization. By optimizing noise directions on the hypersphere through a contrastive objective, SNA straightens the learned marginal velocity field, enabling flow matching models to achieve substantial few-step performance gains at zero additional inference cost.

Limitations & Future Work. A key limitation of SNA is the tension between aggressive alignment and preserving Gaussian prior fidelity. Future work could explore refined regularization schemes. Other possible directions include *deterministic Gaussianization* (directly evolving data points into aligned noise), extending alignment and repulsion to semantic feature spaces, and further combining SNA with complementary trajectory-straightening techniques.

AI USE STATEMENT

In this work, we used generative AI tools solely for the refinement and polishing of this paper. We have not used generative AI tools for the core research, including the methodological formulation, experimental design, or empirical analysis. We have manually reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic Interpolants: A Unifying Framework for Flows and Diffusions, October 2025. URL http://arxiv.org/abs/2303.08797. arXiv:2303.08797.

David Berthelot, Tianrong Chen, Jiatao Gu, Marco Cuturi, Laurent Dinh, Bhavik Chandna, Michal Klein, Josh Susskind, and Shuangfei Zhai. The Coupling Within: Flow Matching via Distilled Normalizing Flows, March 2026. URL http://arxiv.org/abs/2603.09014. arXiv:2603.09014 [cs].

Roy Betser, Eyal Gofer, Meir Yossef Levi, and Guy Gilboa. InfoNCE Induces Gaussian Distribution, June 2026. URL http://arxiv.org/abs/2602.24012. arXiv:2602.24012.

Nicholas M. Boffi, Michael S. Albergo, and Eric Vanden-Eijnden. Flow map matching with stochastic interpolants: A mathematical framework for consistency models, June 2025. URL http://arxiv.org/abs/2406.07507. arXiv:2406.07507 [cs.LG].

Pascal Chang, Kai Lascheit, Jingwei Tang, Markus Gross, and Vinicius C Azevedo. What Is It Like to Be a Noise? An Entropy-based Gaussian Noise Regularization for Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 43471–43481, 2026.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations, July 2020. URL http://arxiv.org/abs/2002.05709. arXiv:2002.05709.

Ho Kei Cheng and Alexander Schwing. The Curse of Conditions: Analyzing and Improving Optimal Transport for Conditional Flow-Based Generation, August 2025. URL http://arxiv.org/abs/2503.10636. arXiv:2503.10636.

Sinho Chewi, Jonathan Niles-Weed, and Philippe Rigollet. Statistical Optimal Transport. November 2024. URL https://arxiv.org/pdf/2407.18163.

Chiensheng Chiang, Kuan-Hsun Tu, Jia-Wei Liao, Cheng-Fu Chou, and Tsung-Wei Ke. COT-FM: Cluster-wise Optimal Transport Flow Matching, March 2026. URL http://arxiv.org/abs/2603.13395. arXiv:2603.13395 [cs].

Quan Dao, Hao Phung, Trung Dao, Dimitris Metaxas, and Anh Tran. Self-Corrected Flow Distillation for Consistent One-Step and Few-Step Text-to-Image Generation, March 2026. URL http://arxiv.org/abs/2412.16906. arXiv:2412.16906 [cs.CV].

Aram Davtyan, Leello Tadesse Dadi, Volkan Cevher, and Paolo Favaro. FASTER INFERENCE OF FLOW-BASED GENERATIVE MODELS VIA IMPROVED DATA-NOISE COUPLING. 2025.

Kilian Fatras, Thibault Séjourné, Nicolas Courty, and Rémi Flamary. Unbalanced minibatch Optimal Transport; applications to Domain Adaptation, March 2021. URL http://arxiv.org/abs/2103.03606. arXiv:2103.03606 [cs.LG].

Nicolas Fournier and Arnaud Guillin. On the rate of convergence in Wasserstein distance of the empirical measure, December 2013. URL http://arxiv.org/abs/1312.2128. arXiv:1312.2128 [math.PR].

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One Step Diffusion via Shortcut Models, June 2025. URL http://arxiv.org/abs/2410.12557. arXiv:2410.12557.

Zhengyang Geng, Ashwini Pokle, William Luo, Justin Lin, and J. Zico Kolter. Consistency Models Made Easy, October 2024. URL http://arxiv.org/abs/2406.14548. arXiv:2406.14548 [cs.LG].

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean Flows for One-step Generative Modeling, May 2025. URL http://arxiv.org/abs/2505.13447. arXiv:2505.13447.

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, January 2018. URL http://arxiv.org/abs/1706.08500. arXiv:1706.08500.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models, December 2020. URL http://arxiv.org/abs/2006.11239. arXiv:2006.11239.

Jan-Christian Hütter and Philippe Rigollet. Minimax estimation of smooth optimal transport maps, June 2020. URL http://arxiv.org/abs/1905.05828. arXiv:1905.05828 [math.ST].

