ANISOTROPIC STOCHASTICITY FOR INFERENCE-TIME SCALING OF FLOW MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Inference-time reward alignment explores multiple candidates during generation and favors those with higher intermediate rewards. Since flow models generate samples deterministically, such exploration relies on marginal-preserving stochastic differential equations (SDEs), which let duplicated candidates branch from a shared intermediate state into distinct trajectories. Prior works add Gaussian noise with equal variance across all dimensions, but much of this isotropic noise falls on directions that barely change the final generated sample. We analyze how such perturbations translate into variation among the predicted endpoints, explaining why isotropic noise can lead to inefficient exploration. Building on this analysis, we propose an anisotropic stochastic transition framework formulated as an SDE that preserves the original flow marginals. Our framework shapes the noise with the posterior covariance of the current state, allocating more noise to directions where perturbations change the final sample more. We further show that the drift adjustment required to preserve the original marginals reduces to an acceleration of the flow. Experiments on text-to-image reward alignment and offline-RL demonstrate improved reward–compute trade-offs, even after accounting for the additional computational overhead.

1 INTRODUCTION

Flow models (Lipman et al., 2023; Esser et al., 2024) have become a powerful framework for high-fidelity generation across diverse modalities. Beyond generating realistic samples, these models often need to be steered toward human preferences or task-specific objectives. Inference-time reward alignment (Uehara et al., 2025; Jain et al., 2025) achieves it by using additional test-time computation to guide generation toward higher-reward outputs, without updating model parameters.

A widely used framework to spend this additional computation is Sequential Monte Carlo (SMC) (Singhal et al., 2025; Kim et al., 2025b). SMC evolves multiple candidate samples, called particles, in parallel and evaluates their rewards at intermediate steps. Since this requires only reward evaluations, SMC can be applied to non-differentiable, black-box rewards (Singhal et al., 2025). As generation proceeds, SMC duplicates high-reward particles and discards less promising ones, directing more computation toward promising candidates.

For these copies to explore different outcomes, the transition must be stochastic, allowing them to follow different paths from the same intermediate state. In flow-based models, however, a deterministic flow ODE defines a unique trajectory from each state, which copies of the same particle evolve identically. To enable branching, the ODE can be replaced with a marginal-preserving SDE (Albergo et al., 2025; Kim et al., 2025a), allowing copies of the same particle to explore different trajectories. The resulting dynamics generate stochastic proposals from a shared state while preserving the original flow marginals.

A common choice for this stochasticity is adding *isotropic noise*, implemented by simply adding scaled standard Gaussian noise. However, not every direction of perturbation matters equally: perturbing an intermediate state along some directions noticeably changes the final sample, while perturbing it along others leaves the final sample nearly unchanged as illustrated in Fig. 1. We find that isotropic noise ignores this directional dependence and allocates much of its perturbation variance to the latter directions. Consequently, perturbed copies of the same particle can still lead to nearly identical final samples, limiting exploration of potentially higher-reward regions. This raises a ques-

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.

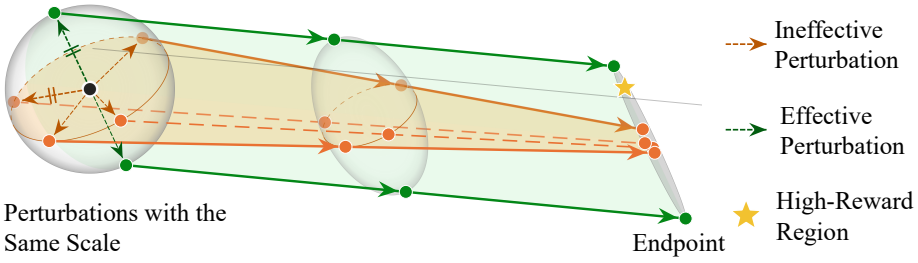


Figure 1: **Perturbation direction matters for endpoint exploration in flow models.** At an intermediate generation step, some perturbations substantially change the predicted endpoint (green), while other perturbations of the same size barely change it (orange). For reward alignment, the former are more useful since they yield distinct candidates and let the search explore a wider range of outcomes. We therefore allocate more noise to directions that strongly affect the endpoint.

tion that remains comparatively less explored: under a fixed noise budget, how should stochasticity be allocated across directions so that the copies lead to more diverse final samples?

We address this question by introducing covariance-shaped stochastic transition kernel. In linear flow matching with Gaussian prior, the posterior covariance is proportional to the endpoint-prediction Jacobian, motivating noise allocation according to endpoint sensitivity. We show that this allocation increases the leading order term of endpoint prediction variance relative to isotropic noise, under a matched perturbation scale. We further derive a marginal-preserving SDE whose drift correction simplifies to a flow acceleration term, develop a practical sampler using the pretrained velocity field, and apply it to inference-time search for reward alignment.

In summary, our main contributions are as follows: **(1)** We analyze how stochastic perturbations affect endpoint predictions, explaining why isotropic noise can lead to inefficient exploration. **(2)** We propose an anisotropic stochastic kernel shaped by the posterior covariance to address this inefficiency. This kernel increases endpoint variance while preserving the original flow marginals. **(3)** We implement these transitions within an SMC procedure and demonstrate improved empirical reward–compute trade-offs on the evaluated text-to-image generation and offline-RL tasks.

2 Preliminaries

2.1 Marginal-Preserving SDEs in Flow Models

Flow matching (Lipman et al., 2023) learns a time-dependent velocity field that transports a simple prior $p_0 = \mathcal{N}(0, I)$ to the data distribution $p_1 = p_{\text{data}}$. We consider the linear interpolant

$$x_t = (1-t)x_0 + tx_1, \qquad x_0 \sim p_0, \quad x_1 \sim p_1. \tag{1}$$

Let p_t denote its marginal density. The corresponding marginal velocity $u_t(x_t) = \mathbb{E}\left[x_1 - x_0 \mid x_t\right]$, generates the marginals through the deterministic ODE

$$dx_t = u_t(x_t)\, dt. \tag{2}$$

SDE sampling (Song et al., 2021; Albergo et al., 2025) aims to preserve the same marginals as the ODE in Eq. 2 while allowing individual trajectories to vary. For a positive semidefinite matrix $A_t(\cdot)$ and a scalar noise schedule $g_t \geq 0$, a marginal-preserving SDE is

$$dx_t = \Big[u_t(x_t) + \underbrace{\frac{g_t^2}{2}\left(\nabla \cdot A_t(x_t) + A_t(x_t)\nabla \log p_t(x_t)\right)}_{\text{drift correction}}\Big] dt + \underbrace{g_t A_t(x_t)^{1/2}\, dW_t}_{\text{stochasticity}}. \tag{3}$$

Here, W_t is standard Brownian motion. The stochasticity term lets trajectories from the same state follow different paths, while the drift correction compensates for the resulting change in probability density. Together, they preserve the ODE marginals p_t under suitable regularity.

A common choice is $A_t(\cdot) := I$, yielding **isotropic noise** and the drift correction $\frac{g_t^2}{2}\nabla \log p_t$ (Kim et al., 2025a). For the Gaussian linear interpolant, the score is directly computable from the marginal velocity field by $\nabla_{x_t} \log p_t(x_t) = (tu_t(x_t) - x_t)/(1-t)$.

2.2 Posterior Mean and Covariance in Flow Matching

For a given intermediate state x_t, the linear interpolant defines a posterior distribution over data endpoints x_1. Its posterior mean serves as an **endpoint prediction**:

$$m_{1|t}(x_t) := \mathbb{E}[x_1 \mid x_t] = x_t + (1-t)u_t(x_t). \tag{4}$$

The posterior covariance describes the uncertainty around this prediction and how it varies across directions:

$$\Sigma_{1|t}(x_t) := \mathrm{Cov}(x_1 \mid x_t) = \mathbb{E}\left[(x_1 - m_{1|t}(x_t))(x_1 - m_{1|t}(x_t))^\top \mid x_t\right]. \tag{5}$$

For an independent standard Gaussian source $x_0 \sim \mathcal{N}(0, I)$ and $t \in (0, 1)$, this covariance is related to the end-point prediction Jacobian by (Cha et al., 2026; Schioppa & Salimans, 2026)

$$\Sigma_{1|t}(x_t) = \frac{(1-t)^2}{t} J_m(x_t) = \frac{(1-t)^2}{t}\left[I + (1-t)J_{x_t}u_t(x_t)\right]. \tag{6}$$

where $J_m(x_t) := J_{x_t} m_{1|t}(x_t)$ is the endpoint-prediction Jacobian and J_{x_t} denotes the Jacobian with respect to the state x_t. The second equality follows by differentiating Eq. 4. Although these identities (Eq. 4 and 6) hold for the exact marginal velocity u_t, in practice we can substitute this with the learned velocity $u_\theta \approx u_t$, yielding estimates of both posterior moments at inference-time.

2.3 SMC-based Inference-Time Reward Alignment

Inference-time reward alignment (Uehara et al., 2025; Singhal et al., 2025) steers generation toward high-reward outputs without updating model parameters. For a reward function $r : \mathbb{R}^d \to \mathbb{R}$ and alignment strength $\beta \geq 0$, the common objective is to sample from the reward-tilted distribution

$$\pi_\beta(x) = Z_\beta^{-1} p_1(x) \exp(\beta r(x)), \qquad Z_\beta := \int_{\mathbb{R}^d} p_1(x) \exp(\beta r(x))\, dx, \tag{7}$$

where we assume $Z_\beta < \infty$ (Kim et al., 2025b).

Sequential Monte Carlo (SMC) (Singhal et al., 2025; Pani et al., 2025) approximates this target using N weighted particles $\{x_t^{(i)}\}_{i=1}^N$. At an intermediate time t, the reward $r(m_{1|t}(x_t))$ of each particle's predicted endpoint (Eq. 4) serves as a proxy for its terminal reward and informs the weight updates. When the weights concentrate, adaptive resampling draws parents with replacement according to their weights. Higher-weight particles are more likely to have multiple offspring, directing the fixed particle budget toward promising candidates.

