TIMESTEP WEIGHTING THROUGH THE LENS OF FLOW MATCHING DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Flow matching learns generative models by predicting velocity fields along a predefined path between a prior and a data distribution. The design of the training objective, in particular the weighting across timesteps, is known to have a strong impact on performance, yet further insight into the underlying generation dynamics is needed to guide its design. In this work, we study marginal flow dynamics through the propagation of velocity prediction errors to the generated endpoint. For Gaussian mixture targets, we derive a decomposition of the velocity Jacobian into a within-component term and a term governed by uncertainty over component assignments, and use this structure to approximate timestep-dependent error amplification. We complement the analysis with perturbation probes that measure endpoint sensitivity along model prediction-error directions. Our analysis and observations suggest a transition from an earlier regime in which component ambiguity contributes to error amplification to a later regime in which this contribution diminishes and within-component dynamics dominate. This transition motivates allocating training emphasis asymmetrically across the generation process. Building on logit-normal weighting, we introduce a parameterized refinement that reduces the relative emphasis on the later regime while retaining endpoint suppression. Experiments demonstrate improved generation quality over the logit-normal baseline across multiple image datasets and all tested model sizes.

1 INTRODUCTION

Diffusion and flow-matching models generate samples by progressively transforming a simple prior distribution into a complex data distribution (Ho et al., 2020; Song et al., 2020; Lipman et al., 2022; Liu et al., 2022). Among these approaches, flow matching provides a simulation-free training framework that learns a velocity field associated with a prescribed probability path. Its simplicity and flexibility have enabled strong performance in image generation (Esser et al., 2024; Labs et al., 2025).

Standard flow matching trains the velocity field using a squared-error objective under uniform timestep sampling and constant loss weighting (Lipman et al., 2022). However, timesteps play different roles: earlier stages establish global structure and select among data modes, while later stages refine local details (Liu et al., 2026; Ren et al., 2026). This suggests allocating training emphasis unevenly across timesteps may improve generation quality. Time-dependent weighting is theoretically valid for flow matching (Billera et al., 2025), and existing work realizes it through nonuniform timestep sampling or loss weighting (Esser et al., 2024; Galinska et al., 2026). In particular, Esser et al. (2024) emphasize intermediate timesteps, motivated by the observation that the optimal velocity simplifies near the endpoints. Among the candidate schemes they compare, logit-normal weighting (Atchison & Shen, 1980) performs best and has been widely adopted (Li & He, 2026; Xie et al., 2024).

Despite its wide adoption, logit-normal weighting was selected by empirical comparison under the broad rationale of emphasizing intermediate timesteps. This leaves open how emphasis should be allocated within this range, and how a finer understanding of the flow dynamics could inform the weighting design. In this work, we study marginal flow dynamics through endpoint sensitivity to velocity prediction errors. Under a Gaussian mixture target, the reference velocity Jacobian decomposes exactly into a within-component term and a term governed by posterior uncertainty over component assignments, and we introduce a scalar approximation of the resulting timestep-dependent endpoint sensitivity. We complement this analysis with probes along the prediction-error directions

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.

of a trained flow matching model on image latent codes. In the settings we study, the marginal flow exhibits an early regime with substantial error amplification from unresolved component assignments, followed by a regime where this contribution vanishes and within-component dynamics dominate, consistent with the navigation and refinement stages observed in (Liu et al., 2026). Moreover, the velocity error itself remains small in the early regime, so the squared velocity loss alone does not reflect its downstream impact.

Taking advantage of the insight from marginal flow dynamics, the greater error amplification observed in the earlier regime motivates assigning it greater relative training emphasis. Building on logit-normal weighting, we introduce a parameterized refinement that attenuates the later regime while retaining the baseline's endpoint suppression. The sensitivity analysis informs the region in which attenuation should begin, and empirical evaluation determines its strength.

Our contributions are three-fold:

- We analyze the dynamics of the reference marginal flow through a GMM-based Jacobian decomposition, identifying two factors that govern timestep-dependent endpoint sensitivity: within-component dynamics and component ambiguity.
- We conduct perturbation probe analysis along model prediction-error directions to examine the effect of component ambiguity on endpoint-error amplification, revealing a transition from an earlier ambiguity-dominated regime to a later within-component-dominated regime.
- Guided by this transition, we propose an asymmetric refinement of existing timestep weightings whose attenuation onset is derived from the component-ambiguity analysis. Evaluations across different model sizes and datasets show consistent performance improvements.

2 Background

2.1 Flow Matching Objectives

Let $p_0 = \mathcal{N}(0, I)$ be the prior and $p_1 = p_{\text{data}}$ the target distribution on $\mathbb{R}^d$, where d is the data dimension and I is the identity matrix. Let $u_t(x)$ be a reference velocity field generating a probability path $\{p_t\}_{t\in[0,1]}$ between these distributions. Consider conditional probability paths $p_t(x|x_1)$ generated by conditional velocity fields $u_t(x|x_1)$, where $x_1 \sim p_{\text{data}}$. Their marginal counterparts are:

$$p_t(x) = \int p_t(x \mid x_1)\, p_{\text{data}}(x_1)\, dx_1, \qquad u_t(x) = \mathbb{E}_{x_1\sim p(x_1|x_t=x)}\big[u_t(x \mid x_1)\big]. \tag{1}$$

Here x_t denotes the intermediate random state, and $p(x_1 \mid x_t = x)$ denotes the posterior distribution of the target sample given that state. Flow matching (Lipman et al., 2022) learns a velocity field $v_\theta(x, t)$ with parameters θ to approximate the reference marginal flow field $u_t(x)$. Its objective is

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}(0,1),\, x\sim p_t}\left[\|v_\theta(x,t) - u_t(x)\|_2^2\right], \tag{2}$$

where $\mathcal{U}(0, 1)$ denotes uniform timestep sampling.

