Test-time Scaling of Diffusions with Flow Maps

Amirmojtaba Sabour[*,1,2,3] **Michael S. Albergo**[*,4,5,6] **Carles Domingo-Enrich**[*,7]

Nicholas M. Boffi[8] **Sanja Fidler**[1,2,3] **Karsten Kreis**[†,1] **Eric Vanden-Eijnden**[†,9,10]

[1] NVIDIA [2] University of Toronto [3] Vector Institute [4] Harvard University [5] Kempner Institute [6] IAIFI [7] Microsoft Research [8] Carnegie Mellon University [9] Courant Institute, New York University [10] Capital Fund Management (CFM)

Project Page: https://flow-map-trajectory-tilting.github.io/

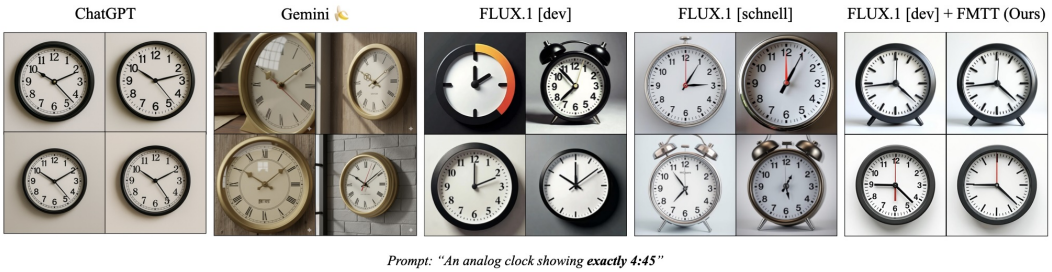


Figure 1: Flow Map Trajectory Tilting (FMTT) implements efficient and accurate test-time search with flow map look-ahead that can overcome model biases and reliably sample from regions of the distribution (e.g., precise clock times) that baselines fail to capture.

Abstract

A common recipe to improve diffusion models at test-time so that samples score highly against a user-specified reward is to introduce the gradient of the reward into the dynamics of the diffusion itself. This procedure is often ill posed, as user-specified rewards are usually only well defined on the data distribution at the end of generation. While common workarounds to this problem are to use an approximate denoiser to estimate what a sample would have been at the end of generation, we propose a simple solution to this problem by working directly with a flow map. By exploiting a relationship between the flow map and velocity field governing the instantaneous transport, we construct an algorithm, Flow Map Trajectory Tilting (FMTT), which leverages a fast yet precise flow map look-ahead and provably performs better ascent on the reward than standard test-time methods involving the gradient of the reward. The approach can be used to either perform exact sampling via importance weighting or principled search that identifies local maximizers of the reward-tilted distribution. We demonstrate the efficacy of our approach against other look-ahead techniques, and show how the flow map enables engagement with complicated reward functions that make possible new forms of image editing, e.g. by interfacing with vision language models.

[*]Equal contribution
[†]Equal supervision

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

1 Introduction

Large-scale foundation models built from diffusions [18, 34] or flow-based transport [1, 2, 24, 25] have become highly successful tools across computer vision and scientific domains. In this paradigm, generation amounts to numerically solving an ordinary or stochastic differential equation (ODE/SDE) whose coefficients are learned neural networks. An active area of research is how to best *adapt* these dynamics at inference time to produce samples that align with a user-specified reward. For example, as shown in Figure 1, a user may want an image of a clock displaying a precise time, which is often generated inaccurately without suitable adaptation. These approaches, often referred to as *guidance*, require no additional re-training and are therefore orthogonal to fine-tuning methods, which modify the model itself through an additional learning stage to modify the quality of generated samples.

While guidance-based approaches can often be made to work well in practice, most methods are somewhat *ad-hoc*, and proceed by postulating a term that may drive the generative equations towards the desired goal. To this end, a common approach is to incorporate the gradient of the reward model, which imposes a gradient ascent-like structure on the reward throughout generation. Despite the intuitive appeal of this approach, typical rewards are defined only at the terminal point of generation – i.e., over a clean image – rather than over the entire generative process. This creates a need to "predict" where the current trajectory will land, in principle necessitating an expensive additional differential equation solve per step of generation. To avoid the associated computational expense of this nested solve, common practice is to employ a heuristic approximation of the terminal point, such as leveraging an approximate, often blurry, one-step denoiser estimate that can be derived from a learned score- or flow-based model. See Figure 2 for visualizations.

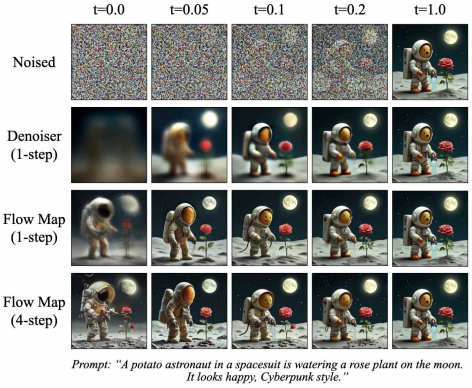


Figure 2: Comparing look-aheads across noise levels t: corrupted inputs, 1-step denoiser, 1-step flow map, and 4-step flow map.

In this paper, we revisit reward guidance through flow maps, a recent approach to flow-based generative modeling that directly learns the solution operator of a probability flow ODE rather than its drift [7, 8, 15, 30]. By leveraging a simple identity of the flow map, we show that an implicit flow can be used to define a reward-guided generative process as in the case of standard flow-based models. With access to the flow map in addition to the implicit flow, we can accurately predict the terminal point of a trajectory using a few function evaluations, vastly improving the approximate prediction relative to denoiser-based techniques – see again Figure 2 – and leading to significantly improved optimization of the reward. In addition, we highlight how to incorporate time-dependent weights throughout the generative process to account for the gradient ascent's failure to equilibrate on the timescale of generation, leading to several new and effective ways to sample high-reward outputs.

Contributions. (*i*) We introduce Flow Map Trajectory Tilting (FMTT), a principled inference-time adaptation procedure for flow maps that uses their efficient yet accurate look-ahead capabilities to incorporate complex rewards into diffusion guidance and search algorithms. (*ii*) Using properties of the flow map, we derive a simple expression for the importance weights in our Jarzynski/SMC scheme, and prove that this scheme is optimal in a simplified Gaussian setting with linear rewards. (*iii*) We show that FMTT has better test-time scaling characteristics than standard ways of embedding rewards into diffusion sampling, and can effectively leverage natural-language rewards specified via pretrained VLMs. On these complex rewards, existing test-time search baselines often offer little to no improvement over naive best-of-N, while flow maps make search reliably effective, significantly improve performance, and solve cases for which prior methods fail.

2 Related Work

Flows and diffusions. Diffusion [18, 34] and flow models [1, 24, 25] are the backbone of efficient, state-of-the-art generative model for continuous data. They are learned by regressing the coefficients that appear in ordinary or stochastic differential equations that fulfill the transport of samples from one distribution to samples from another. Dual to the instantaneous picture of transport is the flow

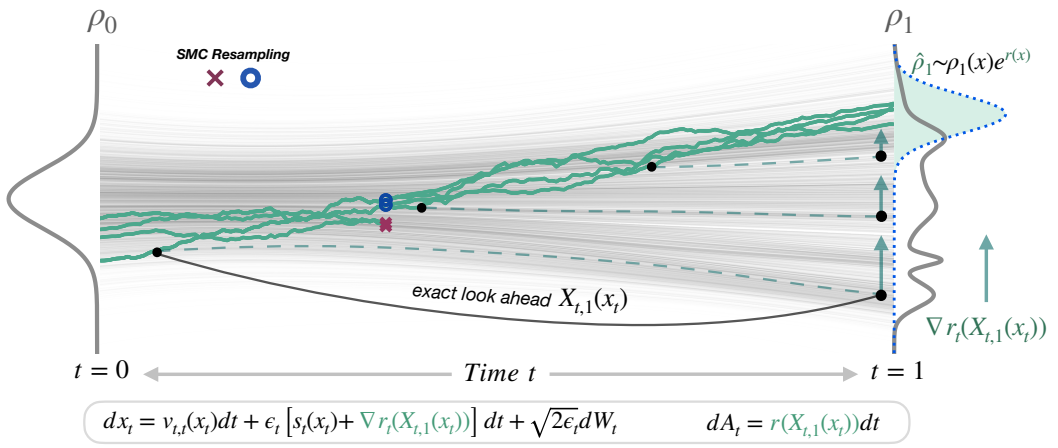


Figure 3: **Overview of Flow Map Trajectory Tilting**. Using the look-ahead map $X_{t,1}(x_t)$ in the diffusion inside the reward, reward information can be principly used through the tilted trajectories (green lines). This allows us to perform better ascent on the reward, and the importance weights A_t take on a remarkably simple form that can be used for both exactly sampling $\hat{\rho}_t$ and search for maximizers of $\hat{\rho}_t$.

map [7, 8, 15, 20, 30, 35], in which we learn not the coefficients in a differential equation that needs to be integrated, but the arbitrary integrator itself. This enables few-step sampling. Our approach in this paper is to combine these perspectives to modify diffusions using the flow map.

Test-time scaling for diffusions. Test-time scaling in diffusions refers to the line of work that trades off compute at inference time to improve the performance of a model or align the generation with a user-specified reward [26]. Certain works use the approximate denoiser associated with the score model to perform look-ahead on the dynamics [21, 31, 40, 45] or do not use any look-ahead at all [27, 32]. However, as discussed later, there is little signal from the denoiser at early times in the generative trajectory. Other works rely on Monte Carlo search algorithms [23, 29], which monotonically increase the reward but reduce sample diversity. As we will see, many of these approaches are compatible with the flow map approach presented here.

Posterior sampling. Reward alignment can also be viewed through the lens of posterior sampling. Introducing a binary optimality variable $\mathcal{O}$ with $p(\mathcal{O}{=}1 \mid x_1) \propto e^{r(x_1)}$, the tilted density $\hat{\rho}_1$ is precisely the posterior $p(x_1 \mid \mathcal{O}{=}1)$, which is the perspective taken by DAS [21]. This makes test-time scaling closely related to solving inverse problems with diffusions [5, 11, 33, 43, 44], where the likelihood is similarly only defined on clean samples and must be approximated along the generative trajectory. DPS [11], for example, evaluates the likelihood at the one-step denoiser estimate of the clean sample, which for a reward-based likelihood recovers the denoiser look-ahead discussed in Section 3 and hence faces the same issue of a weak gradient signal at high noise values. Such methods that evaluate can instead use the flow map to make that prediction; DAPS [44], which obtains its endpoint from a short diffusion rollout, is a natural example. Other approaches rely on structure specific to inverse problems, such as a known measurement operator, a pseudoinverse [33], or a learned likelihood surrogate [5, 43], and do not directly extend to arbitrary rewards.

3 Methodology

We consider the task of generative modeling via continuous-time flow maps, wherein samples $x_0 \in \mathbb{R}^d$ from a base distribution with probability density function (PDF) ρ_0 are mapped via a diffeomorphism to samples x_1 from the target PDF ρ_1 known through empirical data. From there, we will detail how the instantaneous dynamics of this map can be directly adapted (without retraining) to sample a **tilted distribution** favoring a reward, i.e. to sample $\hat{\rho}_1(x) = \rho_1 e^{r(x)+\hat{F}}$ where $r(x)$ is a user specified reward function and $\hat{F} = -\ln \int_{\mathbb{R}^d} \rho_1(x) e^{r(x)} dx$ is a normalization factor.

3.1 Background on Dynamical Generative Modeling

An effective means of instantiating the transport from the base PDF ρ_0 to the target PDF ρ_1 relies on formulating it as the solution to an ordinary differential equation (ODE) $\dot{x}_t = b_t(x_t),\ x_{t=0} \sim \rho_{t=0}$ (1), where $b_t : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ is a velocity field that governs the transport, adjusted so that the solutions to the ODE (1) satisfy $x_{t=1} \sim \rho_1$. As the time-dependent PDF $\rho_t(x)$

of the solutions to (1) at time t satisfies the continuity equation $\partial_t \rho_t = -\nabla \cdot (b_t \rho_t),\ \rho_{t=0} = \rho_0$ (2),this requirement on b_t implies that the solution to (2) is such that $\rho_{t=1} = \rho_1$.

Associated with these dynamics is the two-time **flow map** $X_{s,t} : [0,1]^2 \times \mathbb{R}^d \to \mathbb{R}^d$, which satisfies $X_{s,t}(x_s) = x_t,\ \forall s,t \in [0,1]$ (3). That is, the map jumps along solutions of (1) from time s to time t. Notably, if $s = 0$ and $t = 1$, we could produce a sample under ρ_1 in a single step, though we have the freedom to use more if we so choose. This property, and the relation between the flow map and b_t will be exploited below to devise a principled adaptation procedure for $X_{s,t}$. Importantly, the flow map satisfies the Eulerian equation $\partial_s X_{s,t}(x) + b_s(x) \cdot \nabla X_{s,t}(x) = 0$ (4),which will play a role in simplifying our analysis later. Equation (4) can be obtained by taking the total derivative of (3) with respect to s, using the ODE (1), and evaluating the result at $x_s = x$.