Copies of the same parent would follow identical trajectories under the ODE in Eq. 2. We therefore use stochastic transitions to generate different continuations, a step called *branching*. Applying Euler–Maruyama discretization (Higham, 2001) to the marginal-preserving SDE in Eq. 3 gives

$$x_{t+h} = x_t + h u_t(x_t) + \frac{h g_t^2}{2}\left(\nabla \cdot A_t(x_t) + A_t(x_t)\nabla \log p_t(x_t)\right) + g_t\sqrt{h}\, A_t(x_t)^{1/2}\epsilon, \tag{8}$$

where h is the step size in Eq. 3. Each offspring uses an independent draw of $\epsilon \sim \mathcal{N}(0, I)$ to generate a candidate branch from its parent (Li et al., 2025; Kim et al., 2025b). Together, the base SDE preserves the flow marginals, while weighting and resampling favor high-reward candidates. For completeness, we leave the full SMC procedure in App. C.

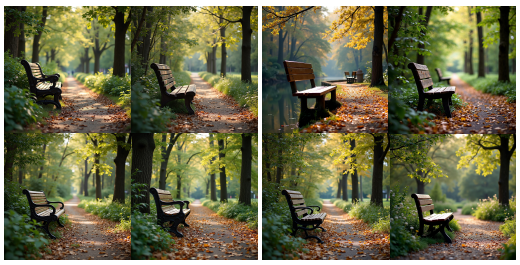
3 Understanding Isotropic Stochasticity

SMC-based reward alignment relies on stochastic transitions to generate diverse endpoint candidates from the same intermediate state and to explore high-reward regions. In this section, we study how stochastic perturbations of an intermediate state translate into variation in its endpoint predictions. We first address the empirical limitations of conventional proposals with isotropic noise in Sec. 3.1, then formally analyze the resulting endpoint-prediction variance in Sec. 3.2 to explain why isotropic stochasticity can be inefficient.

3.1 Limitation of Isotropic Exploration

Effective SMC-based search requires candidates that offer meaningful alternatives for reward-based selection, consistent with prior work (Kim et al., 2025a; Holderrieth et al., 2026) emphasizing the value of exploring diverse outcomes. However, even when stochastic transitions from a shared state x_t produce diverse intermediate states x_{t+h}, their endpoint predictions $m_{1|t+h}(x_{t+h})$ can remain nearly identical, also illustrated in Fig. 1.

We observe this issue in practice, as shown in Fig. 2(a). To isolate the effect of a single stochastic transition, we apply one SDE step from a shared intermediate state and solve all remaining steps with the deterministic ODE. Despite independent perturbations, isotropic proposals yield visually similar candidates, offering few distinct candidates for reward-based selection. This suggests that isotropic noise may spend much of its perturbation budget on directions that have little effect on the generated output. Allocating this budget to more influential directions could instead produce greater variation among candidates. Implementation details and results with varying perturbation norms are provided in App. E.1.



(a) Isotropic (b) Covariance-shaped

Figure 2: **Candidates branched from a shared state.** Final samples obtained from the same intermediate state x_t using one stochastic transition with (a) isotropic or (b) covariance-shaped noise. We choose g_t^{cov} to match the isotropic perturbation norm. Isotropic transitions yield visually similar samples. Prompt: "a photo of a bench."

3.2 Endpoint Variance Analysis

In this section, we theoretically analyze how a single SDE step diversifies endpoint predictions to identify effective perturbation directions. We approach this question by showing how the noise-shaping matrix A_t affects the variance of candidate predictions branching from the same state x_t. Each state x_{t+h} sampled using Eq. 8 yields an endpoint prediction $m_{1|t+h}(x_{t+h})$, and we quantify the spread of these predictions.

Definition 1 (Endpoint variance)**.** *For a stochastic transition from x_t to x_{t+h}, we define the* **endpoint variance** *as*

$$\mathcal{V}_h(x_t) := \operatorname{tr} \operatorname{Cov}\big(m_{1|t+h}(x_{t+h}) \mid x_t\big) .$$

This quantity measures the variation among endpoint predictions induced by branching from the same state.

Our analysis shows that this endpoint variance $\mathcal{V}_h(x_t)$ can be approximated using the noise-shaping matrix A_t of the SDE and the Jacobian of the endpoint prediction with respect to the current state.

Theorem 1. *Assume that the endpoint predictor $m_{1|t}(\cdot)$ is twice continuously differentiable,*

$$\begin{aligned} \mathcal{V}_h(x_t) &= \widetilde{\mathcal{V}}_h(x_t) + o(h), \\ \widetilde{\mathcal{V}}_h(x_t) &:= h g_t^2 \operatorname{tr}\big(J_m(x_t) A_t(x_t) J_m(x_t)^\top\big) . \end{aligned} \tag{9}$$

The proof and required assumptions are provided in App. A.2.

We use the approximation $\widetilde{\mathcal{V}}_h$ in Eq. 9 to compare the local behavior of different proposals.

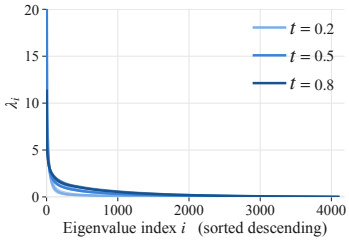


Figure 3: Eigenvalue spectra of $J_m(x_t)$ showing its effectively low-rank structure (details in App. E.2).

Theorem 1 shows that endpoint variance $\mathcal{V}_h$ tends to increase as the noise allocation $A_t(x_t)$ aligns more closely with the directions of $J_m(x_t) := J_{x_t} m_{1|t}(x_t)$. Intuitively, $J_m(x_t)$ describes how strongly perturbations in each direction affect the predicted endpoint, so allocating a fixed noise budget to more sensitive directions can produce larger endpoint variation.

The spectra in Fig. 3 give further evidence of the inefficiency of isotropic perturbations. It shows that $J_m(x_t)$ is effectively low-rank, so the endpoint prediction is sensitive to only a small number of directions. This observation agrees with prior works (Chen et al., 2024; Pi et al., 2026) that high-dimensional data such as images concentrate near lower-dimensional structures. Isotropic proposals ($A_t(\cdot) = I$), however, assign equal noise variance to every direction. Consequently, much of the perturbation budget is spent on directions that barely change the endpoint prediction.

Taken together, these intuitions motivate improving the effectiveness of stochasticity. *Can we choose A_t to induce more effective endpoint variation under the same stochasticity scale while preserving the marginal?* To address this question, we consider the posterior covariance as a choice for A_t in the next section.

4 COVARIANCE-SHAPED ANISOTROPIC STOCHASTIC TRANSITIONS

To address the limitations of isotropic perturbations, we consider *anisotropic* noise-shaping matrices $A_t(x_t)$. However, an arbitrary choice can make the drift correction required for marginal-preserving SDE sampling (Eq. 3) expensive or intractable to evaluate. We choose the posterior covariance $\Sigma_{1|t}(x_t)$ for both its directional and computational properties. By Eq. 6, it is proportional to the endpoint-prediction Jacobian, allocating more noise to directions that more strongly affect the predicted endpoint. Its drift correction simplifies to flow acceleration involving only first-order derivatives of the velocity field, as shown in Sec. 4.2.

4.1 EFFECT OF COVARIANCE SHAPING ON ENDPOINT VARIANCE

We compare isotropic and covariance-shaped proposals at the same state x_t using the endpoint variance approximation in Eq. 9. To isolate the effect of directional noise allocation, we match their total perturbation norms through trace normalization.

Proposition 1 (Comparison of noise allocations)**.** *Fix $t \in (0,1)$ and a state x_t with $\operatorname{tr}\Sigma_{1|t}(x_t) > 0$. For this comparison, use a common scalar $g_t \geq 0$, a step size h, and noise-shaping matrices*

$$\bar{A}_t^{\mathrm{cov}}(x_t) = \frac{d\Sigma_{1|t}(x_t)}{\operatorname{tr}\Sigma_{1|t}(x_t)}, \qquad A_t^{\mathrm{iso}}(x_t) = I,$$

where d is the state dimension. Both stochastic perturbations have expected squared norm $hg_t^2 d$. Under Eq. 6 and the assumptions of Theorem 1, the approximations defined in Eq. 9 satisfy

$$\widetilde{\mathcal{V}}_h^{\mathrm{cov}}(x_t) \geq \widetilde{\mathcal{V}}_h^{\mathrm{iso}}(x_t).$$

The proof is provided in App. A.3.

Proposition 1 highlights the importance of how noise is distributed across directions. The advantage of covariance shaping stems from Eq. 6: because $\Sigma_{1|t}$ is proportional to the endpoint Jacobian J_m, it allocates more noise to directions that strongly affect the endpoint prediction. The size of this advantage depends on how unevenly this sensitivity is spread across directions. Fig. 3 shows that it is concentrated in a small subset of directions, so isotropic noise spends most of its budget on directions with little effect on the endpoint. Covariance shaping exploits this structure, yielding a substantial empirical gain of approximately $94\times$ in endpoint variance over isotropic proposals at the same noise scale as reported in Tab. 1.

Table 1: Endpoint variance under matched perturbation norms. (details in App. E.2)

Method	$\widetilde{V}_h$
Isotropic	1.47
Covariance-shaped.	138.53
Ratio	$\times 94.24$

4.2 SAMPLING FROM THE COVARIANCE-SHAPED SDE

Proposition 1 shows that the direction in which noise is allocated matters. Motivated by this, we shape the noise in stochastic transitions with the posterior covariance by setting $A_t := \Sigma_{1|t}$.

Covariance-shaped Marginal-preserving SDE. Substituting this choice into Eq. 3 gives

$$dx_t = \left[u_t(x_t) + \frac{g_t^2}{2}\left(\nabla \cdot \Sigma_{1|t}(x_t) + \Sigma_{1|t}(x_t)\nabla \log p_t(x_t)\right)\right] dt + g_t \Sigma_{1|t}(x_t)^{1/2} dW_t. \tag{10}$$

Directly evaluating the drift correction in Eq. 10 is prohibitively expensive, since the covariance divergence $\nabla\cdot\Sigma_{1|t}(x_t)$ involves second-order derivatives of the velocity field. Our choice of $A_t = \Sigma_{1|t}$ avoids this cost, using the continuity equation, the entire correction reduces to a flow acceleration that involves only first-order derivatives.