Junwan Kim, Jiho Park, Seonghu Jeon, and Seungryong Kim. Better Source, Better Flow: Learning Condition-Dependent Source Distribution for Flow Matching, May 2026. URL http://arxiv.org/abs/2602.05951. arXiv:2602.05951 [cs.CV].

Lingkai Kong, Molei Tao, Yang Liu, Bryan Wang, Jinmiao Fu, Chien-Chih Wang, and Huidong Liu. AlignFlow: Improving Flow-based Generative Models with Semi-Discrete Optimal Transport, October 2025. URL http://arxiv.org/abs/2510.15038. arXiv:2510.15038.

Sangyun Lee, Beomsu Kim, and Jong Chul Ye. Minimizing Trajectory Curvature of ODE-based Generative Models, May 2023. URL http://arxiv.org/abs/2301.12003. arXiv:2301.12003.

Yexiong Lin, Yu Yao, and Tongliang Liu. Beyond Optimal Transport: Model-Aligned Coupling for Flow Matching, May 2025. URL http://arxiv.org/abs/2505.23346. arXiv:2505.23346 [cs].

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow Matching for Generative Modeling, February 2023. URL http://arxiv.org/abs/2210.02747. arXiv:2210.02747.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow, September 2022. URL http://arxiv.org/abs/2209.03003. arXiv:2209.03003 [cs.LG].

Yihong Luo, Tianyang Hu, Jiacheng Sun, Yujun Cai, and Jing Tang. Learning Few-Step Diffusion Models by Trajectory Distribution Matching, March 2025. URL http://arxiv.org/abs/2503.06674. arXiv:2503.06674.

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, September 2024. URL http://arxiv.org/abs/2401.08740. arXiv:2401.08740 [cs.CV].

Shimon Malnick, Matan Rusanovsky, Ohad Fried, and Shai Avidan. Optimal Transport Flow Matching by Design, June 2026. URL http://arxiv.org/abs/2606.04092. arXiv:2606.04092 [cs.CV].

Alireza Mousavi-Hosseini, Stephen Y. Zhang, Michal Klein, and Marco Cuturi. Flow Matching with Semidiscrete Couplings, January 2026. URL http://arxiv.org/abs/2509.25519. arXiv:2509.25519.

Nazir Nayal, Christopher Wewer, and Jan Eric Lenssen. MixFlow: Mixed Source Distributions Improve Rectified Flows, April 2026. URL http://arxiv.org/abs/2604.09181. arXiv:2604.09181.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding, January 2019. URL http://arxiv.org/abs/1807.03748. arXiv:1807.03748.

Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky T. Q. Chen. Multisample Flow Matching: Straightening Flows with Minibatch Couplings, May 2023. URL http://arxiv.org/abs/2304.14772. arXiv:2304.14772.

Tim Salimans and Jonathan Ho. Progressive Distillation for Fast Sampling of Diffusion Models, June 2022. URL http://arxiv.org/abs/2202.00512. arXiv:2202.00512 [cs.LG].

Richard Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *Ann. Math. Statist.*, 35:876–879, 1964.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations, February 2021. URL http://arxiv.org/abs/2011.13456. arXiv:2011.13456.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency Models, May 2023. URL http://arxiv.org/abs/2303.01469. arXiv:2303.01469.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport, March 2024. URL http://arxiv.org/abs/2302.00482. arXiv:2302.00482.

Tongzhou Wang and Phillip Isola. Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere, August 2022. URL http://arxiv.org/abs/2005.10242. arXiv:2005.10242.

Ling Yang, Zixiang Zhang, Zhilong Zhang, Xingchao Liu, Minkai Xu, Wentao Zhang, Chenlin Meng, Stefano Ermon, and Bin Cui. Consistency Flow Matching: Defining Straight Flows with Velocity Consistency, July 2024. URL http://arxiv.org/abs/2407.02398. arXiv:2407.02398 [cs.CV].

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T. Freeman, and Taesung Park. One-step Diffusion with Distribution Matching Distillation, October 2024. URL http://arxiv.org/abs/2311.18828. arXiv:2311.18828.

Beyond Random Couplings: Spherical Noise Alignment in Generative Flows

Supplementary Material

We provide the following supplementary sections:

- **Section A:** Proofs and theoretical derivations.
- **Section B:** Implementation details, algorithm pseudocode, and hyperparameter settings.
- **Section C:** Further empirical analyses and discussions.
- **Section D:** Comparison of few-step generation performance.
- **Section E:** More qualitative results and visual samples generated.