Throughout this work, we use independent noise-data pairs and straight conditional trajectories,

$$x_t = (1-t)x_0 + tx_1, \qquad x_0 \sim p_0, \quad x_1 \sim p_{\text{data}}, \quad x_0 \perp x_1. \tag{3}$$

Here $x_0 \perp x_1$ denotes independence; $t = 0$ corresponds to noise and $t = 1$ to data. For $0 \le t < 1$, this construction gives

$$u_t(x \mid x_1) = \frac{x_1 - x}{1-t}. \tag{4}$$

Along each conditional trajectory, this reduces to the target velocity $x_1 - x_0$. We therefore write the weighted conditional flow-matching objective as

$$\mathcal{L}_w(\theta) = \mathbb{E}_{t\sim\mathcal{U}(0,1),\, x_0\sim p_0,\, x_1\sim p_{\text{data}}}\left[W(t)\,\|v_\theta((1-t)x_0 + tx_1, t) - (x_1 - x_0)\|_2^2\right]. \tag{5}$$

Here $W(t)$ controls the relative training emphasis across timesteps, with $W(t) = 1$ recovering the standard objective. Throughout this work, we normalize $W(t)$ to have unit mean over $t \in [0, 1]$.

2.2 Temporal Behavior of the Marginal Flow Field

Under the independent linear interpolation, the reference marginal velocity field is $u_t(x) = \mathbb{E}[x_1 - x_0 \mid x_t = x]$, and its state dependence simplifies at the endpoints. At $t = 0$, the state is the noise x_0, independent of the target; at $t = 1$, it is the target x_1, independent of the zero-mean noise. Hence $u_0(x) = \mu_{\text{data}} - x$ and $u_1(x) = x$, where $\mu_{\text{data}} = \mathbb{E}[x_1]$ is the data mean, so the velocity is affine in the state at both endpoints. For smooth target densities under suitable moment and regularity conditions, the marginal field approaches these affine forms near the endpoints.

At intermediate timesteps, the state contains contributions from both noise and data. Writing the marginal velocity in terms of the posterior mean target gives

$$u_t(x) = \frac{m_t(x) - x}{1 - t}, \qquad m_t(x) = \mathbb{E}[x_1 \mid x_t = x], \quad 0 < t < 1, \tag{6}$$

where $m_t(x)$ averages target samples according to their posterior probabilities given the current state. For a multimodal data distribution, different states favor different targets or mixtures of targets, making $m_t(x)$ nonlinear in x and the intermediate field more complex than the affine endpoint fields. This provides an intuition for emphasizing intermediate timesteps during training (Esser et al., 2024).

3 Endpoint Sensitivity Analysis

In this work, we start with the question: ***Which timesteps have greater influence on endpoint placement?*** We answer it by measuring how local state perturbations at each timestep propagate to the endpoint, via the Jacobian of the forward flow map. Under a Gaussian mixture target, this sensitivity decomposes into within-component and component-ambiguity factors, whose temporal behavior on image latents reveals a characteristic transition. Derivations are provided in Appendix A.

3.1 Endpoint Sensitivity Formulation

We first quantify how a local perturbation at time t affects the generated endpoint. Let $\mathcal{F}_{r\leftarrow t}$ denote the forward flow map induced by the reference marginal velocity field, mapping a state at time t to its position at time r, where $0 \le t \le r \le 1$. It is defined by $\frac{\partial}{\partial r}\mathcal{F}_{r\leftarrow t}(x) = u_r(\mathcal{F}_{r\leftarrow t}(x))\,, \qquad \mathcal{F}_{t\leftarrow t}(x) = x$, where x is the state at time t. In particular, $\mathcal{F}_{1\leftarrow t}(x)$ gives the endpoint obtained by evolving x from time t under the reference marginal flow. A local error $\epsilon = \hat{x} - x$, where $\hat{x}$ is the perturbed state, induces a state error $e(r) = \mathcal{F}_{r\leftarrow t}(\hat{x}) - \mathcal{F}_{r\leftarrow t}(x)$ at time $r \in [t, 1]$. We define the sensitivity matrix as $\mathbf{S}(x, r, t) = \left.\frac{\partial e(r)}{\partial \epsilon}\right|_{\epsilon=0}$; setting $r = 1$ gives the endpoint sensitivity. To first order, $e(1) \approx \mathbf{S}\epsilon$ with $\mathbf{S} = \mathbf{S}(x, 1, t)$, so the squared error amplification is $\epsilon^\top \mathbf{S}^\top \mathbf{S}\,\epsilon/\epsilon^\top\epsilon$. Averaging over states and perturbation directions gives

$$s(t) = \frac{1}{d}\mathbb{E}_{x\sim p_t}\left[\operatorname{tr}\big(\mathbf{S}(x, 1, t)^\top \mathbf{S}(x, 1, t)\big)\right], \tag{7}$$

where d is the state dimension. A larger $s(t)$ indicates that errors introduced at timestep t have greater average influence on the final sample.

3.2 Endpoint Sensitivity under a Gaussian Mixture Model

To analyze sensitivity in closed form, we model the target as a Gaussian mixture model (GMM), which preserves multimodality while keeping the marginal velocity and its Jacobian tractable. Let $x_r = \mathcal{F}_{r\leftarrow t}(x)$ be the reference trajectory starting from x at time t, and let $J_u(x, r) = \partial u_r(x)/\partial x$ denote the velocity Jacobian. The sensitivity matrix satisfies

$$\frac{\partial \mathbf{S}(x, r, t)}{\partial r} = J_u(x_r, r)\mathbf{S}(x, r, t), \qquad \mathbf{S}(x, t, t) = I. \tag{8}$$

Thus, the endpoint sensitivity is determined by the Jacobian accumulated along the trajectory from t to 1. We approximate the target distribution by K Gaussian components with a shared isotropic covariance, $p_1(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x; \mu_k, \tau^2 I)$, where π_k is the mixture weight, μ_k is the component mean, and $\tau^2 > 0$ is the shared variance per coordinate.