Theorem 2 (Covariance-aware marginal-preserving SDE)**.** *For linear flow matching with $A_t(x_t) = \Sigma_{1|t}(x_t)$, Eq. 10 simplifies to*

$$dx_t = \left[u_t(x_t) - \frac{g_t^2}{2}(1-t)^2 a_t(x_t)\right] dt + g_t \Sigma_{1|t}(x_t)^{1/2} dW_t, \tag{11}$$

where

$$a_t(x_t) := D_t u_t(x_t) = \partial_t u_t(x_t) + J_{x_t} u_t(x_t) u_t(x_t).$$

We provide the full proof in App. A.4.

The drift simplification in Theorem 2 makes the marginal-preserving correction tractable through the acceleration a_t, without explicitly evaluating the covariance divergence or the product of the covariance with the score.

We sample from our proposed kernel introduced at Eq. 11. using Euler–Maruyama discretization:

$$x_{t+h} = x_t + h\left[u_t(x_t) - \frac{g_t^2}{2}(1-t)^2 a_t(x_t)\right] + g_t\sqrt{h}\,\Sigma_{1|t}(x_t)^{1/2}\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I). \tag{12}$$

Covariance-shaped noise sampling. Sampling covariance-shaped noise requires computing $\Sigma_{1|t}(x_t)^{1/2}\epsilon$, but explicitly forming the covariance matrix and its square root is prohibitively expensive in high dimensions. We therefore adopt the Lanczos sampler (Akhtar et al., 2026), which offers a computationally tractable approximation of this quantity using only covariance–vector products, without explicitly forming either matrix. Under the Jacobian symmetry approximation, Eq. 6 allows each covariance–vector product to be evaluated with a single VJP. Our default of two Lanczos iterations thus requires two VJPs. We examine this choice through an ablation in App. D and Tab. 6.

Computational overhead. Each covariance-shaped transition costs four NFE-equivalents, compared with one for an isotropic transition, where one NFE is the cost of a single evaluation of $u_\theta(x, t)$. The extra cost comes from two sources. Evaluating the velocity and acceleration a_t together requires one forward evaluation and one Jacobian–vector product (JVP), accounting for two NFE-equivalents. Noise sampling requires two additional vector–Jacobian products (VJPs), each counted as one NFE-equivalent.

In practice, our covariance-shaped transitions replace isotropic ones only at branching steps. All reported NFE budgets sum these costs over every evaluated particle and step, irrespective of batching. Since this accounting is a proxy for model computation, App. B provides measured wall-clock time and further implementation details.

Noise schedules. We use $g_t^{\text{iso}} = c(1-t)^2$ for isotropic transitions, following Kim et al. (2025a), and $g_t^{\text{cov}} = ct$ for covariance-shaped transitions, with $c = 3$ by default. Sec. 7.2 further analyzes these schedules and the sensitivity of performance to c.

5 Related Work

5.1 Inference-time reward alignment

Inference-time reward alignment steers pretrained diffusion and flow models toward desired objectives without updating model parameters. Gradient-based methods, such as DPS (Chung et al., 2023), guide denoising using reward gradient. SMC-based methods (Singhal et al., 2025; Kim et al., 2025b; Wu et al., 2023) instead maintain multiple particles, using reward-informed weighting and resampling to favor promising trajectories. For particle initialization, ψ-Sampler (Yoon et al., 2025) generates reward-aware particles, while PATHS (Oh et al., 2026) uses parallel tempering to explore different reward modes.

Our work studies the stochastic transitions applied at branching steps in SMC-based search algorithms for reward alignment. This transition design complements advances in algorithm design and particle selection and can be incorporated into more advanced SMC methods, including those that use reward-gradient guidance.

5.2 Covariance in Diffusion and Flow Models

A growing body of work incorporates covariance information at both training and inference time to improve the generation quality of diffusion and flow models. At training time, existing approaches learn reverse-process variances (Nichol & Dhariwal, 2021) or introduce covariance matching objectives (Ou et al., 2025). At inference time, covariance information can also be extracted from pretrained models without additional training. This enables covariance-aware sampling that improves image generation in the few-step regime (Schioppa & Salimans, 2026), as well as uncertainty quantification that identifies directions of variation in denoised images (Manor & Michaeli, 2024). These advances highlight the value of covariance information for generative model training and inference.

Our work instead focuses on its role in reward alignment for inference -time scaling, where stochastic perturbations should produce distinct endpoint predictions from a shared intermediate state. This

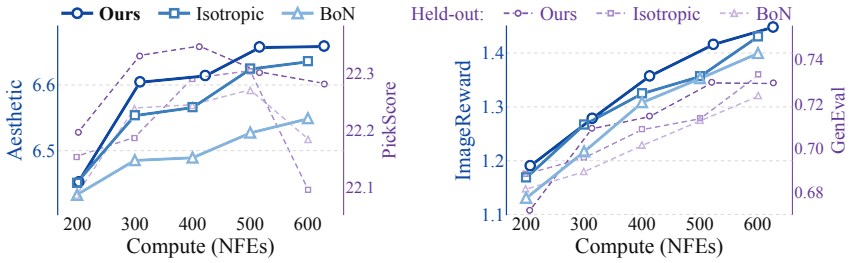


Figure 4: **Reward alignment performance on text-to-image generation.** Dashed curves show evaluation metrics that are not optimized during search (right axes). Compute is measured in NFEs and includes the cost of derivative evaluation and noise sampling.

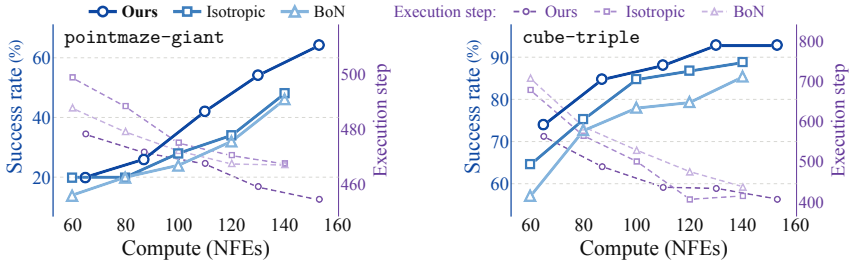


Figure 5: **Success rate results on Offline-RL tasks**. Performance on `pointmaze-giant` (left) and `cube-triple` (right). Dashed curves show execution step at inference.

perspective motivates comparing noise allocations and considering the computational cost of sampling from the corresponding marginal-preserving stochastic transition kernels. Our formulation makes posterior-covariance shaping tractable.

6 Experiments

To show the broad effectiveness of our framework, we validate our covariance-shaped stochastic transitions on two distinct domains, text-to-image generation (Sec. 6.1) and offline-RL (Sec. 6.2).

We compare against two baselines: Best-of-N (BoN), which generates N samples independently from different initial noise and selects the one with the highest reward, and SMC with isotropic transitions, following FK-Steer (Singhal et al., 2025). Our method uses the same SMC procedure and differs only at branching steps, where it replaces the transition with the covariance-shaped transition as described in Sec. 4.2. Detailed setups for baselines and our method are described in App. E.3.

6.1 Reward Alignment on Text-to-Image Generation

Setup We evaluate our method with two distinct rewards. For aesthetic alignment, we follow the setup of RBF (Kim et al., 2025a): the reward is the LAION aesthetic predictor (Schuhmann et al., 2022), and prompts are drawn from the animal prompt set of Black et al. (2024). For text–image alignment, we use ImageReward (Xu et al., 2023) as the target reward on GenEval prompts. Following prior work, we also report held-out metrics as a supplementary metric for assessing potential reward hacking. To this end, we choose PickScore (Kirstain et al., 2023) and GenEval (Ghosh et al., 2023), respectively, that are not optimized during search.

Result Fig. 4 presents results across different inference budgets, measured in NFEs. As it shows, Covariance-shaped anisotropic proposals improve reward–NFE trade-offs accounting for their additional computational cost. For aesthetic alignment, the held-out PickScore remains stable as the aesthetic reward increases, suggesting that the gains do not come from exploiting the reward at the expense of image quality. For text–image alignment, the held-out GenEval score tends to increase alongside ImageReward as inference-time computation scales, suggesting that higher optimization rewards are accompanied by improved text–image alignment. The qualitative results in Figs. 6a–6b suggest that our transitions can improve both aesthetic quality, including style and texture, and semantic alignment, such as number of objects. More qualitative results are in App. F.



(a) Aesthetic alignment (b) Text–image alignment (c) `pointmaze-giant` navigation

Figure 6: **Qualitative comparison of isotropic and covariance-shaped transitions.** Parenthesized values report (a) Aesthetic scores, (b) held-out GenEval-based detection scores, and (c) navigation success rates on maze.

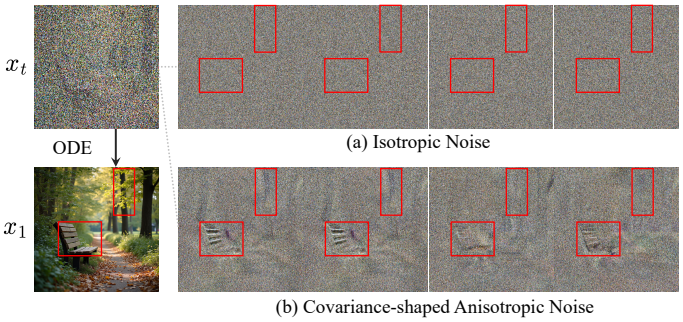


Figure 7: **Spatial structure of perturbations.** (a) Isotropic and (b) covariance-shaped perturbations sampled at the same x_t where t=0.4. Our covariance-shaped perturbations exhibit structure around the bench and background trees, as highlighted by the red boxes.

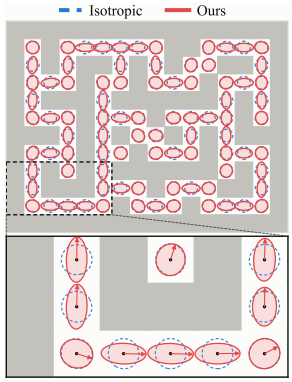


Figure 8: **Action covariance on maze.** Our method allocates more noise along navigable directions, avoiding walls.

6.2 Offline-RL

Setup For offline-RL, we evaluate our method on OGBench (Park et al., 2025) with QGF (Zhou et al., 2026) as the flow backbone, covering both navigation and object manipulation. In `pointmaze-giant`, an agent should navigate a large maze to reach a specified goal location. In `cube-triple`, a robotic arm should arrange three cubes into a target configuration through pick-and-place manipulation. For each environment, we report the mean success rate over five tasks, each evaluated with 30 seeds. We also report execution steps, defined as the number of actions the agent takes to reach the goal. More implementation details are placed in App. E.3.