A PROOFS

A.1 PROOF OF EQUATION (8)

For completeness, we include the derivation of the standard KL divergence decomposition in spherical coordinates. Let $\mathbf{x} \in \mathbb{R}^d$ be a continuous random vector distributed according to distribution Q with density $p(\mathbf{x})$. In spherical coordinates, $\mathbf{x}$ is represented by its norm $r = \|\mathbf{x}\|_2 \in (0, \infty)$ and unit direction $\mathbf{u} = \mathbf{x}/\|\mathbf{x}\|_2 \in \mathbb{S}^{d-1}$.

Let $p(\mathbf{u}, r)$ denote the joint probability density function of $(\mathbf{u}, r) \sim Q_{\mathbf{u},r}$, and let $p_{\mathbf{u}}(\mathbf{u})$ and $p_r(r)$ denote the marginal densities of $Q_{\mathbf{u}}$ and Q_r, respectively.

In this coordinate transformation, the standard Gaussian target $\mathcal{N}(0, \mathbf{I}_d)$ transforms into a product distribution $\sigma \otimes \chi_d$, with joint density $q(\mathbf{u}, r) = p_\sigma(\mathbf{u})p_\chi(r)$. Here, $p_\chi(r)$ is the probability density of the d-dimensional Chi distribution χ_d, and $p_\sigma(\mathbf{u}) = 1/A(\mathbb{S}^{d-1})$ is the probability density of the uniform distribution $\sigma = \mathrm{Unif}(\mathbb{S}^{d-1})$, where $A(\mathbb{S}^{d-1})$ is the surface area of the unit sphere.

We start from the definition of the KL divergence in spherical coordinates:

$$D_{\mathrm{KL}}(Q \,\|\, \mathcal{N}(0, \mathbf{I}_d)) = \int_0^\infty \int_{\mathbb{S}^{d-1}} p(\mathbf{u}, r) \log \left(\frac{p(\mathbf{u}, r)}{p_\sigma(\mathbf{u})p_\chi(r)} \right) \mathrm{d}\mathbf{u}\, \mathrm{d}r \tag{15}$$

$$= \mathbb{E}_{(\mathbf{u},r)\sim Q_{\mathbf{u},r}} \left[\log \left(\frac{p(\mathbf{u}, r)}{p_\sigma(\mathbf{u})p_\chi(r)} \right) \right]. \tag{16}$$

Multiplying and dividing the argument inside the logarithm by the product of marginal densities $p_{\mathbf{u}}(\mathbf{u})p_r(r)$, we split the log-ratio into three terms:

$$\log \left(\frac{p(\mathbf{u}, r)}{p_\sigma(\mathbf{u})p_\chi(r)} \right) = \log \left(\frac{p_{\mathbf{u}}(\mathbf{u})p_r(r)}{p_\sigma(\mathbf{u})p_\chi(r)} \cdot \frac{p(\mathbf{u}, r)}{p_{\mathbf{u}}(\mathbf{u})p_r(r)} \right) \tag{17}$$

$$= \log \left(\frac{p_{\mathbf{u}}(\mathbf{u})}{p_\sigma(\mathbf{u})} \right) + \log \left(\frac{p_r(r)}{p_\chi(r)} \right) + \log \left(\frac{p(\mathbf{u}, r)}{p_{\mathbf{u}}(\mathbf{u})p_r(r)} \right). \tag{18}$$

Taking the expectation of each term under $(\mathbf{u}, r) \sim Q_{\mathbf{u},r}$ gives:

$$D_{\mathrm{KL}}(Q \,\|\, \mathcal{N}(0, \mathbf{I}_d)) = \mathbb{E}_{Q_{\mathbf{u},r}} \left[\log \frac{p_{\mathbf{u}}(\mathbf{u})}{p_\sigma(\mathbf{u})} \right] + \mathbb{E}_{Q_{\mathbf{u},r}} \left[\log \frac{p_r(r)}{p_\chi(r)} \right] + \mathbb{E}_{Q_{\mathbf{u},r}} \left[\log \frac{p(\mathbf{u}, r)}{p_{\mathbf{u}}(\mathbf{u})p_r(r)} \right]. \tag{19}$$

Marginalizing out r in the first term yields the angular divergence $D_{\mathrm{KL}}(Q_{\mathbf{u}} \,\|\, \sigma)$, and marginalizing out $\mathbf{u}$ in the second term yields the radial divergence $D_{\mathrm{KL}}(Q_r \,\|\, \chi_d)$. The third term is the mutual information $D_{\mathrm{KL}}(Q_{\mathbf{u},r} \,\|\, Q_{\mathbf{u}} \otimes Q_r)$ between direction and radius.

Substituting these terms yields Equation (8):

$$D_{\mathrm{KL}}(Q \,\|\, \mathcal{N}(0, \mathbf{I}_d)) = \underbrace{D_{\mathrm{KL}}(Q_{\mathbf{u}} \,\|\, \sigma)}_{\text{Angular mismatch}} + \underbrace{D_{\mathrm{KL}}(Q_r \,\|\, \chi_d)}_{\text{Radial mismatch}} + \underbrace{D_{\mathrm{KL}}(Q_{\mathbf{u},r} \,\|\, Q_{\mathbf{u}} \otimes Q_r)}_{\text{Radius–direction dependence}}, \tag{20}$$

which completes the proof.