Under the independent linear interpolation introduced in Section 2.1, the state distribution conditioned on component k is Gaussian with mean $t\mu_k$ and covariance $\rho_t^2 I$, where $\rho_t^2 = (1-t)^2 + t^2\tau^2$. Given a state x at time t, the posterior probability of component k is $\gamma_k(x,t) = \pi_k \mathcal{N}(x; t\mu_k, \rho_t^2 I) / \sum_{j=1}^K \pi_j \mathcal{N}(x; t\mu_j, \rho_t^2 I)$. The reference marginal velocity then takes the form

$$u_t(x) = a(t)x + b(t)\bar{\mu}, \quad \bar{\mu} = \sum_{k=1}^K \gamma_k \mu_k, \quad a(t) = \frac{t\tau^2 + t - 1}{\rho_t^2}, \quad b(t) = \frac{1-t}{\rho_t^2}. \tag{9}$$

Here, $\bar{\mu}(x,t)$ is the posterior mean of the component centers. Differentiating the velocity with respect to the current state gives the exact decomposition

$$J_u(x,t) = a(t)I + \beta(t)M(x,t), \qquad \beta(t) = \frac{t(1-t)}{\rho_t^4}, \tag{10}$$

where $M(x,t) = \sum_{k=1}^K \gamma_k(x,t)\big(\mu_k - \bar{\mu}(x,t)\big)\big(\mu_k - \bar{\mu}(x,t)\big)^\top$ is the posterior covariance of the component centers. Thus, the velocity Jacobian separates into a within-component term and a component-ambiguity term. The term $a(t)I$ describes the velocity variation within each component. The additional term $\beta(t)M(x,t)$ arises because a change in position changes the posterior component probabilities.

Substituting Equation 10 into Equation 8 yields an exact decomposition of the endpoint sensitivity under the GMM model,

$$\mathbf{S}(x,1,t) = \underbrace{\frac{\tau}{\sqrt{(1-t)^2 + t^2\tau^2}} I}_{\text{Within-component factor}} \underbrace{\mathcal{T} \exp\left(\int_t^1 \beta(r) M(x_r, r)\, dr \right)}_{\text{Component-ambiguity factor}}, \tag{11}$$

where $\mathcal{T}$ denotes time ordering. The within-component factor is isotropic and coincides with the sensitivity of a single Gaussian target $\mathcal{N}(\mu, \tau^2 I)$. The component-ambiguity factor accumulates changes in posterior component probabilities along the trajectory through M, which vanishes as the posterior concentrates on one component.

3.3 Scalar Approximation of Endpoint Sensitivity

To examine how the two factors evolve across timesteps, we introduce an isotropic, state-averaged approximation $J_u(x,t) \approx g(t)I$ with $g(t) = \frac{1}{d}\mathbb{E}_{x\sim p_t}[\mathrm{tr}\, J_u(x,t)] = a(t) + \beta(t)\bar{q}(t)$, where $\bar{q}(t) = \frac{1}{d}\mathbb{E}_{x\sim p_t}[\mathrm{tr}\, M(x,t)]$ measures the average posterior spread of component centers per dimension. The sensitivity matrix then becomes $\mathbf{S}(x,1,t) \approx A(t)I$ with $A(t) = \exp(\int_t^1 g(r)\, dr)$, and substituting into Equation 7 gives $s(t) \approx A(t)^2 \equiv \tilde{s}(t)$. We obtain the scalar squared endpoint sensitivity

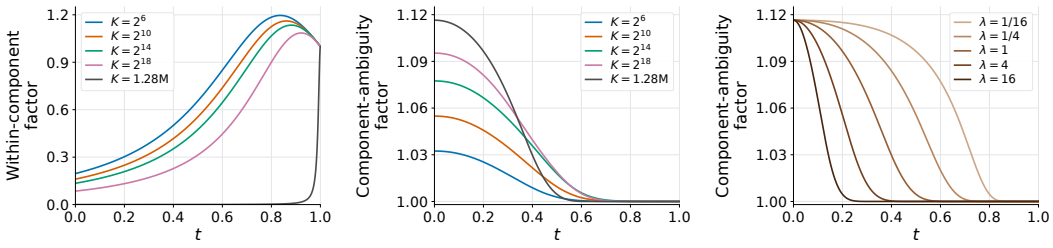
$$\tilde{s}(t) = \underbrace{\frac{\tau^2}{(1-t)^2 + \tau^2 t^2}}_{\text{Within-component factor}} \underbrace{\exp\left(2\int_t^1 \beta(r)\bar{q}(r)\, dr \right)}_{\text{Component-ambiguity factor}}. \tag{12}$$

We next evaluate this approximation on image latent distributions.

3.4 Numerical Study of Temporal Behavior

To evaluate the scalar approximation on image latent distributions, we construct GMMs with 2^6, 2^{10}, 2^{14}, 2^{18}, and 1.28M components from centered ImageNet latents (Deng et al., 2009). For K <1.28M, we apply K-means to obtain the component means, set the mixture weights according to cluster sizes, and estimate the shared variance from the within-cluster residuals. For the 1.28M-component GMM, each latent code serves as a component mean, and we set $\tau^2 = 10^{-4}$, close to the mean posterior variance of the VAE encoder (Kingma & Welling, 2022). The resulting are shown in Figure 1-(a,b).

When $\tau^2 < 1$, the within-component factor increases toward the later stage of the forward process and becomes increasingly concentrated near the data endpoint as the component variance decreases. In contrast, the component-ambiguity factor contributes mainly in the earlier stage and becomes negligible after a characteristic transition in time.



(a) Within-component factor (b) Component-ambiguity factor (c) Effect of covariance scaling

Figure 1: Temporal behavior of the two factors in the scalar sensitivity approximation $\tilde{s}(t)$.

We further examine how this transition depends on the covariance of the component centers. As shown in Figure 1-(c), we rescale the centered 1.28M-component GMM so that its target covariance is multiplied by $\lambda \in \{1/16, 1/4, 1, 4, 16\}$. Increasing the covariance scale shifts the decay of the component-ambiguity factor to earlier timesteps, while decreasing the covariance shifts it later. These results suggest that the temporal transition of component ambiguity depends on the spread of the target distribution.

4 Probing Endpoint Sensitivity along Prediction Errors

The sensitivity analysis in the Section 3 averages uniformly over perturbation directions and characterizes error amplification in the infinitesimal-perturbation limit. In practice, prediction errors from a trained flow-matching model have structured directions and finite magnitudes, and their endpoint effects may differ from this local average. We therefore construct a probe that measures endpoint-error amplification along perturbations induced by actual velocity prediction errors.