Result As shown in Fig. 5, our method extends the empirical Pareto frontier in both environments, and in `pointmaze-giant` the gap widens as compute increases. It also completes the tasks in fewer environment steps, suggesting that the selected actions lead to more direct paths to the goal. These results suggest that the shape of the perturbations matters substantially, as we analyze further in Sec. 7.1: allocating noise where it matters helps particle-based search make better use of additional inference-time computation. Together, these results suggest that covariance-shaped transitions improve inference-time search across different domains.

7 Analysis

7.1 Structure of covariance-aware perturbations.

We visualize the perturbations injected by isotropic and covariance-shaped transitions in image generation and `pointmaze-giant` to analyze how noise allocation across regions and directions can affect exploration at inference-time.

Fig. 7 compares the perturbations injected by isotropic and covariance-shaped transitions at the same intermediate state x_t. Isotropic noise assigns equal variance to every dimension, regardless of x_t. Our covariance-shaped perturbations, in contrast, concentrate around the bench, the main object, and parts of the background trees, suggesting that covariance shaping reflects the structure of the image.

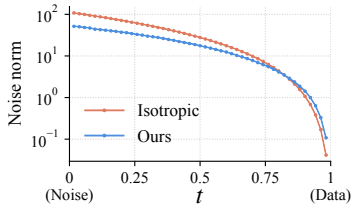


Figure 9: Perturbation norms under the schedules in Sec. 4.2.

Table 2: Aesthetic scores across perturbation scale with varying c. Matched iso. adjusts g_t during denoising to match our perturbation norm.

Method	c=1	c=3	c=5
Isotropic	6.533	6.566	6.507
Matched iso.	6.504	6.572	6.513
Ours	**6.545**	**6.614**	**6.527**

Table 3: Aesthetic reward under DAS (Kim et al., 2025b), an SMC method that combines reward-gradient guidance.

Method	Aesthetic (↑)
Isotropic	8.607
Ours	**9.082**

Because the posterior covariance captures both the magnitude and the directions of uncertainty in the endpoint prediction (Sec. 2.2), these structured perturbations explore diverse candidates rather than spending the noise on directions that leave the output nearly unchanged.

In `pointmaze-giant`, Fig. 8 depicts the injected noise at each agent position as an ellipse aligned with its principal direction, showing that the posterior covariance reflects the local navigation geometry. Because the backbone generates a chunk of future actions, we first aggregate the horizontal and vertical components of the injected noise. We then compute the covariance of this displacement at each position, normalizing its overall scale for visibility. Isotropic noise spreads equally in all directions, whereas our perturbations concentrate along directions that follow nearby walls. As a result, the noise varies the agent's motion along navigable paths rather than spending it on directions that lead into obstacles. Together, these results suggest that covariance-shaped proposals introduce variation where it is meaningful, offering a qualitative explanation of our proposed framework.

7.2 Ablations and Extensions

We show perturbation norms, sensitivity to noise scale, and compatibility with other SMC methods. All ablations and extensions in this section use aesthetic reward alignment with four particles.

Perturbation norm comparison. We use separate noise schedules for the two transitions, setting $g_t^{\text{iso}} = c(1-t)^2$ for the isotropic baseline, following Kim et al. (2025a), and $g_t^{\text{cov}} = ct$ for our method, with $c = 3$ by default. These schedules yield broadly comparable perturbation norms, accounting for both g_t and the noise-shaping matrix, as shown in Fig. 9. While the norms are not identical at every timestep, covariance-shaped transitions are applied only at branching steps (Sec. 4.2), with all remaining steps using the same isotropic transition as the baseline.

To strictly distinguish the effect of noise allocation from that of noise magnitude, we vary the noise scale coefficient c and compare our method with two isotropic baselines: the standard baseline described above and a norm-matched variant. The norm-matched baseline rescales the isotropic noise at each state x_t using $g_t^{\text{iso,match}}(x_t) = g_t^{\text{cov}}\sqrt{\text{tr}(\Sigma_{1|t}(x_t))/d}$, where d is the state dimension, while adjusting the drift correction accordingly (Eq. 3). This ensures that its total noise variance matches that of our method at the same state, while distributing the noise uniformly across directions. As shown in Tab. 2, gains from our method are consistent even under all strictly matched noise scales. Taken together, these results demonstrate that directional noise allocation is the main source of improvement in our method.

Compatibility with other SMC variants. We further verify whether covariance shaping remains beneficial when combined with advanced SMC method. We integrate our transitions into DAS (Kim et al., 2025b), an SMC method with tempered reward-gradient guidance. We follow the official DAS implementation, using a guidance scale of 0.01 and a tempering parameter of $\lambda = 0.008$. As shown in Tab. 3, replacing the isotropic transitions with our covariance-shaped transitions increases the aesthetic reward, suggesting that covariance shaping can also benefit gradient-guided SMC.

8 Conclusion

We studied how the shape of the injected stochasticity affects endpoint diversity in marginal-preserving SDE sampling. Our analysis shows that isotropic noise spends much of its budget on directions that barely change the endpoint prediction. We therefore propose covariance-shaped transitions whose marginal-preserving drift correction reduces to a tractable flow acceleration. Applied at the SMC, they improve reward–compute trade-offs on text-to-image and offline-RL, suggesting that stochastic transition design is a complementary component of inference-time scaling.

AI USE STATEMENT

In this work, we used generative AI tools solely to assist with text readability. We did not use AI to develop theoretical frameworks, design experiments, or interpret results. The authors have manually reviewed and verified all AI-assisted content for accuracy. We take full responsibility for the final content and claims of this work.

ETHICS STATEMENT

We acknowledge and strictly adhere to the ICLR Code of Ethics. This work advances generative modeling by improving sample quality and generation efficiency. While intended for beneficial applications, these improvements could also be misused to generate deceptive or harmful content.

REPRODUCIBILITY STATEMENT

To ensure the reproducibility of this work, we have provided all necessary details required to replicate our results in Appendix. This includes the detailed experimental setups, hyperparameter configurations, and mathematical derivations. We encourage readers to refer to this appendix for a comprehensive guide to reproducing our method and experiments.

REFERENCES

Md Sahil Akhtar, Aymane El Gadarri, Vivek F Farias, and Adam D Jozefiak. The value of covariance matching in gaussian ddpms and the lanczos sampler. *arXiv preprint arXiv:2605.22723*, 2026.

Michael Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Black Forest Labs. Announcing Black Forest Labs. Blog post on Black Forest Labs website, August 2024. URL `https://blackforestlabs.ai/announcing-black-forest-labs/`. Accessed: 2025-05-14.

Yeonwoo Cha, Jaehoon Yoo, Semin Kim, Yunseo Park, Jinhyeon Kwon, and Seunghoon Hong. Training-free refinement of flow matching with divergence-based sampling. *arXiv preprint arXiv:2604.04646*, 2026.

Siyi Chen, Huijie Zhang, Minzhe Guo, Yifu Lu, Peng Wang, and Qing Qu. Exploring low-dimensional subspace in diffusion models for controllable image editing. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2023.

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

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

Desmond J. Higham. An algorithmic introduction to numerical simulation of stochastic differential equations. *SIAM Review*, 43(3):525–546, 2001.

Peter Holderrieth, Douglas Chen, Luca Eyring, Ishin Shah, Giri Anantharaman, Yutong He, Zeynep Akata, Tommi Jaakkola, Nicholas Matthew Boffi, and Max Simchowitz. Diamond maps: Efficient reward alignment via stochastic flow maps. In *Forty-third International Conference on Machine Learning*, 2026.

Vineet Jain, Kusha Sareen, Mohammad Pedramfar, and Siamak Ravanbakhsh. Diffusion tree sampling: Scalable inference-time alignment of diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Jaihoon Kim, Taehoon Yoon, Jisung Hwang, and Minhyuk Sung. Inference-time scaling for flow models via stochastic generation and rollover budget forcing. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 30830–30864, 2025a.

Sunwoo Kim, Minkyu Kim, and Dongmin Park. Test-time alignment of diffusion models without reward over-optimization. In *International Conference on Learning Representations*, volume 2025, pp. 87238–87272, 2025b.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022.

Xiner Li, Yulai Zhao, Chenyu Wang, Gabriele Scalia, Gökcen Eraslan, Surag Nair, Tommaso Biancalani, Shuiwang Ji, Aviv Regev, Sergey Levine, and Masatoshi Uehara. Derivative-free guidance in continuous and discrete diffusion models with soft value-based decoding. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

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

Hila Manor and Tomer Michaeli. On the posterior distribution in denoising: Application to uncertainty quantification. In *The Twelfth International Conference on Learning Representations*, 2024.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8162–8171, 2021.

Myeongjun Oh, Gwangho Kim, and Sungyoon Lee. Parallel tempering initial sampling in inference-time reward alignment. In *ICML 2026 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2026.

Zijing Ou, Mingtian Zhang, Andi Zhang, Tim Z. Xiao, Yingzhen Li, and David Barber. Improving probabilistic diffusion models with optimal diagonal covariance matching. In *The Thirteenth International Conference on Learning Representations*, 2025.

Chinmay Pani, Zijing Ou, and Yingzhen Li. Test-time alignment of discrete diffusion models with sequential monte carlo. In *Second Workshop on Test-Time Adaptation: Putting Updates to the Test! at ICML 2025*, 2025.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. OGBench: Benchmarking offline goal-conditioned RL. In *The Thirteenth International Conference on Learning Representations*, 2025.

Sophia Pi, Mingcheng Lu, Maojiang Su, Weimin Wu, Jerry Yao-Chieh Hu, and Han Liu. Learning manifold data with flow matching. In *Forty-third International Conference on Machine Learning*, 2026.

Andrea Schioppa and Tim Salimans. Covariance-aware sampling for diffusion models. *arXiv preprint arXiv:2605.13910*, 2026.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5b: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

Raghav Singhal, Zachary Horvitz, Ryan Teehan, Mengye Ren, Zhou Yu, Kathleen McKeown, and Rajesh Ranganath. A general framework for inference-time scaling and steering of diffusion models. In *Forty-second International Conference on Machine Learning*, 2025.