B Implementation Details

This section details the model architectures, training hyperparameters, and evaluation metrics used in our experiments.

B.1 Evaluation Metrics

Fréchet Inception Distance (FID). FID evaluates sample quality by comparing the empirical statistics of real and generated images in the feature space of a pre-trained Inception-v3 network Heusel et al. (2018). To match the baselines from other papers, we use the standard `clean-fid` implementation in legacy TensorFlow mode. Reference statistics are calculated on all training images and compared with 50,000 samples generated by the model. FID evaluates both the quality of the generated images and their diversity.

Mean Path Curvature. We apply the trajectory curvature metric introduced by Lee et al. (2023). Let $v_t(x_t)$ denote the predicted network velocity along the path from initial noise x_0 to the generated data x_1. The mean path curvature κ is defined as:

$$\kappa = \mathbb{E}_{t,x_0}\left[\|(x_1 - x_0) - v_t(x_t)\|_2^2\right] \tag{21}$$

where $t \sim U(0,1)$ and $x_0 \sim \rho_0 = \mathcal{N}(0, \mathbf{I})$. In our evaluation, we estimate this expectation by averaging the path curvature over $50,000$ generated samples. Curvature is not comparable across solvers and timestep schedules. However, within a fixed setup, lower values indicate straighter generative paths. Generally, straighter paths are preferred, as they allow sampling with far fewer required sampling steps to reach the same quality.

B.2 Pseudocode

The detailed algorithm for the training loop using our proposed coupling method is provided in Algorithm 1. Compared to the standard training code, we introduce a lightweight inner loop that directly optimizes the noise vectors without backpropagating through the neural network. The inference code remains exactly the same and does not require modification. Note that we scale the learning rate as $\eta/\sqrt{d}$ to match the $O(\sqrt{d})$ norm scale of high-dimensional Gaussian vectors.

Algorithm 1 Training Flow Matching with Spherical Noise Alignment (SNA)

Inputs: Flow model v_θ, batch size N, dimension d, initial coupling $\pi_{\text{init}} \in \Pi(\rho_0, \rho_1)$, SNA learning rate η, SNA optimization steps K, temperatures τ, γ, loss weight λ_S
Outputs: Trained network parameters θ

while network θ has not converged **do**
  Sample pairs $\{(x_{0,i}, x_{1,i})\}_{i=1}^N \sim \pi_{\text{init}}$
  # Spherical Noise Alignment (SNA)
  $\hat{\mathbf{x}}_1 \leftarrow \{x_{1,i}/\|x_{1,i}\|_2\}_{i=1}^N$
  $\hat{\mathbf{x}}_0 \leftarrow \{x_{0,i}/\|x_{0,i}\|_2\}_{i=1}^N$ {▷ *Initialize normalized noise batch*}
  for $t = 1$ **to** K **do**
    $\mathcal{D}_{S,\tau} := -\frac{1}{N}\sum_{i=1}^N \log\left(\frac{\exp(\langle \hat{x}_{0,i}, \hat{x}_{1,i}\rangle/\tau)}{\sum_{j=1}^N \exp(\langle \hat{x}_{0,i}, \hat{x}_{1,j}\rangle/\tau)}\right)$
    $\mathcal{D}_{G,\gamma} := \frac{1}{N}\sum_{i=1}^N \log \sum_{j\neq i} \exp\left(\frac{\langle \hat{x}_{0,i}, \hat{x}_{0,j}\rangle}{\gamma}\right)$
    $\mathcal{L}_{\text{SNA}} := \lambda_S \mathcal{D}_{S,\tau} + \mathcal{D}_{G,\gamma}$
    $\hat{\mathbf{x}}_0 \leftarrow \hat{\mathbf{x}}_0 - \eta \frac{1}{\sqrt{d}} \cdot \nabla_{\hat{\mathbf{x}}_0} \mathcal{L}_{\text{SNA}}$ {▷ *Gradient descent step*}
    $\hat{\mathbf{x}}_0 \leftarrow \{\hat{x}_{0,i}/\|\hat{x}_{0,i}\|_2\}_{i=1}^N$ {▷ *Project back to unit hypersphere*}
  end for
  Sample new norms $r_i \sim \chi_d$ for $i = 1 \ldots N$ {▷ *Resample radial distribution*}
  Let $\mathbf{x}_0^* = \{r_i \cdot \hat{x}_{0,i}\}_{i=1}^N$ be the optimized coupled noise batch
  $\theta \leftarrow \text{FMStep}(\theta, (x_{0,i}^*)_{i=1}^N, (x_{1,i})_{i=1}^N)$ {▷ *Sample t, compute vector field loss, update* θ}
end while
return θ