Stochastic interpolants. One way to instantiate the generative models above is to construct a PDF ρ_t that connects ρ_0 to ρ_1 and then learn the associated velocity field b_t that gives rise to this evolution. A common strategy to construct such a path and regress b_t is that of stochastic interpolants [1, 2, 24, 25], in which ρ_t is defined as the law of the stochastic process $I_t(x_0, x_1) = \alpha_t x_0 + \beta_t x_1$ with $(x_0, x_1) \sim \rho(x_0, x_1)$, where $\rho(x_0, x_1)$ is some coupling from which x_0, x_1 are drawn that marginalizes onto ρ_0, ρ_1, and α_t, β_t are scalar coefficients that satisfy $\alpha_0 = \beta_1 = 1$ and $\alpha_1 = \beta_0 = 0$. A common choice is to use $\alpha_t = 1 - t$ and $\beta_t = t$, which we will use throughout this work for simplicity. Importantly, using these coefficients, the law of this process satisfies (2) with velocity field $b_t(x) = \mathbb{E}[\dot{I}_t | I_t = x] = \mathbb{E}[x_1 | I_t] - \mathbb{E}[x_0 | I_t]$ (5), where we used $\dot{I}_t = x_1 - x_0$ and $\mathbb{E}[\cdot | I_t = x]$ denotes expectation over $\rho(x_0, x_1)$ conditional on $I_t = x$. By Stein's identity, the score is given by $s_t(x) = \nabla \log \rho_t(x) = -\frac{1}{1-t}\mathbb{E}[x_0 | I_t = x]$, and using $x = \mathbb{E}[I_t | I_t = x] = (1-t)\mathbb{E}[x_0 | I_t] + t\mathbb{E}[x_1 | I_t]$, it can be expressed in terms of b_t as $s_t(x) = (t b_t(x) - x)(1-t)^{-1}$ (6).

The velocity field $b_t(x)$ is also the minimizer of a simple quadratic objective [1, 24] which, once learned, can be translated into a function for the score via (6). Using the score, the deterministic ODE can be converted to a stochastic dynamics $dx_t = [b_t(x_t) + \epsilon_t s_t(x_t)]dt + \sqrt{2\epsilon_t} dW_t$ (7), where $\epsilon_t \geq 0$ is an arbitrarily tunable diffusion coefficient and dW_t is an incremental Brownian motion [2]. The solutions to (7) sample the same PDF ρ_t as (1), as can be seen from the fact that the PDF of (7) satisfies the Fokker-Planck equation $\partial_t \rho_t = -\nabla \cdot (b_t \rho_t) + \epsilon_t \nabla \cdot [-s_t \rho_t + \nabla \rho_t]$ (8), which reduces to (2) since $s_t \rho_t = \nabla \rho_t$. The velocity field given in (5) is related to the two-time flow map $X_{s,t}$ via the tangent identity [7, 20] $\lim_{s \to t} \partial_t X_{s,t}(x) = b_t(x)$ (9), which says that for infinitesimally small steps, the variation of the flow map output in time is characterized by the velocity. To automatically enforce the boundary condition $X_{s,s}(x) = x$, we can express the flow map as $X_{s,t}(x) = x + (t-s) v_{s,t}(x)$ (10), where $v_{s,t}(x) : [0,1]^2 \times \mathbb{R}^d \to \mathbb{R}^d$ is a function of s, t, and x defined through this relation. Importantly, *the velocity field is accessible directly from the flow map by using* (9) *on* (10)*:* $v_{t,t}(x) = b_t(x)$ (11). As such, training a flow map model instead of just a velocity gives access to both the drifts in (1) and (7) as well as the any-step model, i.e. the ability to *look ahead* our trajectory.

3.2 Fixing Inference-time Adaptation of Diffusions with Flow Maps

A contemporary question is how best to adapt the SDE (7) to tilt toward samples that score highly against a time-dependent reward $r_t(x)$ satisfying $r_0{=}0$ and $r_{t=1}{=}r$ so that the time-dependent tilted PDF $\hat{\rho}_t{=}\rho_t e^{r_t(x)+\hat{F}_t}$ satisfies $\hat{\rho}_{t=0}{=}\rho_0$ and $\hat{\rho}_{t=1}{=}\hat{\rho}_1{=}\rho_1 e^{r+\hat{F}}$. One may want to *sample exactly* under the tilted PDF $\hat{\rho}_t{=}\rho_t e^{r_t(x)+F_t}$ for scientific applications, or one may want to *find local maximizers* of $\hat{\rho}_t$, such as in image generation to ensure user prompts align with image content.

Tilting the diffusion. A natural way to modify the SDE (7) is to add the gradient of the time-dependent reward r_t to the score, i.e. use

$$d\tilde{x}_t = [b_t(\tilde{x}_t) + \epsilon_t s_t(\tilde{x}_t) + \epsilon_t \nabla r_t(\tilde{x}_t)]dt + \sqrt{2\epsilon_t} dW_t, \quad \tilde{x}_0 \sim \rho_0, \tag{12}$$

assuming $r_0(x) = 0$ to avoid changing the prior ρ_0. However, in practice this faces an issue:

A meaningful $r_t(x)$ is not readily available, as user-specified rewards are usually learned only on the data-distribution, i.e. at time $t = 1$.

One could think of several solutions to this problem: **Naïve (no) look-ahead:** This amounts to using e.g. $r_t(x){=}t \cdot r(x)$. Unfortunately, the gradient dynamics from $t\nabla r(x)$ provides no clear signal at small times when $\tilde{x}_t$ is still far from the region where the reward $r(x)$ is meaningful. **Denoiser look-ahead:** A common workaround for the fact that the reward has no signal for most of the trajectory is to use the *denoiser* $\mathsf{D}_t(x){=}\mathbb{E}[x_1 | I_t{=}x]$ to estimate where the sample would have gone. That is,



Figure 4: Qualitative results using VLM-based rewards. Prompts where the base model fails to generate aligned outputs are corrected by FMTT. Flow map look-ahead produces the most reliable improvements, whereas denoiser look-ahead often remains insufficient to achieve precise prompt following. For additional qualitative results, see Section D.3.2.

instead of $r_t(x)=t\cdot r(x)$, we could instead use $r_t(x)=t\cdot r(\mathsf{D}_t(x))$. This strategy is tractable because the denoiser is readily available from the score. However, this still does not provide useful information early on in the dynamics, as the denoiser is only effective at producing samples close to the data distribution later in the evolution (Figure 2). **Flow map look-ahead:** Intuitively, the above dynamics are better if one works instead with the flow map defined above. Because the flow map allows us to look ahead at any point on the trajectory, e.g. by taking $\tilde{x}_t$ at time t and computing $X_{t,1}(\tilde{x}_t)$, and because the velocity field associated to the flow map is accessible via $\lim_{s\to t}\partial_t X_{s,t}(x) = v_{t,t}(x)$ (from which we can recover the score function $s_t(x)$), we can instead sample with the following SDE:

$$d\tilde{x}_t = v_{t,t}(\tilde{x}_t)\,dt + \epsilon_t\Big[s_t(\tilde{x}_t) + t\nabla_{\tilde{x}_t} r\big(X_{t,1}(\tilde{x}_t)\big)\Big]\,dt + \sqrt{2\epsilon_t}\,dW_t, \quad \tilde{x}_0 \sim \rho_0. \tag{13}$$

where $r_t(x) = t\cdot r(X_{t,1}(x))$ makes use of the *exact look-ahead* to properly evaluate the reward for any t. The above could be interpreted as a mixture of denoising and ascent on the reward.

3.3 Correcting the Dynamics for Unbiased Sampling

While using the flow map composed with the reward makes possible the precise use of the reward for all times in the diffusion trajectory, in applications where it is important to *exactly* sample the tilted distribution $\hat{\rho}_t$, the dynamics in (13) are not sufficient for this. This leads to the second issue, for any version of $r_t(x)$, with or without flow map:

The PDF associated with (12) is not the tilted density $\hat{\rho}_t$.

To see why this is true, note that we can explicitly compare the PDF $\tilde{\rho}_t$ of $\tilde{x}_t$ to that of $\hat{\rho}_t$. Indeed, the PDF $\tilde{\rho}_t(x)$ of $\tilde{x}_t$ satisfies:

$$\partial_t\tilde{\rho}_t + \nabla\cdot(b_t\tilde{\rho}_t) = \epsilon_t\nabla\cdot([-s_t - \nabla r_t]\tilde{\rho}_t + \nabla\tilde{\rho}_t), \tag{14}$$

and we can show explicitly that $\hat{\rho}_t$ satisfies a different, imbalanced equation which can be obtained by expanding $\partial_t\hat{\rho}_t + \nabla\cdot(b_t\hat{\rho}_t)$:

$$\begin{aligned}\partial_t\hat{\rho}_t + \nabla\cdot(b_t\hat{\rho}_t) &= \partial_t(e^{r_t+\hat{F}_t}\rho_t) + \nabla\cdot(b_t e^{r_t+\hat{F}_t}\rho_t)\\ &= (\partial_t r_t + \partial_t\hat{F}_t)\hat{\rho}_t + b_t\cdot\nabla r_t\hat{\rho}_t,\end{aligned} \tag{15}$$

where we used the FPE (8) with $\epsilon_t = 0$ to get the last equality. Since $\nabla\hat{\rho}_t = (s_t + \nabla r_t)\hat{\rho}_t$ we can add the diffusion term $\epsilon_t\nabla\cdot(-(s_t+\nabla r_t)\hat{\rho}_t + \nabla\hat{\rho}_t) = 0$ to (15) to arrive at

$$\partial_t\hat{\rho}_t + \nabla\cdot(b_t\hat{\rho}_t) = \epsilon_t\nabla\cdot((-s_t - \nabla r_t)\hat{\rho}_t + \nabla\hat{\rho}_t) + (b_t\cdot\nabla r_t + \partial_t r_t + \partial_t\hat{F}_t)\hat{\rho}_t. \tag{16}$$

As we see, the extra term $(b_t \cdot \nabla r_t + \partial_t r_t + \partial_t \hat{F}_t)\hat{\rho}_t$ on the RHS of this equation differentiates it from being the law of (12). Nonetheless, we can account for this term with weights A_t emerging as the solution of a different differential equation coming from an adaptation of the Jarzynski equality [19, 37]:

Proposition 3.1 (Jarzynski's estimator). *Assume that $r_0 = 0$ so that $\rho_0^r = \rho_0$. Let $\tilde{x}_t$ solve the SDE (12) with $\tilde{x}_0 \sim \rho_0^r$ and define*

$$A_t = \int_0^t \left(b_s(\tilde{x}_s) \cdot \nabla r_s(\tilde{x}_s) + \partial_s r_s(\tilde{x}_s)\right) ds, \tag{17}$$

Then for all $t \in [0,1]$ and any test function $h : \mathbb{R}^d \to \mathbb{R}$, we have

$$\int_{\mathbb{R}^d} h(x)\hat{\rho}_t(x)dx = \frac{\mathbb{E}[e^{A_t} h(\tilde{x}_t)]}{\mathbb{E}[e^{A_t}]}, \tag{18}$$

where the expectations at the right-hand side are taken over the law of $\tilde{x} = (\tilde{x}_t)_{t\in[0,T]}$.

The proof of this statement in Section A.1 relies on manipulating the augmented FPE of the joint PDF $f_t(x, a)$ of $(\tilde{x}_t, A_t)$. This relation ensures that the lag associated to naively using the gradient of the reward in the diffusion can be compensated for by reweighting the trajectories, and, in addition, these weights can be used to perform resampling of the trajectories as is done in Sequential Monte Carlo (SMC) and birth/death processes, as is depicted in Figure 3. Here, as the trajectories walk out, the walkers can be *resampled* using the importance weights, removing some and duplicating others.

Simplicity of importance weights with the flow map. Interestingly, the importance weights in (17) take on a remarkably simple form when using a flow map look-ahead, as stated in the following:

Proposition 3.2 (Unbiased Flow Map Trajectory Tilting). *Using the same notations as in Proposition 3.1, if $r_t(x) = t \cdot r(X_{t,1}(x))$, then the importance weights defined in (17) reduce to*

$$A_t = \int_0^t r(X_{s,1}(\tilde{x}_s))ds. \tag{19}$$

This result is proven in Section A.2, and relies on a simple modification of the proof of Proposition 3.1 and the Eulerian equation (4). Thanks to the flow map, the complicated derivatives appearing in (17) reduce to simply compounding the reward over the look-ahead trajectory.