4.1 Probe Formulation

Applying the squared amplification ratio from Section 3 to symmetric finite perturbations, we define

$$s_{\text{probe}}(t) = \mathbb{E}_{x\sim p_t}\left[\frac{\|\mathcal{F}_{1\leftarrow t}(x+\epsilon) - \mathcal{F}_{1\leftarrow t}(x-\epsilon)\|_2^2}{4\|\epsilon\|_2^2}\right], \tag{13}$$

where ϵ is a local state perturbation induced by velocity prediction error. Unlike the scalar sensitivity in Equation 7, this probe evaluates finite perturbations along model-induced error directions.

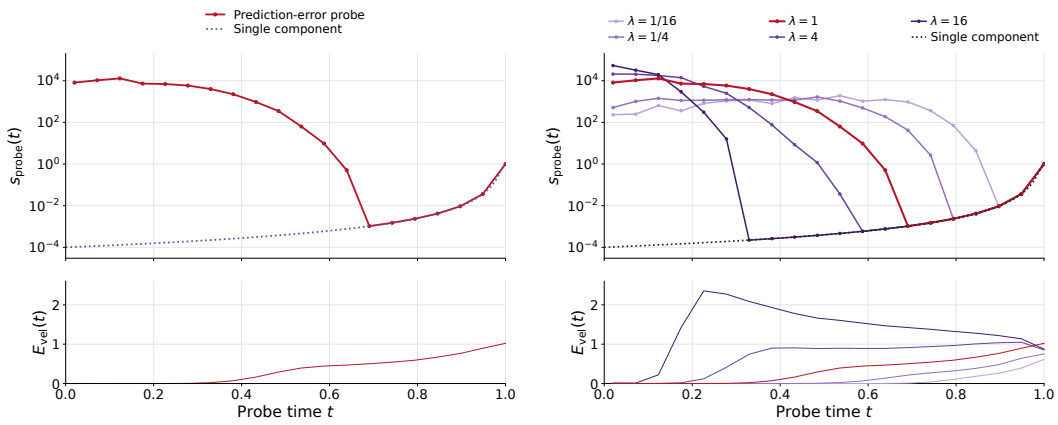
To first order, s_{probe} reduces to the directional amplification $\epsilon^\top \mathbf{S}^\top \mathbf{S}\,\epsilon/\epsilon^\top\epsilon$ averaged over states. To construct ϵ at time $t \geq \Delta t$, we sample $x^- \sim p_{t-\Delta t}$ and compare the reference state x with the one-step Euler state $\hat{x}$ under the learned velocity:

$$x = \mathcal{F}_{t\leftarrow t-\Delta t}(x^-), \qquad \hat{x} = x^- + \Delta t\, v_\theta(x^-, t-\Delta t), \qquad \epsilon = \hat{x} - x. \tag{14}$$

4.2 Probe Implementation and Analysis

We set $\Delta t = 1/50$, matching the commonly used 50-step Euler sampler, and estimate Equation 13 with 2048 Monte Carlo samples per timestep. The flow maps $\mathcal{F}_{r\leftarrow t}$ are integrated with an RK4 solver using the marginal velocity in Equation 9. For comparison, we also measure the velocity prediction error optimized during training, $E_{\text{vel}}(t) = \frac{1}{d}\mathbb{E}_{x^-\sim p_{t-\Delta t}}\|v_\theta(x^-, t-\Delta t) - u_{t-\Delta t}(x^-)\|_2^2$.

As shown in Figure 2-(a), the measured endpoint-error amplification substantially exceeds the within-component factor during the earlier portion of generation, followed by a sharp decrease toward the within-component response. Although the amplification magnitude differs from the isotropic scalar approximation, its temporal behavior is consistent with the decomposition in Section 3.2: additional amplification appears in the earlier interval and rapidly diminishes as the contribution of component ambiguity decreases. Meanwhile, the velocity prediction error remains relatively small during the earlier interval despite the substantially larger endpoint-error amplification. This indicates that local



(a) Original latent scale

(b) Comparison across latent scales

Figure 2: Endpoint-error amplification along model-induced perturbations and velocity prediction error.

velocity-error magnitude alone does not fully reflect the downstream effect of velocity prediction errors on the generated endpoint.

To further examine the connection with component ambiguity, we repeat the probe experiment under the same covariance scaling settings used in Figure 1-(c), while keeping the shared component variance fixed. This changes the covariance of the component centers while leaving the within-component factor unchanged. As shown in Figure 2-(b), the transition in measured amplification shifts with covariance scale, following the same qualitative trend as the component-ambiguity factor in Figure 1-(c). These observations support an association between component ambiguity and the observed transition in endpoint-error amplification. Together with the GMM analysis, the probe results motivate exploring greater relative training emphasis on the earlier interval before this transition.

5 Experiments

5.1 Sensitivity-Guided Weighting Refinement

The analysis in Sections 3 and 4 suggests that prediction errors before the transition can have a larger downstream effect on the generated endpoint, while the contribution of component ambiguity diminishes in the later regime. This motivates increasing the relative training emphasis on the earlier regime compared with the later one. Rather than designing a new weighting function from scratch, we introduce a simple asymmetric refinement that can be applied to an existing base weighting.

Let $W(t)$ denote the base timestep weighting. We define the refined weighting as

$$W'(t) = \frac{D(t; a, b)\, W(t)}{Z}, \quad D(t; a, b) = b^{\,3\xi(t)^2 - 2\xi(t)^3}, \quad \xi(t) = \frac{\max(t - a, 0)}{1 - a}, \tag{15}$$

where $a \in [0, 1)$ controls the onset of attenuation and $b \in (0, 1]$ controls its strength, and $Z = \int_0^1 D(t; a, b)\, W(t)\, dt$ normalizes W' to unit mean over $[0, 1]$.

The multiplier satisfies $D(t; a, b) = 1$ for $t \le a$ and decreases smoothly from 1 to b over $t \in (a, 1]$. Consequently, after mean-one normalization, the relative emphasis of the interval before a is increased while the later interval is attenuated. As illustrated in Figure 3, the refinement preserves the overall shape of the base weighting while introducing an asymmetric adjustment around the transition identified by the sensitivity analysis.

In our implementation, the onset a is determined from the component-ambiguity analysis: we choose the earliest evaluated timestep at which the component-ambiguity factor falls below $1 + 10^{-4}$. The attenuation strength b is treated as an empirical hyperparameter and is fixed to 0.3 in our main experiments. Thus, the sensitivity analysis is used to determine where the weighting profile should change, while the magnitude of the refinement remains a simple empirical choice.