B.3 Flow Model Training

Pixel Space Experiments. For pixel-space models, we evaluate on CIFAR-10 and ImageNet 32×32. We adopt the exact U-Net (ADM) architectures and hyperparameter setups from Tong et al. (2024) for CIFAR-10 and Pooladian et al. (2023) for ImageNet 32×32, with full parameters summarized in Table 5. During training, time steps are sampled uniformly as $t \sim \mathcal{U}(0, 1)$. All models are optimized using Adam ($\beta_1 = 0.9, \beta_2 = 0.999$) without weight decay, and we also apply an Exponential Moving Average (EMA) with a decay factor of 0.9999 to the network weights. We adopt gradient clipping with a maximum norm of 1.0.

The CIFAR-10 models are trained on a single NVIDIA RTX 4090 GPU, while ImageNet 32×32 models are trained on a node with $8\times$ NVIDIA RTX 4090 GPUs. In multi-GPUs settings, the inner loop of our coupling method is synchronized across all GPUs.

Table 5: ADM network architecture and hyperparameters used for pixel space flow model training.

	CIFAR-10	**ImageNet (32x32)**
Channels	128	256
Depth	2	3
Channels multipliers	$1, 2, 2, 2$	$1, 2, 2, 2$
Heads	4	4
Heads channels	64	64
Attention resolution	16	4
Dropout	0.1	0.0
Batch size / GPU	128	128
Effective batch size	128	1024
GPUs	1	8
Iterations	390k	400k
Learning rate	0.0002	0.0001
Learning rate scheduler	Constant	Polynomial Decay
Warmup steps	5k	20k
Model parameters	36M	189M

Latent Space Experiments. For high-resolution generation, following common practice, input images are first compressed into lower-dimensional $4 \times 32 \times 32$ latent representations using the frozen Stable Diffusion Variational Autoencoder (`sd-vae-ft-ema`). The flow network and our SNA objective then operate within this latent space.

For these tasks, we replace the U-Net with the Scalable Interpolant Transformer (SiT) architecture (Ma et al., 2024). The specific architectural configurations and training hyperparameters are detailed in Table 6. For these experiments, we also use the Adam optimizer ($\beta_1 = 0.9, \beta_2 = 0.999$) without weight decay, a fixed learning rate of $1e-4$, and EMA weights with a decay factor of 0.9999. We use the SiT-S/2 configuration trained on a single NVIDIA RTX 4090 GPU for FFHQ, while for ImageNet we scale up to the larger SiT-B/2 architecture, distributing training across $4\times$ NVIDIA RTX 4090 GPUs on a single node. The global batch size is set to 256.

For class-conditional generation, we apply minibatch OT and SNA strictly intra-class following Cheng & Schwing (2025). To ensure sufficient class coverage, we perform the OT and SNA pairing steps on a virtual batch size equal to $C \times N$ (where C is the number of classes). This decouples the batch size of OT from the batch size of the update of the neural network, maintaining an effective batch size per-class of roughly N for the coupling step. This step is necessary to ensure that the prior distribution conditioned on the class does not drift from the Gaussian prior. During sampling, we apply classifier-free guidance with a scale of 2.0.

Table 6: SiT network architecture and hyperparameters used for latent space flow model training.

	FFHQ (256x256)	**ImageNet (256x256)**
Architecture	SiT-S/2	SiT-B/2
Patch size	2×2	2×2
Hidden size	384	768
Layers (Depth)	12	12
Attention heads	6	12
Head dimension	64	64
Latent resolution	32×32	32×32
Dropout	0.0	0.0
Batch size / GPU	256	64
Effective batch size	256	256
GPUs	1	4
Iterations	500k	1M
Learning rate	1×10^{-4}	1×10^{-4}
Learning rate scheduler	Constant	Constant
Warmup steps	0	0
Model parameters	33M	130M

B.4 Hyperparameter Settings

For the inner optimization loop, we use $K = 4$ steps with the Adam optimizer ($\beta_1 = 0.9, \beta_2 = 0.999$) and a learning rate of $\eta = 0.02$. The loss converges quickly and as few as 4 steps are usually sufficient. The temperature parameters controlling the sharpness of the cross-modal alignment and the unimodal generative repulsion are fixed at $\tau = 0.01$ and $\gamma = 0.01$, respectively. Generally, it is recommended to use a similar scale for τ and γ. These settings are applied across all experiments and provide a good starting point for further adjustments.

The primary parameter that requires tuning is the alignment weight (λ_S). For standard SNA, the configuration $\lambda_S \in [1/5, 1]$ performs consistently well. Interestingly, these same ranges were transferred directly to our high-resolution latent models on FFHQ256 and ImageNet256. This shows that the hyperparameters remain robust.