Reward-modified drift. A variant of Proposition 3.1 makes use of not only augmenting the score with the gradient of the reward with the look-ahead, but also the drift itself. Because $v_{t,t}$ is the velocity field of a stochastic interpolant and is related to the score via (6), we can replace the SDE given in (13) with

$$d\tilde{x}_t = \left[v_{t,t}(\tilde{x}_t) + \chi_t \nabla r_t(\tilde{x}_t)\right] dt + \epsilon_t \left[s_t(\tilde{x}_t) + \nabla r_t(\tilde{x}_t)\right] dt + \sqrt{2\epsilon_t} dW_t, \quad \tilde{x}_0 \sim \rho_0, \tag{20}$$

where $(\chi_t)_{t\in[0,1]}$ is arbitrary, and the log-weight ODE with

$$\frac{d\tilde{A}_t}{dt} = b_t(\tilde{x}_t) \cdot \nabla r_t(\tilde{x}_t) + \partial_t r_t(\tilde{x}_t) + \chi_t\big(\|\nabla r_t(\tilde{x}_t)\|^2 + \Delta r_t(\tilde{x}_t) + \langle \nabla r_t(\tilde{x}_t), s_t(\tilde{x}_t)\rangle\big), \tag{21}$$

where $\tilde{A}_0 = 0$. It is proven in Appendix A.3 that these equations also provide an unbiased sampler of the tilted distribution by ensuring (18) holds for any choice of $(\chi_t)_{t\in[0,1]}$, and three choices besides the default $\chi_t \equiv 0$ are studied (Section D.2). This allows further augmenting the sampling process toward the tilt. We use $\chi_t = \lambda \cdot (1-t)/(t)$ in our experiments, where λ is a hyperparameter controlling the strength of the additional tilt. We found $\lambda = 0.1$ to work well in most cases.

From sampling to search: making the most of rewards in practice. Notably, the importance weights defined in either Equations (19) and (21) do not need to be used to perform exact sampling. They can also be used to perform various greedy search algorithms that **search** for samples with high reward. That is, we are free to use top-n sampling approaches in place of the resampling one would usually do in SMC. Unlike Ma et al. [26], this use of top-n still makes use of the gradient of the reward along the trajectory, while also making use of the better signal thanks to the flow map. A full pseudocode can be seen in Algorithm 1.

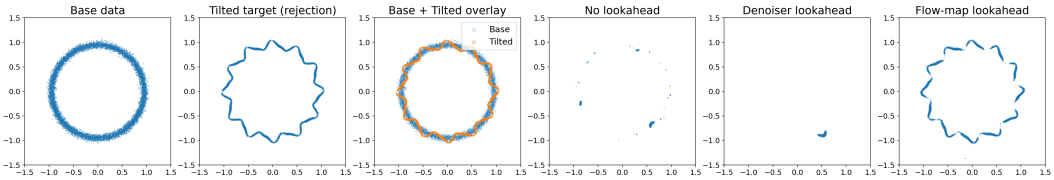


Figure 5: Tilting a 2D toy distribution. We train a flow model on the base ring and define a reward that tilts it toward a zigzag target. With 500-step SMC, no look-ahead and denoiser look-ahead fail to recover the target, while flow map look-ahead captures it effectively. See Section D.1 for details.

3.4 Optimality of Flow Map Look-Ahead

In Figure 5, we visualize the SMC output using different look-aheads on a 2D toy dataset. We train a flow matching model on a ring-shaped distribution and define a reward that tilts this base distribution toward a zigzag-shaped target. Running the SMC algorithm in Equations (13) and (17) with 500 steps and 10000 particles, we find that both naive and denoiser look-aheads require over 150 resampling steps, which degenerates the particle system and produces collapsed blob-like outputs. In contrast, flow map look-ahead uses only 6 resampling steps and accurately captures the tilted target. This difference stems from the large variance of the SMC importance weights when using the unsuccessful look-aheads. For more details, please see Section D.1.

To better understand this effect, we analyze a simple setting where the data distribution is an isotropic Gaussian with standard deviation c and the reward is linear. The following theorem shows that the flow map is the optimal linear look-ahead that minimizes the variance of the incremental log-weights A_t.

Theorem 3.3 (Proof in Section A.6)**.** *Let $\rho_{data}(x) = \mathcal{N}(0, c^2 I)$ be the data distribution, and let $r(x) = u^\top x$ be a linear reward for some u. For the default SMC correction specified by Equations (13) and (17), the flow map is the optimal linear look-ahead that minimizes the variance of the incremental log-weights A_t.*

Although the theorem is proved in a simplified, analytically tractable setting, together with the 2D results it suggests that flow map look-ahead yields more stable SMC behavior than prior look-aheads. Directly measuring this stability is difficult: while SMC gives an unbiased estimate $\hat{Z}_{\mathrm{SMC}}$ of $\mathbb{E}[e^{A_t}]$, its variance can be exponentially large, hard to estimate, and sensitive to choices such as particle count, resampling, and schedule. To address this, in Section B we relate $\mathrm{Var}[\hat{Z}_{\mathrm{SMC}}]$ to the *thermodynamic length*, a simpler proxy computable from the A_t updates. In Section D.2, class-conditional MNIST experiments show that flow map look-ahead again gives the most stable SMC behavior, achieving the lowest thermodynamic length. These theoretical analyses motivate our novel flow map look-ahead framework as a principled and broadly relevant technique applicable also in other diffusion model search and reward guidance methods [21, 31, 32]. This underscores the general impact of FMTT.

4 Experiments

Next, we evaluate our approach on text-to-image generation using a 4-step distilled flow map following Align Your Flow [30], which is trained by distilling the open-source FLUX.1 [dev] model [6]. The flow map is trained in roughly 4 hours on 8 NVIDIA A100 GPUs. For these experiments, $\mathrm{Sampling} = \mathrm{False}, \mathrm{Searching} = \mathrm{True}$ is used in Algorithm 1. We consider three categories of reward functions: 1) *VLM-based rewards* defined through natural language queries, 2) *Geometric rewards* enforcing structural constraints such as symmetry or layout control, 3) *Human preference rewards* capturing visual quality and text alignment. Note that for these experiments, unlike the experiments discussed in Section 3.4 which perform exact sampling, we use the search version of FMTT.

We compare against several *gradient-free* and *gradient-based* baselines. Gradient-free approaches such as Best-of-N [9], Multi-Best-of-N [23], and beam search [14] rely on sampling and selection, and remain confined to the base model's distribution. Gradient-based methods use reward gradients, but differ in how they apply them: ReNO [13] performs gradient ascent in the initial noise latent space, keeping samples tied to the base distribution, whereas our FMTT algorithm (and its ablations) and DAS [21] use the gradient to modify the generative process itself, enabling exploration beyond the model's support and generation of out-of-distribution (OOD) samples. Notably, the gradient of the reward used in FMTT efficiently gives meaningful signal for the whole trajectory thanks to the flow map, enabling OOD sample generation for highly nuanced rewards.

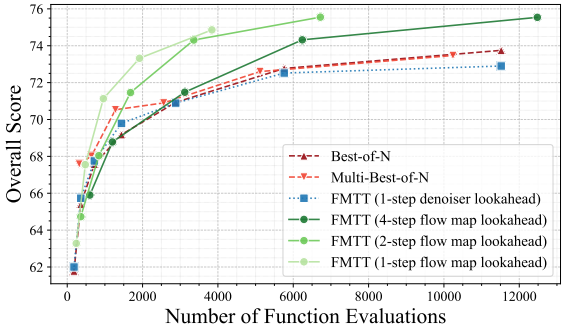


Figure 6: Scaling results on UniGenBench++. Overall score versus compute, measured in network evaluations. FMTT significantly outperforms both Best-of-N and Multi-Best-of-N, achieving higher scores at comparable compute. FMTT with a 1-step denoiser look-ahead fails to beat Best-of-N, highlighting the ineffectiveness of reward gradients at blurry denoised states. See Section D.3.4 for details.

4.1 VLMs as a Judge

In this section, we explore using pretrained vision language models (VLMs) to judge our images. The setup is as follows: we provide the generated image along with a binary yes/no question, and define the reward to be $\mathrm{sigmoid}(\mathrm{logits}[\mathrm{Yes}] - \mathrm{logits}[\mathrm{No}])$. This formulation allows rewards to be expressed entirely in natural language. Additionally, since some VLMs accept multiple image inputs, we can also define rewards that depend on comparisons between the generated image and additional context images. In our experiments, we use Skywork-VL Reward [38] for single-image rewards and Qwen2.5-VL-7B-Instruct [4] for multi-image cases (see Section D.3.2 for details and examples).

To quantitatively benchmark FMTT, we evaluate our method on the UniGenBench++ [39] benchmark using the Skywork-VL reward model. This benchmark contains 600 short English prompts, each requiring 4 generated images. The resulting images are scored by the UniGenBench evaluation model, a fine-tuned variant of Qwen2.5-VL-72B [4], across multiple dimensions including entity layout, text rendering, world knowledge, and more. We ask the following question of the Skywork-VL reward VLM: *'Is {PROMPT} a correct caption for the image? Please answer no if the image is not in high definition (i.e., clear, sharp, not pixelated, and not blurry)."*

We summarize the results as a performance-vs-compute scaling plot in Figure 6. We also provide a summary of all the metrics in Table 1 for the primary methods and the DAS baseline. The plot shows that, when comparing the flow map look-ahead, the 1-step denoiser look-ahead, and standard best-of-N sampling, the denoiser look-ahead offers little to no improvement over the best-of-N baseline. In contrast, FMTT with a flow map look-ahead consistently improves the performance at lower computational cost. We attribute the denoiser's weak performance to its heavily blurred predictions at early, high-noise timesteps, which fall far off the data manifold and result in unhelpful reward gradients. See Figure 2 for side-by-side comparison of the different look-ahead strategies across different noise levels. Additionally, the strong performance obtained with a 1-step flow map look-ahead suggests that FMTT is not overly sensitive to the quality of the pretrained flow map, and is effective even with suboptimal or partially-trained flow maps that may be even cheaper to obtain. For a full performance breakdown including all metrics at all compute budgets, please see Table 4. Comparing the two DAS runs, we further find that flow map look-ahead improves prior test-time scaling methods as well.

Finally, in Section D.3.4, we also evaluate the distilled FLUX.1 [Schnell] model, both with Best-of-N and as an alternative look-ahead. Schnell + Best-of-N achieves the best overall performance in this setting, likely because Schnell is distilled using latent adversarial distillation and trades off diversity for sample quality by concentrating on high-probability modes. However, as a gradient-free method, Best-of-N remains limited to the model's original distribution and can fail on harder rewards that require generating samples outside these modes, such as Figures 1 and 7. Meanwhile, using Schnell as a look-ahead improves over denoiser look-ahead within FMTT, supporting our main claim that stronger look-aheads lead to more effective test-time trajectory tilting.

Table 1: Summary of key quantitative results on UniGenBench++. Entries show the percentage of images judged by the evaluation model to satisfy each criterion. We use 4 particles for the base model and flowmap, 256 particles for best-of-N and FMTT, and 64 particles for DAS. See the extended Table 4 for the complete results.

Method	Mean	Style	World Knowledge	Attribute	Action	Relationship	Compound	Grammar	Logical Reasoning	Entity Layout	Text Generation	NFE
FLUX.1 [dev]	61.78	85.20	86.23	64.53	60.46	63.07	41.24	60.96	24.08	67.16	30.17	180
Flow Map	58.16	79.70	80.22	59.51	57.22	62.44	42.01	57.62	25.92	70.90	3.74	16
FLUX.1 [dev] + Best-of-N	73.75	91.10	**92.88**	76.39	71.58	75.51	61.86	69.52	45.18	80.97	44.83	11520
FMTT - 1-step denoiser look-ahead	72.90	91.00	90.51	73.93	70.72	78.30	60.31	69.65	42.89	81.53	39.94	11520
FMTT - 1-step flow map look-ahead	74.86	**91.90**	92.72	77.78	71.86	77.41	62.50	72.59	46.33	**81.72**	46.55	3840
FMTT - 2-step flow map look-ahead	**75.55**	**91.90**	**92.88**	**78.21**	**72.34**	**79.19**	**65.08**	**72.73**	**46.79**	81.53	**47.70**	6720
DAS - 1-step denoiser look-ahead	66.42	86.90	87.66	68.38	64.35	69.29	52.32	63.90	33.72	72.01	32.76	2880
DAS - 1-step flow map look-ahead	68.14	88.50	89.72	70.09	66.25	70.05	54.16	63.77	33.26	73.51	34.77	1920



Prompt: "A miniature forest with tall pine trees, a glowing campfire, and fireflies drifting in the night sky, ***all inside a keyhole on a black background****"*

Figure 7: Qualitative comparison on masked rewards. Only our flow map-based FMTT reliably satisfies the constraints, only generating content in the unmasked regions. The base model (prompt only), and denoiser look-ahead fail to capture the constraint. For additional examples, see Section D.3.1.