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

Masatoshi Uehara, Yulai Zhao, Chenyu Wang, Xiner Li, Aviv Regev, Sergey Levine, and Tommaso Biancalani. Inference-time alignment in diffusion models with reward-guided generation: Tutorial and review. *arXiv preprint arXiv:2501.09685*, 2025.

Luhuan Wu, Brian L. Trippe, Christian A Naesseth, John Patrick Cunningham, and David Blei. Practical and asymptotically exact conditional sampling in diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

TaeHoon Yoon, Yunhong Min, Kyeongmin Yeo, and Minhyuk Sung. ψ-sampler: Initial particle sampling for SMC-based inference-time reward alignment in score models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Zhiyuan Zhou, Andy Peng, Charles Xu, Qiyang Li, Tobias Springenberg, Kevin Frans, and Sergey Levine. Test-time gradient guidance of flow policies in reinforcement learning. *arXiv preprint arXiv:2606.11087*, 2026.

A PROOFS AND DERIVATIONS

In this section we provide proofs and derivations for the theorems and propositions. Sec. A.1 briefly recalls the setup and clarifies the conditions used in this section. We then establish Theorem 1 in Sec. A.2 by justifying approximation error of endpoint variance, which is used in Sec. A.3 to compare effective variance induced by a isotropic-shaped and a covariance-shaped perturbations, yielding Proposition 1. Finally, we present the proof of Theorem 2 in Sec. A.4

A.1 DEFINITIONS AND ASSUMPTIONS

Endpoint predictor Consider a flow ODE from noise $x_0 \in \mathbb{R}^d \sim p_0$ to data $x_1 \in \mathbb{R}^d \sim p_1$

$$dx_t = u_t(x_t)dt,$$

with linear interpolant

$$x_t = (1-t)x_0 + tx_1, \qquad u_t(x_t) = \mathbb{E}[x_1 - x_0 \mid x_t],$$

which constructs intermediate marginals $x_t \sim p_t$. The endpoint predictor

$$m_{1|t}(\cdot) : x_t \mapsto x_t + (1-t)u_t(x_t), \tag{13}$$

maps intermediate state x_t to endpoint prediction $m_{1|t}(x_t)$, which is posterior mean. We assume that $m_{1|t}(\cdot)$is twice continuously differentiable, its Jacobian is jointly continuous in t and x, and its second derivatives have at most polynomial growth, uniformly over neighboring time. The posterior covariance is written as

$$\Sigma_{1|t}(x_t) := \mathrm{Cov}(x_1 \mid x_t).$$

Gaussian transition preserving marginal Setting up marginal-preserving SDE

$$\begin{aligned} dx_t =& b_t(x_t)dt + g_t A_t(x_t)^{1/2} dW_t, \\ b_t(x_t) =& u_t(x_t) + \frac{g_t^2}{2}(\nabla \cdot A_t(x_t) + A_t(x_t)\nabla \log p_t(x_t)) \end{aligned}$$

with positive semidefinite matrix A_t derives an Euler-Maruyama step

$$x_{t+h} = \underbrace{x_t + hb_t(x_t)}_{=:\bar{x}_{t+h}} + \underbrace{g_t\sqrt{h}A_t(x_t)^{1/2}\epsilon}_{=:\delta_h}, \qquad \epsilon \sim \mathcal{N}(0, I),$$

By defining

$$\bar{x}_{t+h} := x_t + hb_t(x_t), \qquad \delta_h := g_t\sqrt{h}A_t(x_t)^{1/2}\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I),$$

the solver step can be written as a sum of the expectation and the zero-centered stochastic term as

$$x_{t+h} = \bar{x}_{t+h} + \delta_h. \tag{14}$$

We define endpoint variance as

$$\mathcal{V}_h(x_t) := \mathrm{tr}\,\mathrm{Cov}\big(m_{1|t+h}(x_{t+h}) \mid x_t\big)$$

to measure how the endpoint is expected to vary by this transition.

A.2 PROOF OF ENDPOINT VARIANCE APPROXIMATION

This section justifies the approximation of endpoint variance suggested at Theorem 1, under settings and assumptions described in App. A.1. Here, $O_p(\cdot)$ and $o_p(\cdot)$ denote the order in probability, and $O_{L^2}(h)$ or $o_{L^2}(h)$ are used to represent a random vector, whose root mean square is $O(h)$ or $o(h)$, respectively. To represent Euclidean norm or Frobenius norm, $\|\cdot\|$ is used. The symbol $C > 0$ denotes a constant independent of sufficiently small h, whose value may change from line to line.

Proof. Fix $t \in (0,1)$ and x_t. Consider the updated endpoint prediction conditioned on the sample x_{t+h}

$$m_{1|t+h}(x_{t+h})$$

Conduct first-order Taylor expansion on $m_{1|t+h}(x_{t+h})$ around $\bar{x}_{t+h}$ to get

$$m_{1|t+h}(x_{t+h}) = m_{1|t+h}(\bar{x}_{t+h}) + J_{x_{t+h}} m_{1|t+h}(\bar{x}_{t+h})\delta_h + R_h, \tag{15}$$

where the remainder R_h is

$$R_h = \int_0^1 (1-r) D^2_{x_{t+h}} m_{1|t+h}(\bar{x}_{t+h} + r\delta_h)[\delta_h, \delta_h] dr$$

Since $m_{1|t}(\cdot)$ has second derivative with polynomial bound, there exist a constant $C > 0$ and $q > 0$ such that

$$\|D^2_{x_{t+h}} m_{1|t+h}(x_{t+h})\| \le C(1 + \|x_{t+h}\|^q), \qquad x_{t+h} \in \mathbb{R}^d.$$

Therefore when $\bar{x}_{t+h} \to x_t$, so that $\bar{x}_{t+h}$ is bounded at small h,

$$\begin{aligned}
\|R_h\| =& \left\| \int_0^1 (1-r) D^2_{x_{t+h}} m_{1|t+h}(\bar{x}_{t+h} + r\delta_h)[\delta_h, \delta_h] dr \right\| \\
\le& \int_0^1 (1-r) \|D^2_{x_{t+h}} m_{1|t+h}(\bar{x}_{t+h} + r\delta_h)[\delta_h, \delta_h]\| dr \\
\le& \int_0^1 (1-r) \|D^2_{x_{t+h}} m_{1|t+h}(\bar{x}_{t+h} + r\delta_h)\| \|\delta_h\|^2 dr \\
\le& C(1 + \|\bar{x}_{t+h}\|^q + \|\delta_h\|^q)\|\delta_h\|^2 \\
\le& C(1 + \|\delta_h\|^q)\|\delta_h\|^2
\end{aligned}$$

for some constant C. Taking expectation on square of both sides leads to

$$\begin{aligned}
\mathbb{E}[\|R_h\|^2] \le& C\mathbb{E}\big[\|\delta_h\|^4 + 2\|\delta_h\|^{q+4} + \|\delta_h\|^{2q+4}\big] \\
=& O(h^2).
\end{aligned}$$

Since $\delta_h = \sqrt{h}\, g_t A_t(x_t)^{1/2}\epsilon$ with $\epsilon \sim \mathcal{N}(0, I)$, we have $\mathbb{E}\|\delta_h\|^r = O(h^{r/2})$ for every $r > 0$. Hence $\mathbb{E}\|R_h\|^2 = O(h^2)$, which implies $R_h = O_{L^2}(h)$. Hence by definition of $O_{L^2}(\cdot)$,

$$R_h = O_{L^2}(h).$$

Substituting this back to Eq. 15 leads to

$$m_{1|t+h}(x_{t+h}) = m_{1|t+h}(\bar{x}_{t+h}) + J_{x_{t+h}} m_{1|t+h}(\bar{x}_{t+h})\delta_h + O_{L^2}(h).$$

Since $\bar{x}_{t+h} \to x_t$, joint continuity of endpoint prediction implies

$$J_{x_{t+h}} m_{1|t+h}(\bar{x}_{t+h}) = J_x m_{1|t}(x_t) + o(1).$$

Therefore

$$m_{1|t+h}(x_{t+h}) = m_{1|t+h}(\bar{x}_{t+h}) + J_{x_t} m_{1|t}(x_t)\delta_h + o(1)\delta_h + O_{L^2}(h).$$

Again,

$$\mathbb{E}[\|\delta_h\|^2] = O(h), \qquad \mathbb{E}[\|\delta_h\|^2 o(1)^2] = o(h)$$

leads to

$$m_{1|t+h}(x_{t+h}) = m_{1|t+h}(\bar{x}_{t+h}) + J_{x_t} m_{1|t}(x_t)\delta_h + o_{L^2}(\sqrt{h}).$$

Therefore subtracting expectation of both sides yields

$$m_{1|t+h}(x_{t+h}) - \mathbb{E}[m_{1|t+h}(x_{t+h})] = J_{x_t} m_{1|t}(x_t)\delta_h + o_{L^2}(\sqrt{h}).$$

Taking trace of covariance of endpoint predictor to estimate endpoint variance, gives

$$\begin{aligned}
\mathcal{V}_h(x_t) :=& \operatorname{tr}\operatorname{Cov}\big(m_{1|t+h}(x_{t+h}) \mid x_t\big) \\
=& \mathbb{E}\left[\left\|m_{1|t+h}(x_{t+h}) - \mathbb{E}[m_{1|t+h}(x_{t+h})]\right\|^2 \mid x_t\right] \\
=& \mathbb{E}[\|J_{x_t} m_{1|t}(x_t)\delta_h + o_{L^2}(\sqrt{h})\|^2 \mid x_t] \\
=& \mathbb{E}[\|J_{x_t} m_{1|t}(x_t)\delta_h\|^2 \mid x_t] + \mathbb{E}[2[J_{x_t} m_{1|t}(x_t)\delta_h]^\top o_{L^2}(\sqrt{h}) \mid x_t] \\
&+ \mathbb{E}[\|o_{L^2}(\sqrt{h})\|^2 \mid x_t].
\end{aligned}$$

Let $L_h := J_{x_t} m_{1|t}(x_t)\delta_h$, and denote the centered remainder by $r_h = o_{L^2}(\sqrt{h})$. Since $L_h = O_{L^2}(\sqrt{h})$, the Cauchy–Schwarz inequality gives

$$\begin{aligned}\left|\mathbb{E}[2L_h^\top r_h \mid x_t]\right| &\le 2\sqrt{\mathbb{E}[\|L_h\|^2 \mid x_t]}\sqrt{\mathbb{E}[\|r_h\|^2 \mid x_t]} \\ &= O(\sqrt{h})\, o(\sqrt{h}) \\ &= o(h).\end{aligned}$$

Hence,

$$\begin{aligned}\mathcal{V}_h(x_t) =& \mathbb{E}\left[\|J_{x_t} m_{1|t}(x_t)\delta_h\|^2 \mid x_t\right] + o(h) \\ =& \operatorname{tr}\operatorname{Cov}\left[J_{x_t} m_{1|t}(x_t)\delta_h \mid x_t\right] + o(h) \\ =& hg_t^2 \operatorname{tr}(J_{x_t} m_{1|t}(x_t) A_t(x_t) J_{x_t} m_{1|t}(x_t)^\top) + o(h).\end{aligned}$$

Therefore approximation below is justified.

$$\begin{aligned}\mathcal{V}_h(x_t) =& \widetilde{\mathcal{V}}_h(x_t) + o(h) \\ \widetilde{\mathcal{V}}_h(x_t) :=& hg_t^2 \operatorname{tr}(J_{x_t} m_{1|t}(x_t) A_t(x_t) J_{x_t} m_{1|t}(x_t)^\top)\end{aligned} \tag{16}$$

□

A.3 Derivation of Endpoint Variance Inequality

Based on the endpoint variance approximation shown in App. A.2, this section shows derivation of inequality between endpoint variance approximations induced by isotropic and covariance-shaped proposal, which is presented at Proposition. 1.