C Further Analyses And Discussion

C.1 Noise-Target Directional Alignment

To empirically verify that Contrastive Noise Alignment systematically enhances cross-modal pairing structure, we evaluate the directional coupling between matched noise-target pairs (x_0^*, x_1). Specifically, we measure the expected batch cosine similarity between paired samples. As shown in Table 7, standard Independent Coupling (I-FM) has zero correlation in expectation with its assigned targets. While minibatch OT (OT-FM) establishes a pre-structured global coupling (0.0379 on CIFAR-10 and 0.0491 on ImageNet-32), applying SNA significantly increases directional alignment.

Table 7: **Noise-Target Alignment.** Mean cosine similarity (↑) between matched noise and target samples (x_0^*, x_1). Relative gains shown over respective baselines (I-FM and OT-FM).

Dataset	I-FM	SNA ($\lambda_S = 1/2$)	OT-FM	SNA + OT ($\lambda_S = 1/5$)
CIFAR-10	0.0000	0.0453	0.0379	**0.0597** *(+57.5%)*
ImageNet32	0.0000	0.0561	0.0491	**0.0707** *(+44.0%)*

When initializing from OT and optimizing under SNA ($\lambda_S = 1/5$), the average cosine similarity improves to 0.0597 in CIFAR-10 and 0.0707 in ImageNet32 (an increase of more than $1.5\times$ over OT-FM). This confirms that the alignment term $\mathcal{D}_S$ effectively draws the paired noise representations towards their corresponding targets, substantially reducing the directional displacement.

C.2 Computational Efficiency

We analyze the scalability of our method by training throughput as shown in Table 8. SNA introduces minimal overhead relative to independent couplings. Because inner optimization converges in just a few steps, the actual time penalty is practically negligible, costing less than a $\sim 1.5\%$ drop in throughput. Even at larger batch sizes ($N = 1024$) where the exact OT solver becomes a noticeable bottleneck, our contrastive updates scale effectively.

Table 8: **Training throughput comparison** (iterations/s ↑, measured on NVIDIA RTX 4090).

Dataset	I-FM	SNA (ours)	OT-FM	SNA + OT (ours)
CIFAR-10 ($N = 128$)	6.68	6.58 *(-1.5%)*	6.65	6.56 *(-1.4%)*
ImageNet32 ($N = 1024$)	1.77	1.75 *(-1.1%)*	1.31	1.30 *(-0.8%)*

C.3 Post-Hoc Path Regulation via λ_S-Conditioning

We train an adaptive variant of our model in which the alignment weight λ_S is sampled uniformly from the interval $[1, 1/6]$ during training. To handle this continuous conditioning, we compute a sinusoidal embedding of λ_S passed through a 2-layer MLP $\psi(\beta)$, which is directly added to the standard time embedding $\phi(t)$. This explicitly conditions the learned vector field $v_\theta(x_t, t, \lambda_S)$ on the regularization parameter. Intuitively, the variation of λ_S parametrizes a family of transport trajectories between the prior and the data. This allows the model to learn multiple routing strategies.

At inference time, we sample directly from a standard Gaussian prior $x_0 \sim \mathcal{N}(0, \mathbf{I})$ and supply a fixed λ_S to guide the generation path. Table 4 presents the CIFAR-10 results. Tuning λ_S at inference shifts the performance profile between different NFE budgets, allowing users to balance generation quality and path alignment post-hoc without retraining the model.

C.4 VISUALIZING THE GAUSSIANITY LOSS

In Figure 6, we wanted to see if our Gaussianity loss ($\mathcal{D}_{\text{G}}$) could take structured image latents and successfully force them back into standard Gaussian noise. We tested this at two resolutions—32×32 and 64×64—using a fixed batch size of $N = 4096$. Looking at the upper half of the figure, the 32×32 latents actually optimize fairly well. The fact that it generates coherent images tells us that the latent space is picking up the right Gaussian properties. However, the 64×64 latents in the bottom block are more difficult. The increase in dimensionality requires larger batches to properly cover the distribution space.

This shows that while it might be possible to transform structured image latents into noise using our Gaussian regularizer, simply initializing with random Gaussian noise is practically much easier.