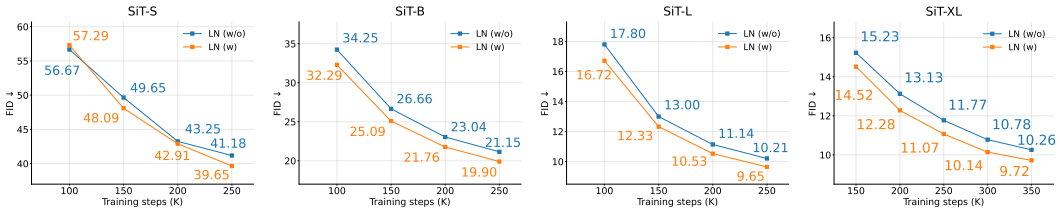


Figure 4: Generation performance across training checkpoints for SiT-S, SiT-B, SiT-L, and SiT-XL on ImageNet.

Table 1: Generation performance across SiT model scales on ImageNet. Lower FID and higher IS indicate better performance. LN denotes logit-normal weighting; LN (w/o) and LN (w) indicate the baseline and its variant with refinement, respectively.

Weighting	SiT-S		SiT-B		SiT-L		SiT-XL	
	FID↓	IS↑	FID↓	IS↑	FID↓	IS↑	FID↓	IS↑
Constant	63.71	11.12	28.21	29.29	11.24	49.75	11.35	47.73
LN (w/o)	41.18	18.15	21.15	32.94	10.21	54.19	10.26	53.36
LN (w)	**39.65**	**18.68**	**19.90**	**34.55**	**9.65**	**56.63**	**9.72**	**55.84**

5.2 Experimental Setup

Models and Weighting Baselines. We use SiT (Ma et al., 2024) as the base architecture and conduct experiments across multiple model scales. Our primary comparison is between logit-normal weighting (Esser et al., 2024) with and without the proposed refinement. To examine whether the refinement is specific to logit-normal weighting, we additionally apply it to constant weighting and CosMap weighting (Esser et al., 2024). All weighting functions are normalized to have unit mean over the training interval.

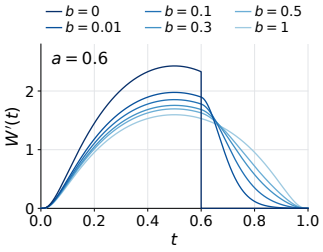


Figure 3: Effect of the asymmetric refinement on mean-one logit-normal weighting.

Datasets and Metrics. Most experiments are conducted on ImageNet (Deng et al., 2009). We additionally evaluate on CIFAR-10 (Krizhevsky & Hinton, 2009) and Food-101 (Bossard et al., 2014) to examine whether the refinement generalizes across datasets. We evaluate generation quality using Fréchet Inception Distance (FID) (Heusel et al., 2018) and Inception Score (IS) (Salimans et al., 2016), where lower FID and higher IS indicate better generation quality under the respective metrics.

Implementation Details. We train all models using the AdamW optimizer with a batch size of 1024, except for SiT-XL, for which we use a batch size of 512. We retain the default learning rates and EMA decay settings of the corresponding SiT architectures. Unless otherwise specified, we use the refinement parameters described in Section 5.1. During evaluation, we generate samples using a 50-step Euler sampler.

5.3 Main Results

We evaluate the proposed refinement across four SiT model scales on ImageNet, training SiT-S, SiT-B, and SiT-L for 250K steps and SiT-XL for 350K steps. For each model scale, we compare logit-normal weighting with and without the proposed refinement under otherwise identical training settings. Generation quality is evaluated at multiple checkpoints throughout training. The training trajectories are shown in Figure 4, while Table 1 reports the performance at the final checkpoint. Across all tested model scales, the refinement achieves lower final FID and higher final IS than the logit-normal baseline, while also yielding lower FID at nearly all evaluated training checkpoints.

To examine whether the improvement extends beyond ImageNet, we additionally evaluate SiT-S and SiT-L on CIFAR-10 and Food-101. For each dataset and model scale, we compare logit-normal weighting with and without the same refinement strategy under matched training settings. The

Table 2: Generation performance on CIFAR-10.

Weighting	SiT-S		SiT-L	
	FID↓	IS↑	FID↓	IS↑
LN (w/o)	35.56	2.22	19.56	**2.39**
LN (w)	**34.00**	**2.26**	**18.94**	2.38

Table 3: Generation performance on Food-101.

Weighting	SiT-S		SiT-L	
	FID↓	IS↑	FID↓	IS↑
LN (w/o)	22.51	30.60	15.94	39.38
LN (w)	**21.90**	**31.86**	**14.36**	**41.85**

Table 4: Refinement across base weightings.

Weighting	FID↓	IS↑
Constant (w/o)	32.47	27.40
Constant (w)	**29.74**	**30.27**
CosMap (w/o)	25.61	28.03
CosMap (w)	**24.09**	**33.64**

Table 5: FID for different attenuation onsets.

Tested a	CIFAR-10 $a = 0.4$	Food-101 $a = 0.5$	ImageNet $a = 0.6$
0.4	**34.00**	22.37	43.45
0.5	34.12	**21.90**	41.17
0.6	35.09	21.94	**39.65**

attenuation onset is selected from the component-ambiguity analysis for each dataset. As shown in Table 2 and Table 3, the refinement consistently reduces FID across both datasets and model scales, while generally improving or maintaining comparable IS. These results suggest that the benefit of the proposed refinement is not specific to ImageNet.

5.4 Ablations and Analysis

Generalization across weighting schemes. To examine whether the proposed refinement is specific to logit-normal weighting, we further apply the same refinement strategy to constant and CosMap weighting. All variants are trained using SiT-B on ImageNet for 200K steps under otherwise identical settings. As shown in Table 4, applying the refinement improves generation performance for both weighting schemes. These results indicate that the proposed refinement is not tied to a particular base weighting, but can be applied as a general modification to existing timestep weighting schemes.