Separately from the UniGenBench++ evaluation, we include additional qualitative examples in Figure 4. The figure highlights two prompts where the base model fails to produce text-aligned outputs. In contrast, using the VLM-based reward, our FMTT algorithm generates outputs that match the input text much more accurately. Figure 8 shows further examples on prompts that describe uncommon scenes, where the base model falls back to the more common interpretation of the prompt and FMTT recovers the intended one. For additional VLM-based experiments, including multi-image settings, please see Section D.3.2. One potential downside of FMTT is the possibility of reward hacking, as the search procedure explicitly maximizes the VLM reward which can contain unwanted artifacts. To mitigate this, we have observed that the yes/no questions must be written with sufficient detail to prevent the algorithm from exploiting loopholes [3]. Further discussion and examples in Section D.3.3.

4.2 Human Preference Rewards

Additionally, we quantitatively evaluate FMTT on GenEval [16]. This benchmark consists of ≈ 550 object-centric prompts and measures the quality of generated images using a pre-trained object detector. For the reward model, we follow prior work [13] and use a linear combination of PickScore [22], HPSv2 [41], ImageReward [42], and CLIPScore [28]. The results are presented in Table 2.

The base model, FLUX.1 [dev], achieves strong scores due to its training on large amounts of object-centric data. Distillation into a 4-step flow map slightly reduces performance but significantly accelerates generation. A simple best-of-N search on top of the flow map recovers this drop and surpasses the base diffusion model while remaining about 30% faster. More advanced search methods, such as multi-best-of-N or beam search, yield additional but modest gains. Using reward gradients with FMTT provides a further (small) improvement. Finally, we also ablate the use of the 4-step flow map look-ahead by comparing FMTT against variants using either a 1-step denoiser or a 4-step diffusion sampler. The flow map look-ahead consistently performs best, in line with our earlier findings.

We also note that FLUX.1 [dev] has already been post-trained with human preference data. This makes it harder for test-time optimization on similar rewards to substantially shift the output distribution further, which likely explains why the gains across the different methods are limited. Nevertheless, FMTT achieves the best overall performance, outperforming all other approaches by clear margin.

Table 2: Quantitative GenEval results. NFE is total number of function evaluations during generation.

Method	Mean ↑	Single Obj. ↑	Two Obj. ↑	Counting ↑	Colors ↑	Position ↑	Attr. Binding ↑	NFE
Diffusions + Flow Maps								
FLUX.1 [dev]	0.65	0.99	0.78	0.70	0.78	0.18	0.45	180
Flow Map	0.62	0.99	0.72	0.63	0.80	0.19	0.39	16
Gradient-Free Search								
FLUX.1 [dev] + Best-of-N	0.75	0.99	0.94	0.83	0.86	0.26	0.57	1440
Flow Map + Best-of-N	0.73	1.00	0.88	0.82	0.85	0.25	0.59	128
Flow Map + Multi-best-of-N	0.76	1.00	0.95	0.84	0.85	0.26	0.69	1280
Flow Map + Beam Search	0.75	1.00	0.92	0.86	0.85	0.29	0.58	1200
Gradient-Based Search								
Flow Map + ReNO	0.71	0.98	0.89	0.79	0.89	0.20	0.57	1280
FMTT (Ours)	**0.79**	**1.0**	**0.97**	**0.90**	**0.91**	**0.30**	**0.64**	1400
FMTT - 1-step denoiser look-ahead	0.75	0.99	0.90	0.87	0.87	0.26	0.59	350
FMTT - 4-step diffusion look-ahead	0.75	0.99	0.93	0.86	0.89	0.27	0.57	1400

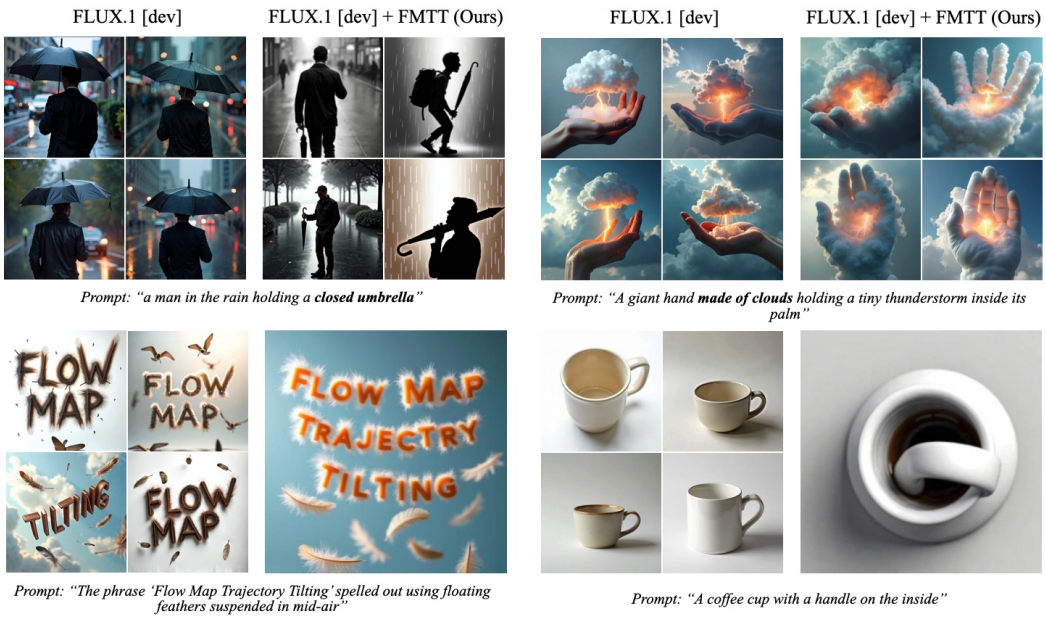


Figure 8: Additional qualitative results using VLM-based rewards on hard prompts. The base model defaults to the common interpretation of the prompt and ignores the unusual requirement highlighted in bold, whereas FMTT satisfies it while remaining realistic.

4.3 Geometric Transformation Rewards

While human preference and VLM-based rewards are sufficient for many high-level objectives, they are often a poor fit when the goal requires fine, pixel-level control. A simple example is enforcing invariance under a geometric transform via $r(x) = -d(x, T(x))$, where $T(x) : \mathbb{R}^d \to \mathbb{R}^d$ is a transformation and $d(\cdot, \cdot)$ is a distance metric. For example, if T is a masking operator, maximizing this reward encourages the masked region to be black, which can be used to control where content is allowed to appear. Similar objectives can be defined for symmetry, anti-symmetry, rotation, and more. These rewards are highly specialized and are typically satisfied only in the far tails of the base model's output distribution, making them a challenging testbed for test-time scaling methods.

Figure 13 shows that the base model roughly follows these objectives but does not fully satisfy them (e.g., small planets break symmetry, the cat's eyes are not anti-symmetric). Prior methods such as multi-best-of-N [23] and ReNO [13] also struggle, either violating the constraints or producing blurry images. In contrast, our gradient-based variants directly modify the dynamics, producing sharper outputs that reliably satisfy the constraints.

For an even harder case, we use a masked-generation reward in Figure 7. FMTT with a denoiser look-ahead increases the reward by darkening the masked region, but it does not consistently move all content into the unmasked area. Using a flow map look-ahead, however, successfully maximizes the reward and generates images that fully satisfy the constraint.

5 Conclusions and Future Work

We presented FMTT, a gradient-based test-time scaling framework for diffusion models that uses an efficient flow map for look-ahead prior to reward evaluation. FMTT consistently improves performance across benchmarks and enables sampling from long-tailed regions where the base model and prior test-time search methods often fail, thereby demonstrating state-of-the-art inference-time guidance capabilities. We also showed compatibility with diverse reward types, including human preference models, geometric transformation objectives, and VLM-based question-answering. Promising directions for future work include integrating FMTT with online RL-based diffusion/flow map fine-tuning literature, applying these ideas to even more complex rewards, and scaling to domains such as image editing, video generation, world models, and scientific applications.

References

[1] Michael S. Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *International Conference on Learning Representations*, 2023.

[2] Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 2025.

[3] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016.

[4] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[5] Sattwik Basu, Chaitanya Amballa, Zhongweiyang Xu, Jorge Vančo Sampedro, Srihari Nelakuditi, and Romit Roy Choudhury. Contrastive diffusion guidance for spatial inverse problems. In *International Conference on Learning Representations*, 2026.

[6] Black Forest Labs. FLUX. https://github.com/black-forest-labs/flux, 2024.

[7] Nicholas M. Boffi, Michael S. Albergo, and Eric Vanden-Eijnden. How to build a consistency model: Learning flow maps via self-distillation. In *Advances in Neural Information Processing Systems*, 2025.

[8] Nicholas M. Boffi, Michael S. Albergo, and Eric Vanden-Eijnden. Flow map matching with stochastic interpolants: A mathematical framework for consistency models. *Transactions on Machine Learning Research*, 2025.

[9] Sourav Chatterjee and Persi Diaconis. The sample size required in importance sampling. *The Annals of Applied Probability*, 2018.

[10] Nicolas Chopin, Sumeetpal S. Singh, Tomás Soto, and Matti Vihola. On resampling schemes for particle filters with weakly informative observations. *The Annals of Statistics*, 2022.

[11] Hyungjin Chung, Jeongsol Kim, Michael T. McCann, Marc L. Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations*, 2023.

[12] Chenguang Dai, Jeremy Heng, Pierre E. Jacob, and Nick Whiteley. An invitation to sequential Monte Carlo samplers. *Journal of the American Statistical Association*, 2020.

[13] Luca Eyring, Shyamgopal Karthik, Karsten Roth, Alexey Dosovitskiy, and Zeynep Akata. ReNO: Enhancing one-step text-to-image models through reward-based noise optimization. In *Advances in Neural Information Processing Systems*, 2024.

[14] Guilherme Fernandes, Vasco Ramos, Regev Cohen, Idan Szpektor, and João Magalhães. Latent beam diffusion models for decoding image sequences. *arXiv preprint arXiv:2503.20429*, 2025.

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

[16] Dhruba Ghosh, Hanna Hajishirzi, and Ludwig Schmidt. GenEval: An object-focused framework for evaluating text-to-image alignment. In *Advances in Neural Information Processing Systems*, 2023.

[17] Roger B. Grosse, Chris J. Maddison, and Russ R. Salakhutdinov. Annealing between distributions by averaging moments. In *Advances in Neural Information Processing Systems*, 2013.

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

[19] Christopher Jarzynski. Nonequilibrium equality for free energy differences. *Physical Review Letters*, 1997.

[20] Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ODE trajectory of diffusion. In *International Conference on Learning Representations*, 2024.

[21] Sunwoo Kim, Minkyu Kim, and Dongmin Park. Test-time alignment of diffusion models without reward over-optimization. In *International Conference on Learning Representations*, 2025.

[22] 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 *Advances in Neural Information Processing Systems*, 2023.

[23] Gyubin Lee, Bao N. Nguyen Truong, Jaesik Yoon, Dongwoo Lee, Minsu Kim, Yoshua Bengio, and Sungjin Ahn. Adaptive inference-time scaling via cyclic diffusion search. In *Advances in Neural Information Processing Systems*, 2025.

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

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

[26] Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, et al. Scaling inference time compute for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[27] Alireza Mousavi-Hosseini, Stephen Y. Zhang, Michal Klein, and Marco Cuturi. Flow matching with semidiscrete couplings. In *International Conference on Learning Representations*, 2026.

[28] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

[29] Vignav Ramesh and Morteza Mardani. Test-time scaling of diffusion models via noise trajectory search. In *Advances in Neural Information Processing Systems*, 2025.

[30] Amirmojtaba Sabour, Sanja Fidler, and Karsten Kreis. Align your flow: Scaling continuous-time flow map distillation. In *Advances in Neural Information Processing Systems*, 2025.

[31] 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 *International Conference on Machine Learning*, 2025.

[32] Marta Skreta, Tara Akhound-Sadegh, Viktor Ohanesian, Roberto Bondesan, Alán Aspuru-Guzik, Arnaud Doucet, Rob Brekelmans, Alexander Tong, and Kirill Neklyudov. Feynman-Kac correctors in diffusion: Annealing, guidance, and product of experts. In *International Conference on Machine Learning*, 2025.

[33] Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*, 2023.

[34] 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.

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

[36] Saifuddin Syed, Alexandre Bouchard-Côté, Kevin Chern, and Arnaud Doucet. Optimised annealed sequential Monte Carlo samplers. *Journal of the Royal Statistical Society, Series B*, 2026.

[37] Suriyanarayanan Vaikuntanathan and Christopher Jarzynski. Escorted free energy simulations: Improving convergence by reducing dissipation. *Physical Review Letters*, 2008.

[38] Xiaokun Wang, Peiyu Wang, Jiangbo Pei, Wei Shen, Yi Peng, Yunzhuo Hao, Weijie Qiu, Ai Jian, Tianyidan Xie, Xuchen Song, et al. Skywork-VL reward: An effective reward model for multimodal understanding and reasoning. *arXiv preprint arXiv:2505.07263*, 2025.