Proof. Fix $t \in (0,1)$ and $x_t \in \mathbb{R}^d$. For the isotropic transition and covariance shaped transition, let the noise shaping matrices

$$A_t^{\mathrm{iso}}(x_t) = I, \quad \bar{A}_t^{\mathrm{cov}}(x_t) = \frac{d\Sigma_{1|t}(x_t)}{\operatorname{tr}\Sigma_{1|t}(x_t)}$$

respectively, to match the expected squared norm of perturbation by trace normalization. Assume $\operatorname{tr}\Sigma_{1|t}(x_t) \neq 0$. Recall that from Eq. 6, Jacobian-covariance equality holds as

$$J_{x_t} m_{1|t}(x_t) = \frac{t}{(1-t)^2}\Sigma_{1|t}(x_t)$$

and from Eq. 16, the endpoint variance can be approximated with

$$\widetilde{\mathcal{V}}_h(x_t) := hg_t^2 \operatorname{tr}(J_{x_t} m_{1|t}(x_t) A_t(x_t) J_{x_t} m_{1|t}(x_t)^\top).$$

The isotropic noise shaping matrix yields endpoint variance approximation as

$$\begin{aligned}\widetilde{\mathcal{V}}_h^{\mathrm{iso}} =& hg_t^2 \frac{t^2}{(1-t)^4} \operatorname{tr}\left(\Sigma_{1|t}^2\right) \\ =& hg_t^2 \frac{t^2}{(1-t)^4} \sum_i \lambda_i^2\end{aligned}$$

where λ_i's are the eigenvalues of $\Sigma_{1|t}$. Notice that from the symmetry of covariance matrix, we can substitute $\Sigma_{1|t}$ to $\Sigma_{1|t}^\top$ and identify λ_i's as real values. Similarly, the covariance shaped noise leads to the endpoint variance approximation

$$\begin{aligned}\widetilde{\mathcal{V}}_h^{\mathrm{cov}} =& hg_t^2 \frac{t^2}{(1-t)^4} \frac{d \operatorname{tr}\left(\Sigma_{1|t}^3\right)}{\operatorname{tr}\left(\Sigma_{1|t}\right)} \\ =& hg_t^2 \frac{t^2}{(1-t)^4} \frac{d\sum_i \lambda_i^3}{\sum_i \lambda_i}.\end{aligned}$$

Let the difference between $\widetilde{\mathcal{V}}_h^{\text{cov}}$ and $\widetilde{\mathcal{V}}_h^{\text{iso}}$ as

$$\Delta\widetilde{\mathcal{V}}_h := \widetilde{\mathcal{V}}_h^{\text{cov}} - \widetilde{\mathcal{V}}_h^{\text{iso}}.$$

Then

$$\begin{aligned}
\Delta\widetilde{\mathcal{V}}_h =& hg_t^2 \frac{t^2}{(1-t)^4}\left[\frac{d\sum_i \lambda_i^3}{\sum_i \lambda_i} - \sum_i \lambda_i^2\right] \\
=& hg_t^2 \frac{t^2}{(1-t)^4}\left[\sum_i \lambda_i\right]^{-1}\left[d\sum_i \lambda_i^3 - \sum_i \lambda_i^2 \sum_i \lambda_i\right] \\
=& hg_t^2 \frac{t^2}{(1-t)^4}\left[\sum_i \lambda_i\right]^{-1}\left[d\sum_i \lambda_i^3 - \sum_i \lambda_i^3 - \sum_{i\neq j} \lambda_i^2\lambda_j\right] \\
=& hg_t^2 \frac{t^2}{(1-t)^4}\left[\sum_i \lambda_i\right]^{-1}\left[(d-1)\sum_i \lambda_i^3 - \sum_{i\neq j} \lambda_i^2\lambda_j\right] \\
=& hg_t^2 \frac{t^2}{(1-t)^4}\left[\sum_i \lambda_i\right]^{-1}\left[\sum_{i<j}(\lambda_i^3 + \lambda_j^3 - \lambda_i^2\lambda_j - \lambda_j^2\lambda_i)\right] \\
=& hg_t^2 \frac{t^2}{(1-t)^4}\left[\sum_i \lambda_i\right]^{-1}\left[\sum_{i<j}(\lambda_i - \lambda_j)^2(\lambda_i + \lambda_j)\right] \geq 0.
\end{aligned}$$

It is a nonnegative value since a covariance matrix is positive semidefinite and has nonnegative eigenvalues. Therefore

$$\Delta\widetilde{\mathcal{V}}_h \geq 0 \quad \Leftrightarrow \quad \widetilde{\mathcal{V}}_h^{\text{cov}} \geq \widetilde{\mathcal{V}}_h^{\text{iso}}.$$

□

A.4 Acceleration Form of The Drift Correction

Below shows the proof of Theorem 2. Assume finite second moments and sufficient regularity for the differentiation and the integration by parts below. We use the convention

$$(\nabla \cdot M)_i = \Sigma_j \partial_{x_j} M_{ij}$$

for matrix M.

Proof. The density and marginal velocity satisfy the continuity equation.

$$\partial_t p_t + \nabla \cdot (p_t u_t) = 0$$

Consider a linear interpolant path through x_t. Since the velocity is constant over time,

$$v = \frac{x_1 - x_t}{1-t}.$$

Therefore the covariance of velocity conditioned on x_t is equal to the scaled endpoint prediction covariance as

$$\text{Cov}(v \mid x_t) = \frac{1}{(1-t)^2}\Sigma_{1|t}(x_t).$$

Consequently, conditional second moment of v is then written as

$$\mathbb{E}[vv^\top \mid x_t] = u_t(x_t)u_t(x_t)^\top + \frac{\Sigma_{1|t}(x_t)}{(1-t)^2}.$$

Consider an arbitrary smooth function $\phi(\cdot) : \mathbb{R}^d \mapsto \mathbb{R}$ that satisfies $[\phi(x)]_{-\infty}^{\infty} = 0$. Then

$$\begin{aligned}\frac{d}{dt}\int \phi(x)p_t(x)u_t(x)dx =& \frac{d}{dt}\mathbb{E}\left[\phi(x_t)u_t(x_t)\right] \\ =& \frac{d}{dt}\mathbb{E}\left[\phi(x_t)\mathbb{E}[v \mid x_t]\right] \\ =& \frac{d}{dt}\mathbb{E}\left[\phi(x_t)v\right] \\ =& \mathbb{E}\left[v\frac{d}{dt}\phi(x_t) + \phi(x_t)\frac{d}{dt}v\right].\end{aligned} \tag{17}$$

Since

$$\frac{dx_t}{dt} = v_t, \qquad \frac{dv}{dt} = 0,$$

Eq. 17 written as

$$\begin{aligned}\mathbb{E}_{x_t\sim p_{x_t}, v\sim p(\cdot|x_t)}\left[v\left(\frac{dx_t}{dt}\right)^\top \nabla_x\phi(x_t)\right] =& \mathbb{E}_{x_t\sim p_{x_t}, v\sim p(\cdot|x_t)}[vv^\top\nabla_x\phi(x_t)] \\ =& \int p_t(x)\mathbb{E}_{v\sim p(\cdot|x_t=x)}[vv^\top]\nabla_x\phi(x)dx \\ =& \int p_t(x)\left[u_t(x)u_t(x)^\top + \frac{\Sigma_{1|t}(x)}{(1-t)^2}\right]\nabla_x\phi(x)dx \\ =& \int p_t(x)u_t(x)u_t(x)^\top\nabla_x\phi(x)dx \\ &+ \int p_t(x)\frac{\Sigma_{1|t}(x)}{(1-t)^2}\nabla_x\phi(x)dx\end{aligned}$$

Here, integration by parts gives

$$\begin{aligned}&\int p_t(x)u_t(x)u_t(x)^\top\nabla_x\phi(x)dx + \int p_t(x)\frac{\Sigma_{1|t}(x)}{(1-t)^2}\nabla_x\phi(x)dx \\ =& -\int \phi(x)\nabla\cdot(p_t(x)u_t(x)u_t(x)^\top)dx - \frac{1}{(1-t)^2}\int\phi(x)\nabla\cdot(p_t(x)\Sigma_{1|t}(x))dx\end{aligned} \tag{18}$$

However, the initial term of Eq. 17 is also equivalent as

$$\int \frac{d}{dt}[\phi(x)p_t(x)u_t(x)]dx = \int \phi(x)\partial_t[p_t(x)u_t(x)]dx \tag{19}$$