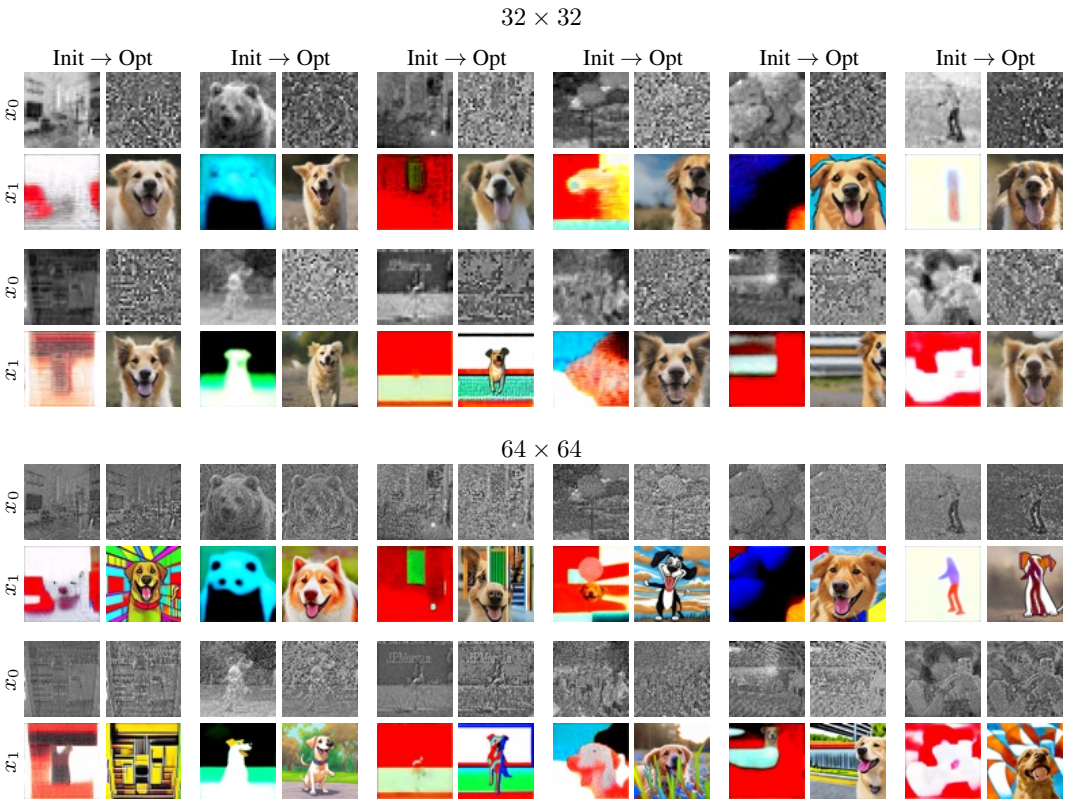


Figure 6: **Testing Gaussianity Loss.** We evaluate the capacity of our Gaussianity term $\mathcal{D}_{\text{G}}$ to convert structural image latents (*Init*) into Gaussian noise (*Opt*) using a batch size of $N = 4096$. In each section, the top rows show the latent space, and the bottom rows show the resulting SDXL-Turbo generations. Note that for the 32×32 resolution, some artifacts stem from using the non-native resolution of SDXL-Turbo; nevertheless, the model's ability to generalize suggests Gaussian properties. Higher-dimensional spaces (bottom) require much larger batches to achieve good results.

D FEW-STEP SAMPLING QUALITY

To illustrate how SNA performs in the few-step regime, Figure 7, Figure 8 and Figure 9 compare qualitative results across different sampling budgets. As seen in the figures, SNA consistently produces higher quality samples than the baselines, especially when constrained to just 2 or 4 NFEs.