[39] Yibin Wang, Zhimin Li, Yuhang Zang, Jiazi Bu, Yujie Zhou, Yi Xin, Junjun He, Chunyu Wang, Qinglin Lu, Cheng Jin, et al. UniGenBench++: A unified semantic evaluation benchmark for text-to-image generation. *arXiv preprint arXiv:2510.18701*, 2025.

[40] Luhuan Wu, Brian Trippe, Christian Naesseth, David Blei, and John P. Cunningham. Practical and asymptotically exact conditional sampling in diffusion models. In *Advances in Neural Information Processing Systems*, 2023.

[41] Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023.

[42] 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 *Advances in Neural Information Processing Systems*, 2023.

[43] Linfeng Ye, Shayan Mohajer Hamidi, Mert Pilanci, and Konstantinos N. Plataniotis. CL-DPS: A contrastive learning approach to blind nonlinear inverse problem solving via diffusion posterior sampling. In *International Conference on Learning Representations*, 2026.

[44] Bingliang Zhang, Wenda Chu, Julius Berner, Chenlin Meng, Anima Anandkumar, and Yang Song. Improving diffusion inverse problem solving with decoupled noise annealing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[45] Tao Zhang, Jia-Shu Pan, Ruiqi Feng, and Tailin Wu. VFScale: Intrinsic reasoning through verifier-free test-time scalable diffusion model. In *International Conference on Learning Representations*, 2026.

Appendix

A Proofs **15**
- A.1 Proof of Proposition 3.1 . . . 15
- A.2 Proof of Proposition 3.2 . . . 16
- A.3 Modifying the Drift with the Flow Map Reward . . . 16
- A.4 Solving the SDE/ODE System Numerically . . . 19
  - A.4.1 On the best-performing approach in practice . . . 21
- A.5 Lemmas used in Section A.3 . . . 21
- A.6 Flow Map Look-ahead is Optimal for Linear Rewards in a Gaussian Setting . . . 23

B Analyzing the Performance of Test-Time Sampling Algorithms **25**
- B.1 Simulating the Dynamics with SMC . . . 25
- B.2 The Incremental and Total Discrepancies . . . 26
- B.3 The Variance of the SMC Normalization Constant . . . 27
- B.4 Optimizing the Annealing Schedule to Minimize the Total Discrepancy . . . 27

C Implementation Details **29**

D Additional Experiments **30**
- D.1 Toy 2D Experiments . . . 30
- D.2 MNIST Experiments . . . 31
- D.3 Text-to-Image Experiments . . . 36
  - D.3.1 Additional Geometric Reward Examples . . . 36
  - D.3.2 Additional VLM-as-judge Examples . . . 37
  - D.3.3 VLM Reward Hacking . . . 37
  - D.3.4 Complete UniGenBench++ Results . . . 38

E Broader Impact **39**

F Limitations **39**

G Licenses **40**

A Proofs

A.1 Proof of Proposition 3.1

Consider the coupled SDE/ODE:

$$\begin{aligned} d\tilde{x}_t &= \big(b_t(\tilde{x}_t) + \epsilon_t s_t(\tilde{x}_t) + \epsilon_t \nabla r_t(\tilde{x}_t)\big)dt + \sqrt{2\epsilon_t} dW_t, \qquad \tilde{x}_0 \sim \rho_0, \\ dA_t &= \big(b_t(\tilde{x}_t) \cdot \nabla r_t(\tilde{x}_t) + \partial_t r_t(\tilde{x}_t)\big)\, dt, \qquad A_0 = 0. \end{aligned} \tag{22}$$

Let $f_t(x, a)$ be the joint probability density of $(\tilde{x}_t, A_t)$. Then, it satisfies the Fokker-Planck equation

$$\begin{aligned} \partial_t f_t(x,a) &= -\nabla_x \cdot \big(\big(b_t(x) + \epsilon_t s_t(x) + \epsilon_t \nabla r_t(x)\big) f_t(x,a)\big) - \partial_a\big(\big(b_t(x)\cdot\nabla r_t(x) + \partial_t r_t(x)\big) f_t(x,a)\big) \\ &\quad + \epsilon_t \Delta_x f_t(x,a), \\ f_0(x,a) &= \rho_0(x)\delta_0(a). \end{aligned} \tag{23}$$

Observe that if we let $\bar{\rho}_t(x) = \int_{\mathbb{R}} f_t(x,a)e^a\, da$, $\bar{\rho}$ satisfies

$$\begin{aligned} \partial_t \bar{\rho}_t(x) &= \int_{\mathbb{R}} \partial_t f_t(x,a) e^a\, da \\ &= \int_{\mathbb{R}} \big(-\nabla_x \cdot \big(\big(b_t(x) + \epsilon_t s_t(x) + \epsilon_t \nabla r_t(x)\big) f_t(x,a)\big) - \big(\big(b_t(x)\cdot\nabla r_t(x) + \partial_t r_t(x)\big)\partial_a f_t(x,a) \\ &\qquad + \epsilon_t \Delta_x f_t(x,a)\big) e^a\, da \\ &= -\nabla_x \cdot \big(\big(b_t(x) + \epsilon_t s_t(x) + \epsilon_t \nabla r_t(x)\big) \int_{\mathbb{R}} f_t(x,a) e^a\, da\big) \\ &\qquad - \big(b_t(x)\cdot\nabla r_t(x) + \partial_t r_t(x)\big) \int_{\mathbb{R}} \partial_a f_t(x,a) e^a\, da + \epsilon_t \Delta_x \bigg(\int_{\mathbb{R}} f_t(x,a) e^a\, da \bigg) \\ &= -\nabla_x \cdot \big(\big(b_t(x) + \epsilon_t s_t(x) + \epsilon_t \nabla r_t(x)\big)\bar{\rho}_t(x)\big) + \big(b_t(x)\cdot\nabla r_t(x) + \partial_t r_t(x)\big)\bar{\rho}_t(x) + \epsilon_t \Delta_x \bar{\rho}_t(x). \end{aligned} \tag{24}$$

where the fourth inequality holds through integration by parts:

$$\int_{\mathbb{R}} \partial_a f_t(x,a) e^a\, da = [f_t(x,a)e^a]_{-\infty}^{\infty} - \int_{\mathbb{R}} f_t(x,a)\partial_a e^a\, da = -\int_{\mathbb{R}} f_t(x,a) e^a\, da = -\bar{\rho}_t(x). \tag{25}$$

We can reinterpret equation (24) as stating that for any test function h,

$$\int_{\mathbb{R}^d} h(x)\bar{\rho}_t(x)\, dx = \int_{\mathbb{R}^d}\int_{\mathbb{R}} f_t(x,a) e^a h(x)\, da\, dx = \mathbb{E}[e^{A_t} h(\tilde{x}_t)]. \tag{26}$$

However, $\bar{\rho}_t$ is not a normalized density for $t \geq 0$: if we integrate both sides of (24) over $\mathbb{R}^d$, and we use the divergence theorem, we obtain that

$$\partial_t \int_{\mathbb{R}^d} \bar{\rho}_t(x)\, dx = \int_{\mathbb{R}^d} \big(b_t(x)\cdot\nabla r_t(x) + \partial_t r_t(x)\big)\bar{\rho}_t(x)\, dx, \qquad \int_{\mathbb{R}^d} \bar{\rho}_0(x)\, dx = 1 \tag{27}$$

If we define $F_t = \int_{\mathbb{R}^d} \bar{\rho}_t(x)\, dx$, and we define $\hat{\rho}_t(x) = \bar{\rho}_t(x)/F_t$, we obtain that

$$\begin{aligned} \partial_t \hat{\rho}_t(x) &= -\nabla_x \cdot \big(\big(b_t(x) + \epsilon_t s_t(x) + \epsilon_t \nabla r_t(x)\big)\frac{\bar{\rho}_t(x)}{F_t}\big) + \big(b_t(x)\cdot\nabla r_t(x) + \partial_t r_t(x)\big)\frac{\bar{\rho}_t(x)}{F_t} + \epsilon_t \Delta_x \frac{\bar{\rho}_t(x)}{F_t} \\ &\quad - \frac{\partial_t F_t}{F_t}\frac{\bar{\rho}_t(x)}{F_t}, \end{aligned} \tag{28}$$

and by (27), if we define $\hat{F}_t = \log F_t$, we have that

$$\partial_t \hat{F}_t = \frac{\partial_t F_t}{F_t} = \int_{\mathbb{R}^d} \big(b_t(x)\cdot\nabla r_t(x) + \partial_t r_t(x)\big)\frac{\bar{\rho}_t(x)}{F_t}\, dx = \int_{\mathbb{R}^d} \big(b_t(x)\cdot\nabla r_t(x) + \partial_t r_t(x)\big)\hat{\rho}_t(x)\, dx. \tag{29}$$

Plugging this into the right-hand side of (28) and substituting $\hat{\rho}_t(x) = \bar{\rho}_t(x)/F_t$ yields a PDE for $\hat{\rho}_t$ which matches (16):

$$\partial_t \hat{\rho}_t(x) = -\nabla_x \cdot \big(\big(b_t(x) + \epsilon_t s_t(x) + \epsilon_t \nabla r_t(x)\big)\hat{\rho}_t(x)\big) + \big(b_t(x) \cdot \nabla r_t(x) + \partial_t r_t(x) - \partial_t \hat{F}_t\big)\hat{\rho}_t(x) + \epsilon_t \Delta_x \hat{\rho}_t(x). \tag{30}$$

To show that equation (18) holds, we rely on (26) and the fact that $F_t = \int_{\mathbb{R}^d} \int_{\mathbb{R}} f_t(x,a) e^a \, da \, dx = \mathbb{E}[e^{A_t}]$:

$$\int_{\mathbb{R}^d} h(x)\hat{\rho}_t(x)\, dx = \frac{1}{F_t} \int_{\mathbb{R}^d} h(x)\bar{\rho}_t(x)\, dx = \frac{\mathbb{E}[e^{A_t} h(\tilde{x}_t)]}{\mathbb{E}[e^{A_t}]}. \tag{31}$$

A.2 Proof of Proposition 3.2

When $r_t(x) = t \cdot r(X_{t,1}(x))$, the log-weight A_t defined in (17) satisfies the ODE

$$\begin{aligned} \frac{\mathrm{d}A_t}{\mathrm{d}t} &= t\, b_t(\tilde{x}_t) \cdot \nabla_{\tilde{x}_t} r(X_{t,1}(\tilde{x}_t)) + \partial_t\big(t\, r(X_{t,1}(\tilde{x}_t))\big) \\ &= t b_t(\tilde{x}_t) \cdot \nabla X_{t,1}(\tilde{x}_t)^\top \nabla r(X_{t,1}(\tilde{x}_t)) + r(X_{t,1}(\tilde{x}_t)) + t\, \partial_t X_{t,1}(\tilde{x}_t) \cdot \nabla r(X_{t,1}(\tilde{x}_t)) \\ &= t \nabla r(X_{t,1}(\tilde{x}_t)) \cdot \big(\partial_t X_{t,1}(\tilde{x}_t) + \nabla X_{t,1}(\tilde{x}_t) b_t(\tilde{x}_t)\big) + r(X_{t,1}(\tilde{x}_t)), \end{aligned} \tag{32}$$

The Eulerian identity states that

$$\partial_t X_{t,1}(x) + \nabla X_{t,1}(x) b_t(x) = 0. \tag{33}$$

To prove (33), we write $0 = \partial_s(X_{s,1} \circ X_{t,s})(x) = \partial_s X_{s,1}(X_{t,s}(x)) + \nabla X_{s,1}(X_{t,s}(x)) \partial_s X_{t,s}(x)$, we use that $\partial_s X_{t,s}(x) = b(X_{t,s}(x))$, and we set $s = t$. Plugging (33) into the right-hand side of (32) yields

$$\frac{\mathrm{d}A_t}{\mathrm{d}t} = r(X_{t,1}(\tilde{x}_t)), \tag{34}$$

which concludes the proof.

A.3 Modifying the Drift with the Flow Map Reward

While the position and log-weight dynamics in Proposition 3.1 and Proposition 3.2 are remarkably simple to compute, they are not the only ones that can be used to sample from the tilted distribution. In particular, if we add an additional term $\chi_t \nabla r_t(\tilde{x}_t)$ to the position SDE, we can adjust the log-weights dynamics so that the coupled system still lets us compute expectations according to the tilted distribution $\hat{\rho}_t$ for all $t \in [0,1]$. Namely, the reward-modified position SDE reads:

$$d\tilde{x}_t = [v_{t,t}(\tilde{x}_t) + \chi_t \nabla r_t(\tilde{x}_t)]\, dt + \epsilon_t\, [s_t(\tilde{x}_t) + \nabla r_t(\tilde{x}_t)]\, dt + \sqrt{2\epsilon_t} dW_t, \quad \tilde{x}_0 \sim \rho_0. \tag{35}$$

There are four choices of χ_t of particular interest:

(i) **Default dynamics**: $\chi_t = 0$. This choice recovers the dynamics introduced in Section 3.3.