Since Eq. 18 and Eq. 19 are equivalent for arbitrary $\phi(\cdot)$,

$$\partial_t[p_t(x)u_t(x)] + \nabla\cdot(p_t(x)u_t(x)u_t(x)^\top) + \frac{1}{(1-t)^2}\nabla\cdot(p_t(x)\Sigma_{1|t}(x)) = 0 \tag{20}$$

holds. Define the acceleration

$$a_t(x) := \partial_t u_t(x) + J_x u_t(x)u_t(x)$$

and use continuity equation to derive

$$\begin{aligned}&\partial_t(p_t(x)u_t(x)) + \nabla\cdot(p_t(x)u_t(x)u_t(x)^\top) \\ &\quad= u_t(x)\partial_t p_t(x) + p_t(x)\partial_t u_t(x) + p_t(x)J_x u_t(x)u_t(x) + u_t(x)\nabla\cdot(p_t(x)u_t(x)) \\ &\quad= p_t(x)(\partial_t u_t(x) + J_x u_t(x)u_t(x)) + u_t(x)(\partial_t p_t(x) + \nabla\cdot(p_t(x)u_t(x))) \\ &\quad= p_t(x)a_t(x)\end{aligned}$$

Substituting this to Eq. 20 gives

$$\begin{aligned}a_t(x) &= -\frac{1}{p_t(x)(1-t)^2}\nabla\cdot(p_t(x)\Sigma_{1|t}(x)) \\ &= -\frac{1}{(1-t)^2}[\nabla\cdot\Sigma_{1|t}(x) + \Sigma_{1|t}\nabla\log p_t(x)].\end{aligned}$$

Model	Kernel	Wall-clock time (s)	Relative latency	Peak memory (GiB)
FLUX.1-dev (Aesthetic)	Isotropic	0.39	–	38.6
	Covariance-shaped	1.33	3.41×	40.6
QGF (`pointmaze-giant`)	Isotropic	0.57	–	1.14
	Covariance-shaped	1.17	2.05×	1.72

Table 4: Computational cost comparison per stochastic transition. Each transition processes the indicated particle batch, including velocity evaluation, drift correction, noise sampling, and state updates. Memory denotes the maximum peak allocated GPU memory across repetitions.

Consequently, the SDE at Eq. 11 reduces to

$$dx_t = \left[u_t(x_t) - \frac{g_t^2}{2}(1-t)^2 a_t(x_t)\right] dt + g_t \Sigma_{1|t}(x_t)^{1/2} dW_t$$

□

B Computational Overhead

Reusing the velocity computation graph. As mentioned in Sec. 4.2, we reuse the velocity computation graph across acceleration evaluation and covariance-shaped noise sampling to avoid repeated forward evaluations during Lanczos iterations. At a fixed state (x_t, t), the flow acceleration is

$$a_t(x_t) = \partial_t u_t(x_t) + J_x u_t(x_t) u_t(x_t).$$

This expression can be evaluated as a Jacobian–vector product of $(x, t) \mapsto u_t(x)$ along the direction $(u_t(x_t), 1)$, with the direction treated as fixed during differentiation. We retain the reverse-mode graph of the velocity output for the subsequent noise-sampling step.

Each Lanczos iteration requires a covariance–vector product. By Eq. 6, for a vector v,

$$\Sigma_{1|t}(x_t) v = \frac{(1-t)^2}{t} \left[v + (1-t) J_x u_t(x_t) v\right]. \tag{21}$$

For the exact flow velocity, the covariance identity implies that $J_x u_t(x_t)$ is symmetric. We therefore evaluate these products using vector–Jacobian products,

$$J_x u_t(x_t)^\top v = \nabla_{x_t} \left\langle u_t(x_t), v \right\rangle, \tag{22}$$

computed by backpropagating through the retained velocity graph, with v treated as fixed. For a learned velocity model, this substitution relies on approximate Jacobian symmetry; it is not an exact identity for an arbitrary neural network.

The same graph is reused for all Lanczos vectors at the current state. Thus, each iteration requires one additional backward pass but no additional velocity forward pass. For m Lanczos iterations, the incremental sampling cost consists of m backward passes and the small-matrix operations within Lanczos. Graph reuse reduces repeated computation at the cost of retaining intermediate activations. We release the graph after sampling and construct a new one at the next state.

Measured computational cost. To complement the NFE accounting, we report wall-clock time and peak GPU memory usage. Tab. 4 compares the latency and memory usage of individual transitions, while Tab. 5 reports overall runtime and task performance. Since covariance-shaped transitions are applied only at resampling steps, their higher per-transition cost results in a modest increase in overall runtime. All measurements were performed with four particles on an NVIDIA H200 GPU.

C Sequential Monte Carlo for Reward Alignment

SMC with max potentials. To prioritize high-reward samples, we adopt the max potential used in the default text-to-image configuration of FK-Steer (Singhal et al., 2025). For each particle i, we

Task	Kernel	Resampling count	Wall-clock time (s)	Performance (↑)
FLUX.1-dev (Aesthetic)	Isotropic	2.3	41.72	6.566
	Covariance-shaped	2.4	44.25	6.614
QGF (`pointmaze-giant`)	Isotropic	1.0	13.34	0.200
	Covariance-shaped	1.0	14.35	0.267

Table 5: Runtime and performance under adaptive resampling. Runtime is measured in seconds per returned image or action, including search-time reward evaluations and selection. Performance denotes Aesthetic score, and success rate for the corresponding tasks.

maintain its current state $x^{(i)}$, normalized weight $w^{(i)}$, maximum intermediate reward $M^{(i)}$, and accumulated log potential $S^{(i)}$ along its ancestral trajectory. These quantities are updated in place as time advances from 0 to 1. Intermediate rewards are evaluated using $m_{1|t}(x) = x + (1-t)u_\theta(x,t)$. Resampling jointly copies $(x^{(i)}, M^{(i)}, S^{(i)})$ from the selected parent and resets only the particle weights. For readability, the algorithm displays weights in probability space; the same updates can be evaluated in log space for numerical stability.

At each step, we compute the effective sample size (ESS) from the normalized particle weights. If the ESS falls below $N/2$, we resample particles according to their weights and reset the weights to $1/N$. The resulting offspring then evolve with independent noise: our method uses the covariance-shaped kernel at these branching steps, while the baseline uses the isotropic kernel. Both methods use the isotropic kernel when no resampling occurs. Each particle also carries its running maximum reward $M^{(i)}$ and accumulated log-potential $S^{(i)}$, which are copied along with its state during resampling. The terminal correction removes the accumulated intermediate potentials and applies the final reward.

Transition kernels. The isotropic baseline uses the same isotropic transition for both kernels. Our sampler replaces the branching kernel with the covariance-shaped transition in Sec. 4.2 and retains the isotropic kernel otherwise. Offspring receive independent noise, including those copied from the same parent.

D Lanczos Sampler Details

In this section, we describe in detail the Lanczos sampler used to build covariance-shaped noise for the stochastic transitions in Sec. 4.2.

Lanczos noise sampling. The stochastic transition in Eq. 12 requires computing $\Sigma_{1|t}(x_t)^{1/2}\epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$. We approximate this quantity using the Lanczos sampler (Akhtar et al., 2026), which targets the same matrix square-root–vector product and requires only covariance–vector products. For a fixed state x_t and time $t \in (0,1)$, write $\Sigma = \Sigma_{1|t}(x_t)$.

Starting from $q_1 = \epsilon/\|\epsilon\|$, Lanczos successively applies Σ and orthogonalizes the resulting vectors to construct an orthonormal basis $Q_m = [q_1, \ldots, q_m]$ of the Krylov subspace

$$\mathcal{K}_m(\Sigma, \epsilon) = \mathrm{span}\{\epsilon, \Sigma\epsilon, \ldots, \Sigma^{m-1}\epsilon\}.$$

The projected matrix

$$T_m = Q_m^\top \Sigma Q_m \in \mathbb{R}^{m\times m}$$

is symmetric positive semidefinite in exact arithmetic. The approximation is

$$z_m = \|\epsilon\| Q_m T_m^{1/2} e_1 \approx \Sigma^{1/2}\epsilon, \tag{23}$$

where $e_1 \in \mathbb{R}^m$ is the first coordinate vector. We compute the small matrix square root by eigendecomposition: if $T_m = U_m \Lambda_m U_m^\top$, then $T_m^{1/2} = U_m \Lambda_m^{1/2} U_m^\top$. If the Krylov subspace stops growing before m iterations, the construction uses the attained dimension. The resulting stochastic increment is $g_t\sqrt{h}\, z_m$.

Algorithm 1 SMC reward alignment with ESS-based resampling.

Require: Prior p_0, reward r, endpoint predictor $m_{1|t}$, alignment strength β, particle count N
Require: Time grid $0 = t_0 < \cdots < t_T = 1$; isotropic and covariance-shaped kernels $K^{\text{iso}}_{t,h}$ and $K^{\text{cov}}_{t,h}$

Draw $x^{(i)} \sim p_0$ independently for $i = 1, \ldots, N$
Initialize $w^{(i)} \leftarrow 1/N$, $M^{(i)} \leftarrow -\infty$, and $S^{(i)} \leftarrow 0$ for all i
for each consecutive pair $(t, t+h)$ in the time grid **do**
    ESS $\leftarrow \left(\sum_{i=1}^{N} (w^{(i)})^2\right)^{-1}$
    $K_{t,h} \leftarrow K^{\text{iso}}_{t,h}$
    if ESS $\leq N/2$ **then**
        Resample N tuples $(x^{(i)}, M^{(i)}, S^{(i)})$ with replacement using probabilities $w^{(i)}$
        $w^{(i)} \leftarrow 1/N$ for all i
        For our method, set $K_{t,h} \leftarrow K^{\text{cov}}_{t,h}$ ▷ Baseline retains $K^{\text{iso}}_{t,h}$
    end if
    for $i = 1, \ldots, N$ **do**
        Draw $x^{(i)} \sim K_{t,h}(x^{(i)}, \cdot)$ using independent noise
        if $t + h < 1$ **then**
            $M^{(i)} \leftarrow \max\{M^{(i)}, r(m_{1|t+h}(x^{(i)}))\}$
            $\ell \leftarrow \beta M^{(i)}$
        else
            $\ell \leftarrow \beta r(x^{(i)}) - S^{(i)}$ ▷ Terminal correction
        end if
        $w^{(i)} \leftarrow w^{(i)} \exp(\ell)$
        $S^{(i)} \leftarrow S^{(i)} + \ell$
    end for
    Normalize $w^{(i)} \leftarrow w^{(i)} / \sum_{j=1}^{N} w^{(j)}$ for all i
end for
return $\{(x^{(i)}, w^{(i)})\}_{i=1}^{N}$ and the particle with the highest reward

Each iteration requires one covariance–vector product. By Eq. 6, for the exact marginal velocity,

$$\Sigma_{1|t}(x_t)v = \frac{(1-t)^2}{t}\left[v + (1-t)J_{x_t}u_t(x_t)v\right]. \tag{24}$$

In practice, we substitute the learned velocity u_θ and, under the Jacobian symmetry approximation described in App. B, evaluate the Jacobian–vector product using

$$J_{x_t}u_\theta(x_t, t)^\top v = \nabla_{x_t}\langle u_\theta(x_t, t), v\rangle,$$

with v held fixed during differentiation. Reusing the velocity computation graph therefore requires one additional VJP per iteration. We use $m = 2$ by default and examine this choice in the ablation below.