Figure 7: **Qualitative few-step sampling results on FFHQ256.**

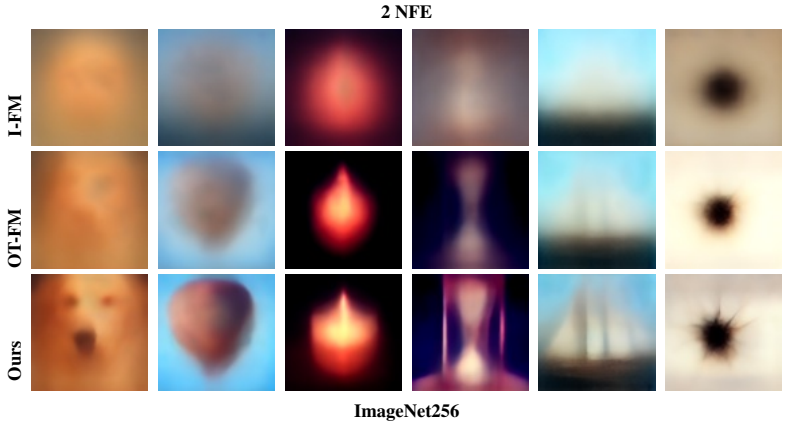
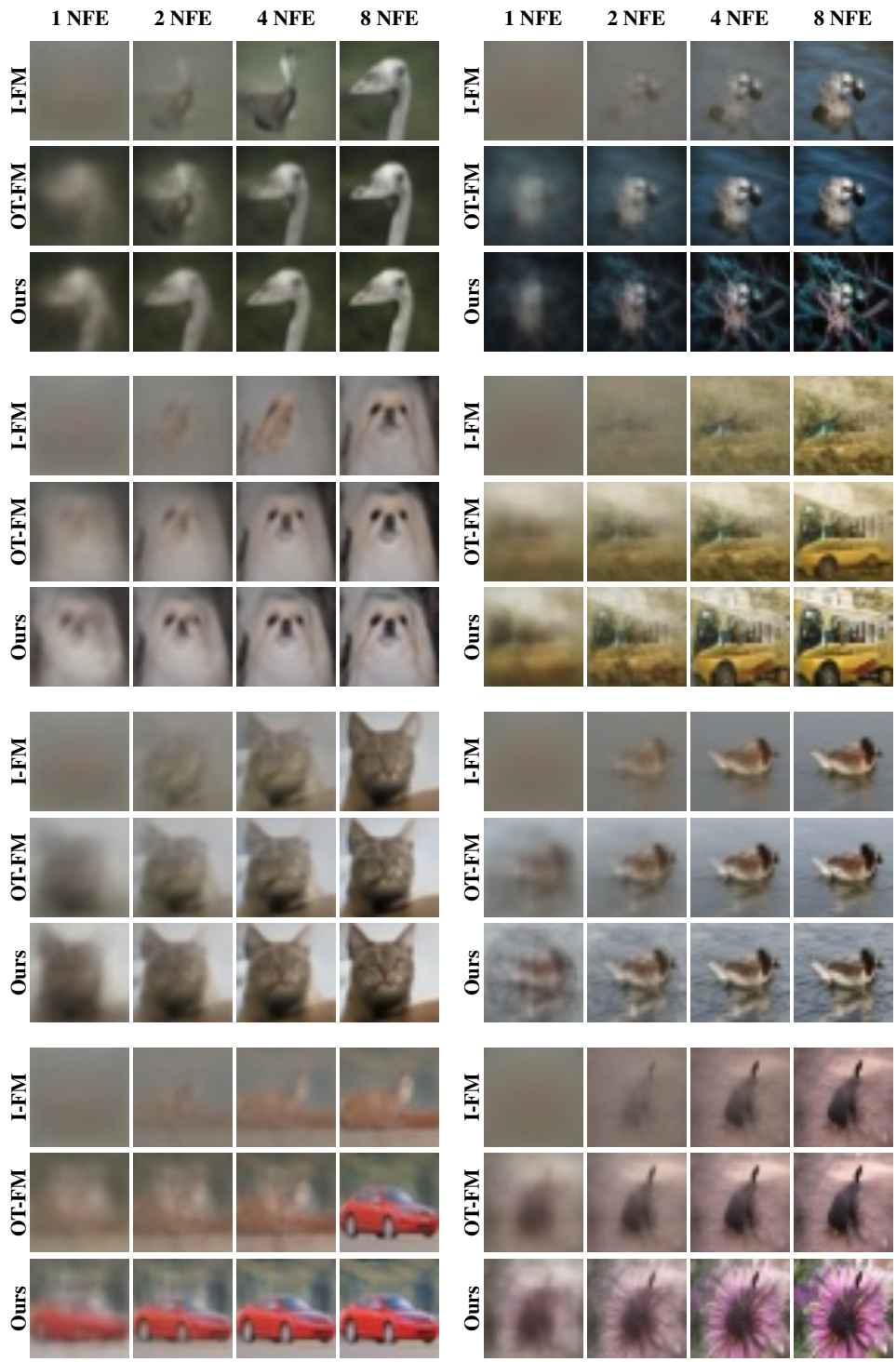


Figure 8: **Qualitative few-step sampling results on ImageNet256.**



(a) CIFAR-10 (b) ImageNet32

Figure 9: **Qualitative comparison across different step counts.** Comparison of sampling trajectories for CIFAR-10 (left) and ImageNet32 (right). At 1 and 2 NFEs, our method maintains significantly better global structure than the baseline variants.

E MORE QUALITATIVE RESULTS

We provide additional qualitative examples generated using SNA. All samples are produced using the Euler solver at 64 sampling steps. The results are presented in Figure 10 for CIFAR-10, Figure 11 for ImageNet-32, and Figure 12 for FFHQ-256, while the class-conditional samples for ImageNet-256 are shown in Figure 13.



Figure 10: **Uncurated 32×32 samples on CIFAR-10** ($\lambda_S = 1/5$).



Figure 11: **Uncurated 32×32 samples on ImageNet32** ($\lambda_S = 1/30$)



Figure 12: **Uncurated 256×256 samples on FFHQ256** ($\lambda_S = 1/5$)



Class label = “coral reef” (973)



Class label = “volcano” (980)



Class label = “bagel” (931)



Class label = “arctic fox” (279)



Class label = “golden retriever” (207)



Class label = “geyser” (974)

Figure 13: **Uncurated class-conditional samples on ImageNet256** ($\lambda_S = 1/5$, CFG = 4.0)