(ii) **Tilted score dynamics**: $\chi_t = \eta_t := \alpha_t(\frac{\dot{\beta}_t}{\beta_t}\alpha_t - \dot{\alpha}_t)$. When the dynamics (1) has been learned using stochastic interpolants, the score $s_t(x)$ and the vector field $b_t(x)$, or equivalently $v_{t,t}(x)$, are related to each other via the equation

$$b_t(x) = \alpha_t(\frac{\dot{\beta}_t}{\beta_t}\alpha_t - \dot{\alpha}_t) s_t(x) + \frac{\dot{\beta}_t}{\beta_t} x \tag{36}$$

which (6) is a particular case of, when $\alpha_t = 1 - t$, $\beta_t = t$. The score of the tilted distribution at time t is $\hat{s}_t(x) = \nabla \log \hat{\rho}_t(x) = s_t(x) + \nabla r_t(x)$, and if we replace $s_t(x)$ by $\hat{s}_t(x)$ in (36), we obtain

$$b_t(x) = \alpha_t(\frac{\dot{\beta}_t}{\beta_t}\alpha_t - \dot{\alpha}_t)\hat{s}_t(x) + \frac{\dot{\beta}_t}{\beta_t} x = v_{t,t}(x) + \alpha_t(\frac{\dot{\beta}_t}{\beta_t}\alpha_t - \dot{\alpha}_t)\nabla r_t(x). \tag{37}$$

(iii) **Local tilt dynamics**: $\chi_t = \epsilon_t$. If we let $P(x_{(k+1)h}|x_{kh})$ be the transition kernel for the Euler-Maruyama discretization of the SDE

$$\mathrm{d}\tilde{x}_t = \big(v_{t,t}(\tilde{x}_t) + \epsilon_t s_t(\tilde{x}_t)\big)\mathrm{d}t + \sqrt{2\epsilon_t}\mathrm{d}W_t, \tag{38}$$

the local tilt dynamics arises from tilting $P(x_{(k+1)h}|x_{kh})$ according to the reward $r_{(k+1)h}(x_{(k+1)h})$, which yields the transition kernel $P^r(x_{(k+1)h}|x_{kh}) \propto P(x_{(k+1)h}|x_{kh})\exp(r_{(k+1)h}(x_{(k+1)h}))$. By Lemma A.4, when $h \to 0$, $P^r(x_{(k+1)h}|x_{kh})$ can be viewed as the transition kernel for the SDE

$$d\tilde{x}_t = [v_{t,t}(\tilde{x}_t) + \epsilon_t \nabla r_t(\tilde{x}_t)]\, dt + \epsilon_t\, [s_t(\tilde{x}_t) + \nabla r_t(\tilde{x}_t)]\, dt + \sqrt{2\epsilon_t} dW_t, \tag{39}$$

which amounts to setting $\chi_t = \epsilon_t$ in (20).

(iv) **Base dynamics**: $\chi_t = -\epsilon_t$. With this choice, the SDE (20) becomes

$$d\tilde{x}_t = [v_{t,t}(\tilde{x}_t) + \epsilon_t s_t(\tilde{x}_t)]\, dt + \sqrt{2\epsilon_t} dW_t, \tag{40}$$

which does not involve the reward r_t.

The following proposition, which generalizes Proposition 3.1, shows the log-weight dynamics that we must use for generic reward multipliers χ_t, written in three different ways. Observe that

Proposition A.1 (Unbiased Map Tilting with reward-modified vector field)**.** *Let $(\chi_t)_{t\in[0,1]}$ be an arbitrary real valued function, and let r_t be an arbitrary time-dependent reward. Let $(\tilde{x}_t, A_t)$ be a solution of*

$$d\tilde{x}_t = \big(b_t(\tilde{x}_t) + \chi_t \nabla r_t(\tilde{x}_t) + \epsilon_t s_t(\tilde{x}_t) + \epsilon_t \nabla r_t(\tilde{x}_t)\big)dt + \sqrt{2\epsilon_t} dW_t, \qquad \tilde{x}_0 \sim \rho_0, \tag{41}$$

$$dA_t = \big(b_t(\tilde{x}_t)\cdot\nabla r_t(\tilde{x}_t) + \partial_t r_t(\tilde{x}_t) + \chi_t\big(\|\nabla r_t(\tilde{x}_t)\|^2 + \Delta r_t(\tilde{x}_t) + \langle \nabla r_t(\tilde{x}_t), s_t(\tilde{x}_t)\rangle\big)\big)\, dt, \ A_0 = 0. \tag{42}$$

Then for all $t \in [0,1]$ and any test function $h : \mathbb{R}^d \to \mathbb{R}$, we have

$$\int_{\mathbb{R}^d} h(x)\hat{\rho}_t(x)dx = \frac{\mathbb{E}[e^{A_t} h(\tilde{x}_t)]}{\mathbb{E}[e^{A_t}]}, \tag{43}$$

where the expectations at the right-hand side are taken over the law of $(\tilde{x}, A)$. The equation (42) for A_t can be rewritten as

$$\begin{aligned} dA_t = &\big(b_t(\tilde{x}_t)\cdot\nabla r_t(\tilde{x}_t) + \partial_t r_t(\tilde{x}_t) + \chi_t\big(\|\nabla r_t(\tilde{x}_t)\|^2 + \langle \nabla r_t(\tilde{x}_t), s_t(\tilde{x}_t)\rangle\big)\big)\, dt \\ &+ \frac{\chi_t}{\sqrt{2\epsilon_t}}\nabla r_t(\tilde{x}_t)\cdot \mathrm{d}^- W_t - \frac{\chi_t}{\sqrt{2\epsilon_t}}\nabla r_t(\tilde{x}_t)\cdot \mathrm{d}W_t, \ A_0 = 0. \end{aligned} \tag{44}$$

Equation (42) can be rewritten in a third way as

$$\begin{aligned} \mathrm{d}A_t = &\begin{cases} \frac{\mathrm{d}}{\mathrm{d}s}\mathbb{E}[e^{r_s(X^\chi_s) - r_t(\tilde{x}_t)} \mid X^\chi_t = \tilde{x}_t]\big|_{s=t}\, \mathrm{d}t, & \text{if } \chi_t \geq 0, \\ \big(2\frac{\mathrm{d}}{\mathrm{d}s} r_s(X^0_s)\big|_{\substack{X^0_t = \tilde{x}_t \\ s=t}} - \frac{\mathrm{d}}{\mathrm{d}s}\mathbb{E}[e^{r_s(X^\chi_s) - r_t(\tilde{x}_t)} \mid X^\chi_t = \tilde{x}_t]\big|_{s=t}\big)\, \mathrm{d}t, & \text{if } \chi_t < 0, \end{cases} \\ A_0 = 0, & \end{aligned} \tag{45}$$

where X^χ is a solution of the SDE

$$\mathrm{d}X^\chi_t = \big(b_t(X^\chi_t) + |\chi_t| s_t(X^\chi_t)\big)\mathrm{d}t + \sqrt{2|\chi_t|}\mathrm{d}W_t, \tag{46}$$

and X^0 is a solution of the ODE $\mathrm{d}X^0_t = b_t(X^\chi_t)\,\mathrm{d}t$.

Proof. **(i) Proof of equation** (43). When we replace $b_t(x)$ by $\tilde{b}_t(x) = b_t(x) + \chi_t \nabla r_t(x)$, the analog of equation (15) is:

$$\begin{aligned} \partial_t\hat{\rho}_t + \nabla\cdot(\tilde{b}_t\hat{\rho}_t) &= \partial_t(e^{r_t + \hat{F}_t}\rho_t) + \nabla\cdot(\tilde{b}_t e^{r_t+\hat{F}_t}\rho_t) \\ &= (\partial_t r_t + \partial_t \hat{F}_t)\hat{\rho}_t + e^{r_t+\hat{F}_t}\partial_t\rho_t + \big(b_t + \chi_t\nabla r_t\big)\cdot\nabla r_t\hat{\rho}_t + e^{r_t+\hat{F}_t}\nabla\cdot((b_t + \chi_t\nabla r_t)\rho_t) \\ &= (\partial_t r_t + \partial_t\hat{F}_t)\hat{\rho}_t + b_t\cdot\nabla r_t\hat{\rho}_t + \chi_t\|\nabla r_t\|^2\hat{\rho}_t + \chi_t e^{r_t+\hat{F}_t}\nabla\cdot(\nabla r_t\rho_t) \\ &= \big(b_t\cdot\nabla r_t + \partial_t r_t + \chi_t\big(\|\nabla r_t\|^2 + \Delta r_t + \langle\nabla r_t, s_t\rangle\big) + \partial_t\hat{F}_t\big)\hat{\rho}_t, \end{aligned} \tag{47}$$

where the third equality holds because $\partial_t \rho_t + \nabla \cdot (b_t \rho_t) = 0$ by the FPE (8) with $\epsilon_t \equiv 0$, and in the fourth equality we used the definition $s_t(x) = \nabla \log \rho_t(x)$. Hence, when we replace b_t by $\tilde{b}_t$, the terms $b_t \cdot \nabla r_t + \partial_t r_t$ get replaced by $b_t \cdot \nabla r_t + \partial_t r_t + \chi_t\big(\|\nabla r_t\|^2 + \Delta r_t + \langle \nabla r_t, s_t \rangle\big)$.

Given equation (47), the proof of the statement (43) is analogous to the proof of Proposition 3.1 in Section A.1, simply replacing b_t by $b_t + \chi_t \nabla r_t$, and $b_t \cdot \nabla r_t + \partial_t r_t$ by $b_t \cdot \nabla r_t + \partial_t r_t + \chi_t\big(\|\nabla r_t\|^2 + \Delta r_t + \langle \nabla r_t, s_t \rangle\big)$. We omit a full proof.

(ii) Proof of equation (44). We proceed to prove that (44) is equivalent to (42). Observe that the solution of equation (42) is

$$A_t = \int_0^t \Big(b_t(\tilde{x}_t) \cdot \nabla r_t(\tilde{x}_t) + \partial_t r_t(\tilde{x}_t) + \chi_\tau\big(\|\nabla r_\tau(\tilde{x}_\tau)\|^2 + \Delta r_\tau(\tilde{x}_\tau) + \langle \nabla r_\tau(\tilde{x}_\tau), s_\tau(\tilde{x}_\tau) \rangle\big) \Big)\, dt. \tag{48}$$

Next, we handle the term ∇r_t. Applying Lemma A.5, we obtain that

$$\begin{aligned} &\int_0^t \chi_\tau \Delta r_\tau(\tilde{x}_\tau)\, d\tau = \int_0^t \tau \chi_\tau \Delta (r \circ X_{\tau,1})(\tilde{x}_\tau)\, \mathrm{d}\tau \\ &= \int_t^{t'} \frac{\tau \chi_\tau}{\sqrt{2\epsilon_\tau}} \nabla (r \circ X_{\tau,1})(\tilde{x}_\tau) \cdot \mathrm{d}^- W_\tau - \int_t^{t'} \frac{\tau \chi_\tau}{\sqrt{2\epsilon_\tau}} \nabla (r \circ X_{\tau,1})(\tilde{x}_\tau) \cdot \mathrm{d} W_\tau, \end{aligned} \tag{49}$$

And plugging this back into (48) yields equation (44).

(iii) Proof of equation (45). Finally, we prove that (45) is equivalent to (42). We introduce the necessary concepts first. The semi-group for a (time-inhomogeneous) Markov process is defined as

$$P_{s,t} f(x) = \mathbb{E}\big[f(X_t) \,\big|\, X_s = x \big], \qquad 0 \le s \le t.$$

The (time-dependent) infinitesimal generator is defined by

$$L_t f_t(x) = \lim_{h \downarrow 0} \frac{P_{t,t+h} f_{t+h}(x) - f_t(x)}{h} = \frac{\mathrm{d}}{\mathrm{d}s} \mathbb{E}[f_s(X_s) \,|\, X_t = x]\big|_{s=t}. \tag{50}$$

For the process X^χ that solves (46), the infinitesimal generator takes the form

$$L_t^\chi f_t(x) = \partial_t f_t(x) + (b_t(x) + |\chi_t| s_t(x)) \cdot \nabla f_t(x) + |\chi_t| \Delta f_t(x). \tag{51}$$

We define the Hamilton-Jacobi-Bellman (HJB) residual $\mathcal{R}_t^\chi$ for the reward r_t and the process X^χ as

$$\mathcal{R}_t^\chi(x) = b_t(x) \cdot \nabla r_t(x) + \partial_t r_t(x) + |\chi_t|\big(\|\nabla r_t(x)\|^2 + \Delta r_t(x) + \nabla r_t(x) \cdot s_t(x)\big). \tag{52}$$

The term "HJB residual" stems from the fact that the Hamilton-Jacobi-Bellman equation for a time-dependent reward r_t and process X^χ can be expressed as $\mathcal{R}_t^\chi(x) = 0$ for all t and x[1].