The ideal Lanczos construction assumes a symmetric positive semidefinite covariance operator. With a learned velocity and finitely many iterations, the resulting noise is an approximation to the target Gaussian perturbation.

Table 6: Aesthetic reward and runtime for different Lanczos iteration counts m.

m	Aesthetic (↑)	Runtime (s)
1	6.542	43.79
2	6.614	44.25
3	6.622	46.13
4	6.606	47.85
5	6.613	49.05

Lanczos iteration ablation. We vary the number of Lanczos iterations $m \in \{1, 2, 3, 4, 5\}$ and measure aesthetic reward and wall-clock time to assess the efficiency of our default choice, $m = 2$. Reward performance is similar for $m \geq 2$, while runtime increases monotonically with m. This suggests that two iterations capture enough covariance structure for effective reward-guided search in our setting, with additional iterations offering limited practical benefit. These results support $m = 2$ as an

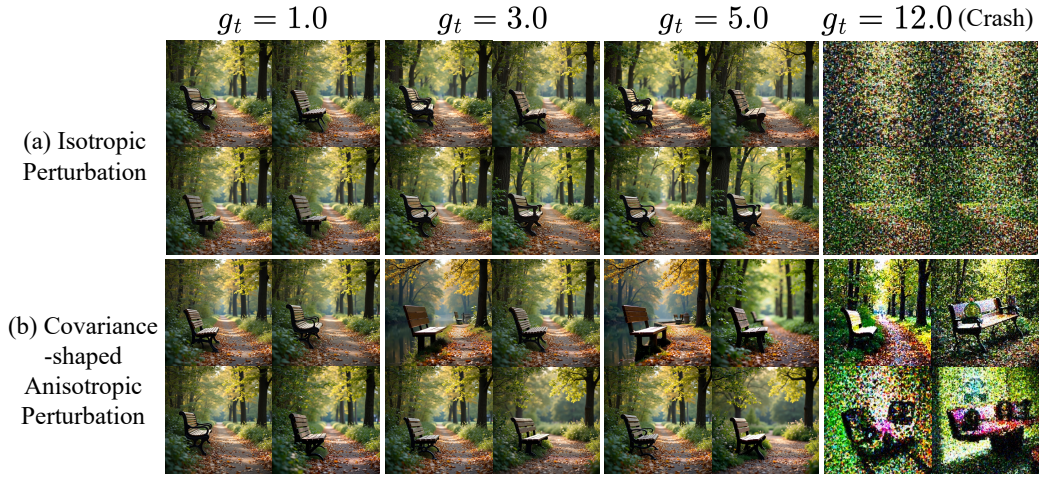


Figure 10: **Effect of noise scale on branching diversity.** Samples are generated from a shared intermediate state using one stochastic transition with varying g_t, followed by deterministic ODE sampling. Covariance-shaped transitions produce greater visual variation at moderate noise scales, while excessive noise degrades sample quality for both methods.

efficient default, although comparable rewards do not establish convergence of the Lanczos approximation itself.

For m=1 in Tab. 6 , the Lanczos approximation reduces to $z_1 = \sqrt{q_1^\top \Sigma_{1|t}(x_t) q_1}\,\epsilon$, where $q_1 = \epsilon/\|\epsilon\|$. Thus, it only rescales the initial Gaussian draw without changing its direction. Additional iterations introduce Krylov directions that allow richer covariance shaping. The lower reward observed at $m = 1$ is consistent with this limited flexibility, suggesting that additional directions are useful for effective exploration in our setting.

E Implementation Details

E.1 Endpoint diversity visualization

Details of Figure 2. We use FLUX.1-dev with the prompt "a photo of a bench." Images are generated at a resolution of 512×512 using 100 sampling steps and a guidance scale of 3.5. Starting from a single initial noise sample, we integrate the deterministic flow ODE using the Euler solver to obtain a fixed intermediate state x_t at $t = 0.05$. We then branch from this shared state by applying one Euler–Maruyama step with either isotropic or covariance-shaped noise, using independent Gaussian draws across branches. All subsequent steps use the deterministic Euler solver, so the variation among candidates within each method originates from this single stochastic transition.

For this qualitative comparison, the stochastic increments are

$$\delta_h^{\text{iso}} = g_t^{\text{iso}} \sqrt{h}\,\epsilon, \qquad \delta_h^{\text{cov}} = g_t^{\text{cov}} \sqrt{h}\,\Sigma_{1|t}(x_t)^{1/2}\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I),$$

where h is the step size. We set $g_t^{\text{iso}} = 5.0$ and choose g_t^{cov} heuristically at this fixed state to obtain a comparable perturbation magnitude. This scale choice is specific to the state and branching step used in the visualization and does not enforce exact matching of the expected squared perturbation norms.

Scaling the noise schedule. Simply increasing g_t can increase endpoint variance, but may also increase discretization error at a fixed step size. For isotropic noise, the Euler–Maruyama update is

$$x_{t+h} = x_t + h u_t(x_t) + \underbrace{\frac{h g_t^2}{2} \nabla \log p_t(x_t)}_{\text{drift correction}} + \underbrace{g_t \sqrt{h}\,\epsilon}_{\text{stochastic perturbation}}, \qquad \epsilon \sim \mathcal{N}(0, I).$$

Increasing g_t amplifies not only the random perturbation but also the drift correction needed to preserve the marginals. For example, doubling g_t doubles the noise amplitude while quadrupling the drift correction. Euler–Maruyama evaluates this correction only at the beginning of the step, so a larger displacement can make this local approximation less accurate. Consequently, the discretized transition can deviate from the intended marginals even though the continuous-time SDE preserves them.

Maintaining accuracy as g_t increases therefore generally requires a smaller step size. Both the correction and the noise magnitude are controlled by hg_t^2: reducing h proportionally to g_t^{-2} keeps their scales unchanged. However, this also keeps the leading-order endpoint variance

$$\widetilde{\mathcal{V}}_h(x_t) = hg_t^2 \|J_m(x_t)\|_F^2$$

unchanged at the same state. Thus, increasing the noise scale alone faces a trade-off between endpoint diversity and numerical accuracy. Covariance shaping aims to improve the endpoint variance obtained from a given perturbation budget by allocating noise to more influential directions.

E.2 Spectral and Endpoint-Variance Measurements

We describe the measurements reported in Fig. 3 and Tab. 1. We use FLUX.1-dev Black Forest Labs (2024) with the default Euler solver and the prompt "a photo of a bench." To make full Jacobian computation and storage tractable on limited resources, we use an image resolution of 128×128, corresponding to a latent dimension of 4096.

To limit the cost of computing full Jacobians, we evaluate only at $t \in \{0.2, 0.5, 0.8\}$ along 50-step denoising trajectories. At each selected timestep, we explicitly compute and store the full endpoint-prediction Jacobian $J_m(x_t)$ for each of 64 initial noise seeds. We use these matrices to obtain the eigenvalue spectra and evaluate the leading-order endpoint variance from Theorem 1,

$$\widetilde{\mathcal{V}}_h(x_t) = hg_t^2 \operatorname{tr}\big(J_m(x_t)A_t(x_t)J_m(x_t)^\top\big).$$

Fig. 3 reports spectra averaged over the 64 seeds at each timestep, while Tab. 1 reports endpoint-variance estimates averaged over both the seeds and the three timesteps.

E.3 Reward Alignment

SMC Hyperparameters. We follow FK-Steer with a normalized ESS threshold of 0.5. Particle weights are updated using intermediate rewards evaluated on posterior-mean endpoint predictions. We select the final particle with the highest optimization reward and evaluate all image metrics on that same particle.

To account for differences in reward scale, we use $\beta = 4.0$ for aesthetic alignment and $\beta = 8.0$ for ImageReward alignment and offline-RL tasks. To allow resampling when $N = 2$, we use a normalized ESS threshold of 0.75 for this setting and 0.5 otherwise.

Text-to-image generation. We use FLUX.1-dev as the backbone and generate images at a resolution of 512×512 using the Euler solver with 100 denoising steps and a guidance scale of 3.5. We use the same random seeds across all methods for a controlled comparison. Fig. 4 reports inference-time scaling results for particle counts $N \in \{2, 3, 4, 5, 6\}$.

Offline RL. We reproduce QGF as the flow-policy backbone, largely following its official implementation. For each environment, we trained the model for 500k steps using Adam with a constant learning rate of 3×10^{-4} and a batch size of 1024. Also following the default QGF setting, we use an implicit Q-learning (IQL) critic (Kostrikov et al., 2022) to estimate the Q-function.

At inference time, we use 20 denoising steps to generate chunks of five future actions. Each action has two dimensions in `pointmaze-giant` and five in `cube-triple`, yielding chunk dimensions of 10 and 25, respectively. Fig. 5 reports inference-time scaling results for particle counts $N \in \{3, 4, 5, 6, 7\}$. The maximum number of execution steps is 500 for `pointmaze-giant` and 1,000 for `cube-triple`.

F MORE QUALITATIVE RESULTS

In this section, we present more qualitative results on text-to-image reward alignment. Fig. 11 shows additional text-to-image samples from BoN, SMC with isotropic transitions, and SMC with our covariance-shaped transitions under the same seeds and compute budget. Our method yields higher aesthetic scores in (a) and better adherence to the object counts, spatial relations, and attributes specified in the GenEval prompts in (b).



(a) Aesthetic alignment.



(b) Text–image alignment.

Figure 11: **Qualitative comparisons** for text-to-image generation. Our method produces images that better satisfy the respective objectives.