Choice of attenuation onset. We next examine the choice of the attenuation onset a, which determines when the weighting begins to decrease relative to the base weighting function. Based on the component-ambiguity analysis, the transition occurs at different timesteps for different datasets, suggesting different choices of a. We therefore compare several candidate onset values around the predicted transition for ImageNet, CIFAR-10, and Food-101. As shown in Table 5, the onset suggested by the ambiguity analysis achieves the best performance among the tested choices for each dataset. This agreement supports the use of the temporal transition identified by our analysis to determine where the weighting profile should begin to change.

Consistency across training checkpoints. We further evaluate the probe at multiple training checkpoints to examine whether the observed temporal transition depends on the training stage. Using the same reference marginal flow, we compute endpoint-error amplification for models trained for 50K to 250K steps. As shown in Figure 5, the amplification magnitude varies across checkpoints at earlier timesteps, while the curves become closely aligned as t increases and exhibit a nearly identical sharp decrease around the component-ambiguity transition. This suggests that the location of the temporal transition is largely stable throughout training, supporting the use of a fixed timestep-weighting refinement during optimization.

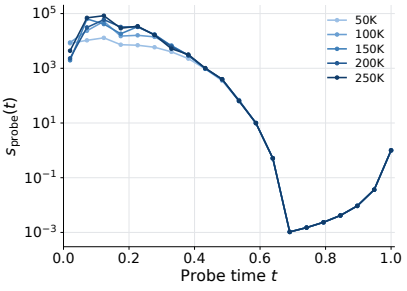


Figure 5: Endpoint-error amplification across training checkpoints.

6 Related Work

Diffusion and flow matching. Diffusion models (Sohl-Dickstein et al., 2015; Song & Ermon, 2019; Ho et al., 2020; Song et al., 2020) transform noise into data through iterative denoising, formulated either as a stochastic differential equation (SDE) or as its deterministic ordinary differential equation (ODE) counterpart (Song et al., 2020; Karras et al., 2022). Flow matching (Lipman et al., 2022; Albergo & Vanden-Eijnden, 2022; Liu et al., 2022) unifies continuous-time generative modeling by directly learning velocity fields along a predefined path between a prior and a target distribution. Subsequent work has explored architectures (Peebles & Xie, 2023; Ma et al., 2024), path constructions (Lipman et al., 2022), average-velocity formulations (Geng et al., 2025), and prediction targets (Li & He, 2026). We fix velocity prediction with a squared-error loss and analyze the ODE of the marginal flow, which lets us study error propagation directly along the trajectory.

Timestep weighting. In diffusion models, parameterizations such as noise or target prediction induce implicit timestep weightings with different training behaviors (Ho et al., 2020; Song et al., 2020; Kingma & Gao, 2023; Li & He, 2026). Explicit schemes such as Min-SNR (Hang et al., 2023) and P2 (Choi et al., 2022) reweight timesteps to improve optimization, and task-driven weightings target goals such as frequency control (Ren et al., 2026). These schemes, however, are motivated by signal-to-noise ratio, denoising behavior, or task-specific goals, and do not prescribe how to allocate emphasis to velocity errors along a flow-matching trajectory. For flow matching, time-dependent weighting has been shown to be theoretically valid and empirically important (Billera et al., 2025; Gagneux et al., 2026). CSFlow (Galinska et al., 2026) aligns the spatial frequencies generated at each stage with the human contrast sensitivity function, and FREPix (Lin et al., 2026) weights low- and high-frequency velocity errors separately over time. For 2-rectified flow, a U-shaped timestep distribution emphasizes the endpoints based on their higher training errors (Lee et al., 2024), a motivation that does not transfer to independently paired noise and data. SD3 (Esser et al., 2024) empirically compares several schemes for rectified flow, including logit-normal and CosMap, and adopts logit-normal. Given its simplicity and effectiveness, we use logit-normal as our primary baseline and evaluate whether our refinement further improves it.

Temporal behavior of flow-based generation. Oracle velocity analysis reveals a transition from an early navigation stage, where multiple data modes influence the marginal velocity, to a later refinement stage dominated by individual samples (Liu et al., 2026). From a frequency perspective, global low-frequency structure emerges before fine details (Ren et al., 2026). These studies characterize what information forms at each stage; we instead study how velocity errors at each timestep propagate to the endpoint, which relates marginal-flow dynamics directly to timestep weighting.

7 Conclusion

We analyze how velocity prediction errors at different timesteps propagate to the generated endpoint in flow matching. Under a Gaussian-mixture target, we decompose endpoint sensitivity into within-component and component-ambiguity factors, and show through model-induced probes that component ambiguity causes stronger early-stage error amplification before a characteristic transition. This motivates a refinement that increases relative emphasis before the transition, with the onset determined by the component-ambiguity analysis. Across multiple weighting schemes, datasets, and model scales, the refinement consistently improves generation performance, while additional studies support the connection between the weighting design and the analyzed marginal-flow dynamics.

These results also suggest several directions for further investigation. Our current refinement considers only a simple family of asymmetric weighting adjustments, and a broader exploration of candidate functions may further improve performance. The proposed refinement is only one instantiation of the temporal asymmetry revealed by our analysis, while the underlying insight is independent of its specific functional form. Our experiments are conducted over training horizons that already reveal stable convergence trends and consistent relative improvements, while longer training would enable a more precise assessment of the eventual performance gain and comparison with fully converged benchmarks. In addition, our analysis represents the matrix-valued Jacobian through a scalar, state-averaged approximation. This abstraction captures the dominant temporal trend, but richer directional or higher-order approximations could further characterize variations across samples and velocity-error directions, potentially leading to a more detailed understanding of flow-matching dynamics.

AI USE STATEMENT

(This section is **required** and does not count toward the page limit.)

In this work, we used generative AI tools for debugging code and for checking mathematical derivations for potential errors (e.g., typos, sign mistakes, inconsistent notation, or missing conditions). We have not used generative AI tools for generating the core research ideas, formulating the proposed method, or carrying out the theoretical derivations and proofs; all of these were conceived and developed entirely by the authors. Additionally, we used generative AI tools for improving the layout and visual presentation of figures. We have reviewed all AI-assisted work. Specifically, all code fixes suggested by AI tools were manually inspected, and the affected experiments were re-run and verified by the authors; every potential issue flagged by AI tools in the formulas was independently re-derived and checked by the authors before any change was made, and suggestions judged incorrect were discarded; and AI-assisted figure adjustments were limited to formatting and presentation, with all plotted data and results verified against our original experimental outputs. 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 and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022.