We start with the case $\chi_t \ge 0$. The term in the right-hand side of (42) can be rewritten as follows:

$$\begin{aligned} &b_t(x) \cdot \nabla r_t(x) + \partial_t r_t(x) + \chi_t\big(\|\nabla r_t(x)\|^2 + \Delta r_t(x) + \langle \nabla r_t(x), s_t(x) \rangle\big) = \mathcal{R}_t^\chi(x) \\ &= e^{-r_t(x)} [L_t^\chi e^{r_t}](x) = e^{-r_t(x)} \frac{\mathrm{d}}{\mathrm{d}s} \mathbb{E}[e^{r_s(X_s^\chi)} \,|\, X_t^\chi = x]\big|_{s=t} = \frac{\mathrm{d}}{\mathrm{d}s} \mathbb{E}[e^{r_s(X_s^\chi) - r_t(x)} \,|\, X_t^\chi = x]\big|_{s=t}. \end{aligned} \tag{53}$$

where the second equality holds by Lemma A.6, and the third equality holds by the definition of the infinitesimal generator in (50). This concludes the case $\chi_t \ge 0$.

We move on to the case $\chi_t \ge 0$. The term in the right-hand side of (42) can be rewritten as follows:

$$\begin{aligned} &b_t(x) \cdot \nabla r_t(x) + \partial_t r_t(x) + \chi_t\big(\|\nabla r_t(x)\|^2 + \Delta r_t(x) + \langle \nabla r_t(x), s_t(x) \rangle\big) \\ &= 2\big(b_t(x) \cdot \nabla r_t(x) + \partial_t r_t(x)\big) \\ &\quad - \big(b_t(x) \cdot \nabla r_t(x) + \partial_t r_t(x) + |\chi_t|\big(\|\nabla r_t(x)\|^2 + \Delta r_t(x) + \langle \nabla r_t(x), s_t(x) \rangle\big)\big) \\ &= 2\mathcal{R}_t^0(x) - \mathcal{R}_t^\chi(x) = 2\frac{\mathrm{d}}{\mathrm{d}s}\big(e^{r_s(X_s^0) - r_t(x)}\big)\big|_{\substack{X_t^0 = x \\ s=t}} - \frac{\mathrm{d}}{\mathrm{d}s} \mathbb{E}[e^{r_s(X_s^\chi) - r_t(\tilde{x}_t)} \,|\, X_t^\chi = x]\big|_{s=t} \\ &= 2\frac{\mathrm{d}}{\mathrm{d}s} r_s(X_s^0)\big|_{\substack{X_t^0 = x \\ s=t}} - \frac{\mathrm{d}}{\mathrm{d}s} \mathbb{E}[e^{r_s(X_s^\chi) - r_t(\tilde{x}_t)} \,|\, X_t^\chi = x]\big|_{s=t} \end{aligned} \tag{54}$$

where $\mathcal{R}_t^0$ is defined as $\mathcal{R}_t^\chi$ with $\chi = 0$, and the last equality holds by equation (53) applied to $\mathcal{R}_t^0$ and $\mathcal{R}_t^\chi$. □

[1] Note that in stochastic optimal control, the HJB equation is usually written for the value function $V_t = -r_t$.

The following corollary particularizes Proposition A.1 to the case in which the time-dependent reward is built from the flow map $X_{t,1}$, thus generalizing Proposition 3.2.

Corollary A.2 (Unbiased Map Tilting with reward-modified vector field from flow map)**.** *When* $r_t(x) = t \cdot r(X_{t,1}(x))$*, equations* (42)*,* (44) *and* (45) *simplify respectively to*

$$\mathrm{d}A_t = \Big(\frac{r_t(\tilde{x}_t)}{t} + \chi_t\big(\|\nabla r_t(\tilde{x}_t)\|^2 + \Delta r_t(\tilde{x}_t) + \langle \nabla r_t(\tilde{x}_t), s_t(\tilde{x}_t)\rangle\big)\Big)\,\mathrm{d}t,\ A_0 = 0. \tag{55}$$

$$\begin{aligned}\mathrm{d}A_t = &\Big(\frac{r_t(\tilde{x}_t)}{t} + \chi_t\big(\|\nabla r_t(\tilde{x}_t)\|^2 + \langle \nabla r_t(\tilde{x}_t), s_t(\tilde{x}_t)\rangle\big)\Big)\,\mathrm{d}t \\ &+ \frac{\chi_t}{\sqrt{2\epsilon_t}}\nabla r_t(\tilde{x}_t)\cdot \mathrm{d}^- W_t - \frac{\chi_t}{\sqrt{2\epsilon_t}}\nabla r_t(\tilde{x}_t)\cdot \mathrm{d}W_t,\ A_0 = 0.\end{aligned} \tag{56}$$

$$\mathrm{d}A_t = \begin{cases} \partial_s \mathbb{E}[e^{r_s(X_s^\chi) - r_t(\tilde{x}_t)} \mid X_t^\chi = \tilde{x}_t]\big|_{s=t}\,\mathrm{d}t, & \text{if } \chi_t \geq 0, \\ \big(\frac{2r_t(\tilde{x}_t)}{t} - \partial_s \mathbb{E}[e^{r_s(X_s^\chi) - r_t(\tilde{x}_t)} \mid X_t^\chi = \tilde{x}_t]\big|_{s=t}\big)\,\mathrm{d}t, & \text{if } \chi_t < 0, \end{cases} \qquad A_0 = 0, \tag{57}$$

Proof. Using the argument in Section A.2 yields

$$b_t(x)\cdot\nabla r_t(x) + \partial_t r_t(x) = r(X_{t,1}(x)) = \frac{r_t(x)}{t}. \tag{58}$$

Substituting this into equations (42), (44) and (45) yields the simplified forms. □

Remark A.3 (Setting $\chi_t = 0$ in Proposition A.1 and Corollary A.2). Proposition A.1 and Corollary A.2 show that the default choice $\chi_t = 0$ is the only one for which the evolution of A_t simplifies. Namely, when $\chi_t = 0$, equations (42), (44) and (45) in Proposition A.1 simplify respectively to

$$\mathrm{d}A_t = \big(b_t(\tilde{x}_t)\cdot\nabla r_t(\tilde{x}_t) + \partial_t r_t(\tilde{x}_t)\big)\,\mathrm{d}t, \tag{59}$$

$$\mathrm{d}A_t = \big(b_t(\tilde{x}_t)\cdot\nabla r_t(\tilde{x}_t) + \partial_t r_t(\tilde{x}_t)\big)\,\mathrm{d}t, \tag{60}$$

$$\mathrm{d}A_t = \frac{\mathrm{d}}{\mathrm{d}s} r_s(X_s^0)\big|_{\substack{X_t^0 = \tilde{x}_t \\ s=t}}\,\mathrm{d}t, \tag{61}$$

and equations (55), (56), and (57) in Corollary A.2 become

$$\mathrm{d}A_t = \frac{r_t(\tilde{x}_t)}{t}\,\mathrm{d}t, \tag{62}$$

$$\mathrm{d}A_t = \frac{r_t(\tilde{x}_t)}{t}\,\mathrm{d}t, \tag{63}$$

$$\mathrm{d}A_t = \frac{\mathrm{d}}{\mathrm{d}s} r_s(X_s^0)\big|_{\substack{X_t^0 = \tilde{x}_t \\ s=t}}\,\mathrm{d}t, \tag{64}$$

A.4 Solving the SDE/ODE System Numerically

We need to simulate the coupled dynamics on weights and positions numerically. The SDE (41) for $\tilde{x}_t$ can be solved via a K-step Euler-Maruyama scheme with time discretization $(t_k)_{k=0}^K$ and updates of the form

$$\begin{aligned}X_k = X_{k-1} + (t_k - t_{k-1})[&b_{t_{k-1}}(X_{k-1}) + \chi_{t_{k-1}}\nabla r_{t_{k-1}}(X_{k-1}) \\ &+ \epsilon_{t_{k-1}}(s_{t_{k-1}}(X_{k-1}) + \nabla r_{t_{k-1}}(X_{k-1}))] + \sqrt{2\epsilon_k(t_k - t_{k-1})}\xi_{k-1},\end{aligned} \tag{65}$$

where $\xi_{k-1} \sim N(0, \mathrm{I})$. For generic χ_t, solving the dynamics for the log-weight A_t requires one of the following approaches, which correspond to equations (42), (44) and (45), respectively:

(i) **Estimating the Laplacian** Δr_t: If solving (42), we need to estimate the Laplacian $\Delta r_t(x)$, which can be done using the Hutchinson trace estimator $\widehat{\Delta r_t}(x) = \frac{1}{2M\varepsilon}\sum_{m=1}^M z_m \cdot \big[\nabla r_t(x + \varepsilon z_m) - \nabla r_t(x - \varepsilon z_m)\big]$, where $z_m \sim \mathcal{N}(0, I)$ or Rademacher, $\varepsilon \ll 1$. Then, the log-weights can be simulated with Euler updates of the form:

$$\begin{aligned}A_k = A_{k-1} + (t_k - t_{k-1})\big[&b_{t_{k-1}}(X_{k-1})\cdot\nabla r_{t_{k-1}}(X_{k-1}) + \partial_t r_{t_{k-1}}(X_{k-1}) \\ &+ \chi_{t_{k-1}}\big(\|\nabla r_{t_{k-1}}(X_{k-1})\|^2 + \widehat{\Delta r_{t_{k-1}}}(X_{k-1}) \\ &+ \langle \nabla r_{t_{k-1}}(X_{k-1}), s_{t_{k-1}}(X_{k-1})\rangle\big)\big]\end{aligned} \tag{66}$$

When $\chi_{(k-1)h} = 0$, we do not need to compute the estimate the Laplacian, which makes the update much faster.

(ii) **Estimating the difference of forward and backward Itô integrals**: If solving (44), we estimate

$$\begin{aligned}&\int_{t_{k-1}}^{t_k} \frac{\chi_\tau}{\sqrt{2\epsilon_\tau}} \nabla r_\tau(X_{k-1}) \cdot \mathrm{d}^- W_\tau - \int_{t_{k-1}}^{t_k} \frac{\chi_\tau}{\sqrt{2\epsilon_\tau}} \nabla r_\tau(\tilde{x}_\tau) \cdot \mathrm{d}W_\tau \\ &\approx \chi_{t_k} \sqrt{\frac{t_k - t_{k-1}}{2\epsilon_{t_k}}} \nabla r_{t_k}(X_k) \cdot \xi_{k-1} - \chi_{t_{k-1}} \sqrt{\frac{t_k - t_{k-1}}{2\epsilon_{t_{k-1}}}} \nabla r_{t_{k-1}}(X_{k-1}) \cdot \xi_{k-1}.\end{aligned} \tag{67}$$

Thus, the update reads

$$\begin{aligned}A_k = A_{k-1} + (t_k - t_{k-1}) \Big[& b_{t_{k-1}}(X_{k-1}) \cdot \nabla r_{t_{k-1}}(X_{k-1}) + \partial_t r_{t_{k-1}}(X_{k-1}) \\ & + \chi_{t_{k-1}} \big(\|\nabla r_{t_{k-1}}\|^2 + \nabla r_{t_{k-1}} \cdot s_{t_{k-1}} \big)(X_{k-1}) \Big] \\ & + \chi_{t_k} \sqrt{\frac{t_k - t_{k-1}}{2\epsilon_{t_k}}} \nabla r_{t_k}(X_{k-1}) \cdot \xi^n_{k-1} - \chi_{t_{k-1}} \sqrt{\frac{t_k - t_{k-1}}{2\epsilon_{t_{k-1}}}} \nabla r_{t_{k-1}}(X_{k-1}) \cdot \xi^n_{k-1}.\end{aligned} \tag{68}$$

(iii) **Estimating expectations of** $\exp(r_t)$: If solving (45), we approximate

$$\begin{aligned}&\frac{\mathrm{d}}{\mathrm{d}s} \mathbb{E}[e^{r_s(X^\chi_s) - r_t(x)} \mid X^\chi_t = x]\big|_{s=t} \\ &\approx \frac{1}{M(t_k - t_{k-1})} \sum_{m=1}^{M} \exp\Big(r_{t_k}(x + (t_k - t_{k-1})(b_{t_{k-1}}(x) + |\chi_{t_{k-1}}| s_{t_{k-1}}(x)) \\ &\qquad + \sqrt{2|\chi_{t_{k-1}}|(t_k - t_{k-1})} \xi^{(m)}_{k-1}) - r_{t_{k-1}}(x) \Big), \quad \xi^{(m)}_{k-1} \sim N(0, \mathrm{I}).\end{aligned} \tag{69}$$

where we used a finite-difference approximation of the derivative, and the Euler-Maruyama update corresponding to the SDE (46). Thus, for $\chi_{(k-1)h} \geq 0$ the update reads

$$\begin{aligned}A_k = A_{k-1} + \frac{1}{M} \sum_{m=1}^{M} \exp\Big(& r_{t_k}(X_{k-1} + (t_k - t_{k-1})(b_{t_{k-1}}(X_{k-1}) + |\chi_{t_{k-1}}| s_{t_{k-1}}(X_{k-1})) \\ & + \sqrt{2|\chi_{t_{k-1}}|(t_k - t_{k-1})} \xi^{(m)}_{k-1}) - r_{t_{k-1}}(X_{k-1}) \Big), \quad \xi^{(m)}_{k-1} \sim N(0, \mathrm{I}).\end{aligned} \tag{70}$$