Jhon Atchison and Sheng M Shen. Logistic-normal distributions: Some properties and uses. *Biometrika*, 67(2):261–272, 1980.

Lukas Billera, Hedwig Nora Nordlinder, and Ben Murrell. Time dependent loss reweighting for flow matching and diffusion models is theoretically justified. *arXiv preprint arXiv:2511.16599*, 2025.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014.

Jooyoung Choi, Jungbeom Lee, Chaehun Shin, Sungwon Kim, Hyunwoo Kim, and Sungroh Yoon. Perception prioritized training of diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11472–11481, 2022.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

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

Anne Gagneux, Ségolène Martin, Rémi Gribonval, and Mathurin Massias. Training flow matching: The role of weighting and parameterization. *arXiv preprint arXiv:2603.06454*, 2026.

Malgorzata Galinska, Bart Pogodzinski, and Jan Eric Lenssen. Csflow: Aligning flow matching with human contrast sensitivity. *arXiv preprint arXiv:2606.08833*, 2026.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025.

Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via min-snr weighting strategy. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 7441–7451, 2023.

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, 2018. URL https://arxiv.org/abs/1706.08500.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

Diederik Kingma and Ruiqi Gao. Understanding diffusion objectives as the elbo with simple data augmentation. *Advances in Neural Information Processing Systems*, 36:65484–65516, 2023.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. URL https://arxiv.org/abs/1312.6114.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

Sangyun Lee, Zinan Lin, and Giulia Fanti. Improving the training of rectified flows, 2024. URL https://arxiv.org/abs/2405.20320.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise, 2026. URL https://arxiv.org/abs/2511.13720.

Mingfeng Lin, Jiakun Chen, Liang Han, and Liqiang Nie. Frepix: Frequency-heterogeneous flow matching for pixel-space image generation, 2026. URL https://arxiv.org/abs/2605.06421.

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

Haoming Liu, Jinnuo Liu, Yanhao Li, Liuyang Bai, Yunkai Ji, Yuanhe Guo, Shenji Wan, and Hongyi Wen. From navigation to refinement: Revealing the two-stage nature of flow-based diffusion models through oracle velocity, 2026. URL https://arxiv.org/abs/2512.02826.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

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*, pp. 23–40. Springer, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Sucheng Ren, Qihang Yu, Ju He, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. Frequency-aware flow matching for high-quality image generation. *arXiv preprint arXiv:2604.15521*, 2026.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans, 2016. URL https://arxiv.org/abs/1606.03498.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. pmlr, 2015.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, and Song Han. Sana: Efficient high-resolution image synthesis with linear diffusion transformers, 2024. URL https://arxiv.org/abs/2410.10629.

A Derivation of the GMM Sensitivity Decomposition

This appendix derives the results in Section 3.2. Throughout, $x_t = (1-t)x_0 + t\,x_1$ with $x_0 \sim \mathcal{N}(0, I)$ independent of $x_1 \sim p_1$, and $p_1(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x; \mu_k, \tau^2 I)$ with $\tau > 0$.

A.1 Sensitivity Dynamics

For a small perturbation ϵ, the state deviation satisfies $e(r) = \mathcal{F}_{r\leftarrow t}(x+\epsilon) - \mathcal{F}_{r\leftarrow t}(x) = \mathbf{S}(x, r, t)\epsilon + o(\|\epsilon\|)$, where the sensitivity matrix is the Jacobian of the flow map, $\mathbf{S}(x, r, t) = \partial \mathcal{F}_{r\leftarrow t}(x)/\partial x$. Differentiating $\partial_r \mathcal{F}_{r\leftarrow t}(x) = u_r(\mathcal{F}_{r\leftarrow t}(x))$ with respect to x and applying the chain rule gives $\partial_r \mathbf{S}(x, r, t) = J_u(x_r, r)\,\mathbf{S}(x, r, t)$, and $\mathcal{F}_{t\leftarrow t}(x) = x$ gives $\mathbf{S}(x, t, t) = I$. This is Equation 8.

A.2 Intermediate Marginals and Posterior Responsibilities

Conditioned on component k, $x_1 \sim \mathcal{N}(\mu_k, \tau^2 I)$ is independent of x_0, so x_t is a linear combination of independent Gaussians:

$$x_t \mid k \sim \mathcal{N}\big(t\mu_k, \rho_t^2 I\big)\,, \qquad \rho_t^2 = (1-t)^2 + t^2\tau^2. \tag{S1}$$

The intermediate marginal is therefore again a GMM, $p_t(x) = \sum_k \pi_k \mathcal{N}(x; t\mu_k, \rho_t^2 I)$, and Bayes' rule gives the responsibilities $\gamma_k(x, t)$ in Section 3.2. At $t = 0$ all components share the mean 0, so $\gamma_k(x, 0) = \pi_k$ for every x.

A.3 Marginal Velocity

The marginal velocity is $u_t(x) = \mathbb{E}[x_1 - x_0 \mid x_t = x] = \sum_k \gamma_k(x, t)\, u_t^{(k)}(x)$, where $u_t^{(k)}(x) = \mathbb{E}[x_1 - x_0 \mid x_t = x, k]$. Conditioned on k, (x_0, x_1, x_t) are jointly Gaussian with $\mathrm{Cov}(x_1, x_t \mid k) = t\tau^2 I$ and $\mathrm{Cov}(x_0, x_t \mid k) = (1-t)I$. Gaussian conditioning gives $\mathbb{E}[x_1 \mid x_t = x, k] = \mu_k + \frac{t\tau^2}{\rho_t^2}(x - t\mu_k)$ and $\mathbb{E}[x_0 \mid x_t = x, k] = \frac{1-t}{\rho_t^2}(x - t\mu_k)$, so

$$u_t^{(k)}(x) = \mu_k + a(t)\,(x - t\mu_k) = a(t)\,x + \big(1 - t\,a(t)\big)\mu_k, \qquad a(t) = \frac{t\tau^2 + t - 1}{\rho_t^2}. \tag{S2}$$

Using $\rho_t^2 - t(t\tau^2 + t - 1) = (1-t)^2 - t^2 + t = 1 - t$, the coefficient of μ_k is $1 - t\,a(t) = (1-t)/\rho_t^2 = b(t)$. Since $a(t)$ and $b(t)$ do not depend on k, averaging over the responsibilities yields $u_t(x) = a(t)x + b(t)\bar{\mu}(x, t)$, which is Equation 9.

A.4 Velocity Jacobian

Only $\bar{\mu}(x, t) = \sum_k \gamma_k(x, t)\mu_k$ depends on x beyond the linear term. With $\nabla_x \log \mathcal{N}(x; t\mu_k, \rho_t^2 I) = -(x - t\mu_k)/\rho_t^2$, differentiating the softmax form of γ_k gives

$$\nabla_x \gamma_k(x, t) = \gamma_k\Big(\nabla_x \log \mathcal{N}_k - \sum_j \gamma_j \nabla_x \log \mathcal{N}_j\Big) = \frac{t}{\rho_t^2}\,\gamma_k\,(\mu_k - \bar{\mu}). \tag{S3}$$

Hence $\partial\bar{\mu}/\partial x = \sum_k \mu_k \nabla_x \gamma_k^\top = \frac{t}{\rho_t^2}\sum_k \gamma_k \mu_k (\mu_k - \bar{\mu})^\top = \frac{t}{\rho_t^2} M(x, t)$, where the last step subtracts $\sum_k \gamma_k \bar{\mu}(\mu_k - \bar{\mu})^\top = 0$. Therefore

$$J_u(x, t) = a(t)I + b(t)\frac{t}{\rho_t^2} M(x, t) = a(t)I + \beta(t) M(x, t), \qquad \beta(t) = \frac{t(1-t)}{\rho_t^4},$$

which is Equation 10. Note that $\beta(t) \geq 0$ and $M(x, t) \succeq 0$, with $\beta(0) = \beta(1) = 0$.

A.5 Matrix Endpoint Sensitivity

The within-component coefficient is a logarithmic derivative of the conditional scale: since $\frac{d}{dr}\rho_r^2 = 2\big(r(1+\tau^2) - 1\big)$,

$$a(r) = \frac{1}{2}\frac{d}{dr}\log \rho_r^2, \qquad \exp\left(\int_t^1 a(r)\,dr\right) = \frac{\rho_1}{\rho_t} = \frac{\tau}{\rho_t}. \tag{S4}$$

Because $a(r)I$ commutes with every matrix, we write $\mathbf{S}(x,r,t) = \exp(\int_t^r a)\,\Phi(r)$. Substituting into Equation 8 cancels the $a(r)I$ term and leaves $\partial_r\Phi(r) = \beta(r)M(x_r,r)\Phi(r)$ with $\Phi(t) = I$, whose solution is the time-ordered exponential $\Phi(1) = \mathcal{T}\exp(\int_t^1 \beta(r)M(x_r,r)\,dr)$. Combining with Equation S4 gives Equation 11.

The within-component factor is a lower bound. For any vector v, $\partial_r\|\Phi(r)v\|_2^2 = 2\beta(r)\,(\Phi v)^\top M(x_r,r)(\Phi v) \geq 0$, so the component-ambiguity factor never contracts: every singular value of $\Phi(1)$ is at least 1. Consequently, $\mathrm{tr}(\mathbf{S}^\top\mathbf{S})/d \geq \tau^2/\rho_t^2$ for every x, and the exact scalar sensitivity satisfies $s(t) \geq \tau^2/\rho_t^2$. The within-component factor is therefore an exact lower bound on $s(t)$, and all additional amplification is attributable to component ambiguity.

A.6 Scalar Approximation

Taking the trace of $M(x,t)$ gives the per-dimension posterior spread of component centers,

$$q(x,t) = \frac{1}{d}\,\mathrm{tr}\,M(x,t) = \frac{1}{d}\Big[\sum_k \gamma_k(x,t)\|\mu_k\|_2^2 - \big\|\bar{\mu}(x,t)\big\|_2^2\Big], \tag{S5}$$

and $\bar{q}(t) = \mathbb{E}_{x\sim p_t}[q(x,t)]$. Since the reference flow transports p_t to p_r, averaging $J_u(x_r,r)$ over trajectories started from $x \sim p_t$ is equivalent to averaging over $x_r \sim p_r$, so the isotropic approximation $J_u(x_r,r) \approx g(r)I$ uses $g(r) = a(r) + \beta(r)\bar{q}(r)$ at every $r \in [t,1]$. Then $\mathbf{S}(x,1,t) \approx \exp(\int_t^1 g)\,I$, and

$$\tilde{s}(t) = \exp\left(2\int_t^1 a(r)\,dr\right)\exp\left(2\int_t^1 \beta(r)\bar{q}(r)\,dr\right) = \frac{\tau^2}{\rho_t^2}\exp\left(2\int_t^1 \beta(r)\bar{q}(r)\,dr\right),$$

which is Equation 12. The approximation replaces the state-dependent time-ordered exponential with the exponential of a state-averaged rate; it is exact for $K = 1$, where $M \equiv 0$, and it preserves the lower bound of Section A.5.

Boundary behavior of $\bar{q}(t)$. Since $\gamma_k(x,0) = \pi_k$, we have $\bar{q}(0) = \frac{1}{d}\sum_k \pi_k\|\mu_k - \mu_\pi\|_2^2$ with $\mu_\pi = \sum_k \pi_k\mu_k$, i.e., the per-dimension spread of the component centers. For $t > 0$, observing x_t is equivalent to observing $x_t/t = \mu_k + \eta_t$ with $\eta_t \sim \mathcal{N}(0, [(1-t)^2/t^2 + \tau^2]I)$, whose noise level decreases in t. The posterior spread $\bar{q}(t)$, which equals the per-dimension MMSE of the component center, therefore decreases from its prior value toward the residual ambiguity $\bar{q}(1)$. In general, $\bar{q}(1)$ need not vanish when the components overlap. Nevertheless, since $\beta(1) = 0$, the ambiguity-induced rate $\beta(t)\bar{q}(t)$ vanishes at the endpoint. When the components are well separated relative to τ, the responsibilities become nearly one-hot and $\bar{q}(t)$ itself becomes correspondingly small.