Similarly,

$$\frac{\mathrm{d}}{\mathrm{d}s} r_s(X^0_s)\big|_{\substack{X^0_t = x \\ s=t}} \approx \frac{r_{t_k}(x + (t_k - t_{k-1}) b_{t_k}(x)) - r_{t_k}(x)}{t_k - t_{k-1}}. \tag{71}$$

Thus, for $\chi_{(k-1)h} \leq 0$ the update reads

$$\begin{aligned}A_k = A_{k-1} & + 2\big(r_{t_k}(X_{k-1} + (t_k - t_{k-1}) b_{t_k}(X_{k-1})) - r_{t_{k-1}}(X_{k-1}) \big) \\ & - \frac{1}{M} \sum_{m=1}^{M} \exp\Big(r_{t_k}(X_{k-1} + (t_k - t_{k-1})(b_{t_{k-1}}(X_{k-1}) + |\chi_{t_{k-1}}| s_{t_{k-1}}(X_{k-1})) \\ & \qquad + \sqrt{2\epsilon_k (t_k - t_{k-1})} \xi^n_{k-1}) - r_{t_{k-1}}(X_{k-1}) \Big), \quad \xi^{(m)}_{k-1} \sim N(0, \mathrm{I}).\end{aligned} \tag{72}$$

Observe that for $\chi_{(k-1)h} = 0$, equations (72) and (70) simplify to

$$A_k = A_{k-1} + r_{t_k}(X_{k-1} + (t_k - t_{k-1}) b_{t_k}(X_{k-1})) - r_{t_{k-1}}(X_{k-1}), \tag{73}$$

which makes the update much less expensive.

Observe that following Corollary A.2, when $r_t(x) = t \cdot r(X_{t,1}(x))$ the terms $b_{t_{k-1}}(X_{k-1}) \cdot \nabla r_{t_{k-1}}(X_{k-1}) + \partial_t r_{t_{k-1}}(X_{k-1})$ in (66) and (68) can be replaced by $\frac{r_{t_{k-1}}(X_{k-1})}{t_{k-1}}$.

A.4.1 On the best-performing approach in practice

When $\chi_t \neq 0$, the approach *(i)* requires evaluating the gradient ∇r_t at M points to obtain an estimate with variance $\Theta(1/M)$, and the approach *(iii)* requires evaluating r_t at M points to obtain an estimate with variance $\Theta(1/M)$. Meanwhile, the approach *(ii)* only requires evaluating the gradient ∇r_t at the iterates $(X_k)_{k=0}^K$. Hence, a priori, the approach *(ii)* is preferable.

However, when the reward functions r is a neural network, such as a classifier, a vision-language model, or a CLIP model, the gradients ∇r are very noisy as r is highly non-smooth: if we perturb a sample x with an amount of Gaussian noise small enough that the noiseless sample x and the noised sample $\hat{x}$ have very similar reward values, and compute cosine similarity of the gradients $\nabla r(x)$ and $\nabla r(\hat{x})$, we generally obtain low values ($\ll 0.5$). The noisy gradients cause the difference of integrals in (67) to be poor, as the gradients $\nabla r_{t_k}(X_k)$ and $\nabla r_{t_{k-1}}(X_{k-1})$ are much more misaligned than they would be if the function was smoother. The noisy gradient issue also affect approach *(i)*, but it does not affect approach *(ii)*, which only relies on evaluations of r_t but not its gradient. The issue can be mitigated by replacing the gradient evaluations ∇r_t by averages of the evaluations on noised samples, effectively estimating the gradient of r_t convolved with a Gaussian. However, running approach *(ii)* with this mitigation requires using several evaluations of ∇r_t anyway. In practice, we observe that approach *(iii)* performs best, and that is the approach we used in our MNIST experiments in Section D.2.

A.5 Lemmas used in Section A.3

Lemma A.4 (Simulating the local tilt dynamics)**.** *Consider the Euler-Maruyama discretization of the SDE* $\mathrm{d}\tilde{x}_t = b_t^\epsilon(\tilde{x}_t)\mathrm{d}t + \sqrt{2\epsilon_t}\mathrm{d}W_t$*, where* $b_t^\epsilon(x) = b_t(x) + \epsilon_t s_t(x)$*, which corresponds to the conditional density*

$$P(x_{(k+1)h}|x_{kh}) = \frac{\exp\big(-\|x_{(k+1)h} - x_{kh} - hb_{kh}^\epsilon(x_{kh})\|^2/(4h\epsilon_{kh})\big)}{(4\pi h\epsilon_{kh})^{d/2}}. \tag{74}$$

If we define the conditional density

$$P^r(x_{(k+1)h}|x_{kh}) \propto P(x_{(k+1)h}|x_{kh})\exp(r_{(k+1)h}(x_{(k+1)h})), \tag{75}$$

in the limit $h \to 0$*, this conditional density can be written as*

$$P^r(x_{(k+1)h}|x_{kh}) = \frac{\exp\big(-\|x_{(k+1)h} - x_{kh} - h(b_{kh}^\epsilon(x_{kh}) + 2\epsilon_{kh}\nabla r_{kh}(x_{kh}))\|^2/(4h\epsilon_{kh})\big)}{(4\pi h\epsilon_{kh})^{d/2}}, \tag{76}$$

which is the conditional density for the Euler-Maruyama discretization of the SDE

$$dX_t = (b_t^\epsilon(X_t) + 2\epsilon_t\nabla r_t(X_t))\mathrm{d}t + \sqrt{2\epsilon_t}\mathrm{d}B_t. \tag{77}$$

Proof. We use a discrete version of Itô's lemma:

$$\begin{aligned} r_{(k+1)h}(x_{(k+1)h}) \approx r_{kh}(x_{kh}) + h\big(\partial_t r_{kh}(x_{kh}) + \epsilon_{kh}\Delta r_{kh}(x_k)\big) \\ + \langle\nabla r_{kh}(x_k), x_{(k+1)h} - x_{kh}\rangle + O(h^{3/2}). \end{aligned} \tag{78}$$

Thus,

$$\begin{aligned} &P^r(x_{(k+1)h}|x_{kh}) \\ &\propto \exp\left(-\frac{\|x_{(k+1)h} - x_{kh} - hb_{kh}^\epsilon(x_{kh})\|^2}{4h\epsilon_{kh}} + \langle\nabla r_{kh}(x_k), x_{(k+1)h} - x_{kh}\rangle + O(h^{3/2})\right) \\ &\propto \exp\left(-\frac{\|x_{(k+1)h} - x_{kh} - h(b_{kh}^\epsilon(x_{kh}) + 2\epsilon_{kh}\nabla r_{kh}(x_k))\|^2}{4h\epsilon_{kh}} + O(h^{3/2})\right). \end{aligned} \tag{79}$$

□

Lemma A.5. *Assume that* $(\tilde{x}_t)_{t\in[0,1]}$ *satisfies the SDE* (41)*. We have that*

$$\int_t^{t'} \tau\chi_\tau\Delta(r\circ X_{\tau,1})(\tilde{x}_\tau)\,\mathrm{d}\tau = \int_t^{t'} \frac{\tau\chi_\tau}{\sqrt{2\epsilon_\tau}}\nabla(r\circ X_{\tau,1})(\tilde{x}_\tau)\cdot\mathrm{d}^-W_\tau - \int_t^{t'} \frac{\tau\chi_\tau}{\sqrt{2\epsilon_\tau}}\nabla(r\circ X_{\tau,1})(\tilde{x}_\tau)\cdot\mathrm{d}W_\tau, \tag{80}$$

where the forward and backward Itô integrals are defined respectively as

$$\int_t^{t'} H_\tau \cdot \mathrm{d}W_\tau \;\coloneqq\; \lim_{|\pi|\to 0} \sum_{k=0}^{n-1} H_{t_k} \cdot \big(W_{t_{k+1}} - W_{t_k}\big), \tag{81}$$

$$\int_t^{t'} H_\tau \cdot \mathrm{d}^- W_\tau \;\coloneqq\; \lim_{|\pi|\to 0} \sum_{k=0}^{n-1} H_{t_{k+1}} \cdot \big(W_{t_{k+1}} - W_{t_k}\big). \tag{82}$$

Here, $(H_s)_{t\le s\le t'}$ is a process adapted to the filtration induced by the Brownian motion W such that $\mathbb{E}[\int_t^{t'} \|H_s\|^2\, ds] < +\infty$, $\pi = \{t = t_0 < t_1 < \cdots < t_n = t'\}$ is a partition of $[t, t']$ with mesh $|\pi| = \max_k(t_{k+1} - t_k)$, and the limits are L^2 limits.

Proof. By definition of the forward and backward Itô integrals,

$$\int_0^t H_\tau\, d^- W_\tau \;-\; \int_0^t H_\tau\, dW_\tau = \lim_{|\pi|\to 0} \sum_{k=0}^{n-1} \big(H_{t_{k+1}} - H_{t_k}\big) \cdot \big(W_{t_{k+1}} - W_{t_k}\big) := [H, W]_t, \tag{83}$$

where $[H, W]_t$ is known as the quadratic variation of H and W. Let us set $H_\tau = \frac{\tau\chi_\tau}{\sqrt{2\epsilon_\tau}}\nabla(r \circ X_{\tau,1})(\tilde{x}_\tau) = \gamma_\tau \nabla(r \circ X_{\tau,1})(\tilde{x}_\tau)$, where we defined $\gamma_\tau = \frac{\tau\chi_\tau}{\sqrt{2\epsilon_\tau}}$. By Itô's lemma,

$$\begin{aligned}
&d(\gamma_\tau \cdot \nabla(r \circ X_{\tau,1}))(\tilde{x}_\tau) \\
&= \gamma_t \Big(\partial_t \nabla(r \circ X_{t,1})(\tilde{x}_\tau) + \nabla^2(r \circ X_{t,1})(\tilde{x}_\tau) \cdot \big(\tilde{b}_t + \epsilon_t s_t - \epsilon_t \nabla r_t\big)(\tilde{x}_\tau) + \epsilon_t \nabla \cdot \nabla^2(r \circ X_{t,1})(\tilde{x}_\tau) \Big)\, \mathrm{d}t \\
&\quad + \dot{\gamma}_t \nabla(r \circ X_{t,1})(\tilde{x}_\tau)\, \mathrm{d}t + \sqrt{2\epsilon_t}\gamma_t \nabla^2(r \circ X_{t,1})(\tilde{x}_\tau) dW_t.
\end{aligned} \tag{84}$$

When we simplify the quadratic variation, only the stochastic term survives:

$$\begin{aligned}
[H, W]_t &= \lim_{|\pi|\to 0} \sum_{k=0}^{n-1} \sqrt{2\epsilon_{t_k}}\gamma_{t_k} \langle \nabla^2(r \circ X_{t_k,1})(\tilde{x}_{t_k})(W_{t_{k+1}} - W_{t_k}), W_{t_{k+1}} - W_{t_k} \rangle \\
&= \lim_{|\pi|\to 0} \sum_{k=0}^{n-1} \sqrt{2\epsilon_{t_k}}\gamma_{t_k} \mathrm{Tr}\big(\nabla^2(r \circ X_{t_k,1})(\tilde{x}_{t_k})\big)(t_{k+1} - t_k) = \int_0^t \sqrt{2\epsilon_\tau}\gamma_\tau \Delta(r \circ X_{\tau,1})(\tilde{x}_\tau)\, d\tau \\
&= \int_0^t \tau\chi_\tau \Delta(r \circ X_{\tau,1})(\tilde{x}_\tau)\, d\tau,
\end{aligned} \tag{85}$$

which concludes the proof. □

Lemma A.6. *The infinitesimal generator L_t^χ defined in (51) and the HJB residual defined in (52) fulfill:*

$$[L_t^\chi e^{r_t}](x) = e^{r_t(x)} \mathcal{R}_t^\chi(x). \tag{86}$$

Proof. Applying the Hopf-Cole transformation, we obtain that

$$\begin{aligned}
\partial_t e^{r_t(x)} &= e^{r_t(x)} \partial_t r_t(x) \\
&= e^{r_t(x)} \big(\mathcal{R}_t^\chi(x) - b_t(x) \cdot \nabla r_t(x) - |\chi_t| \big(\|\nabla r_t(x)\|^2 + \Delta r_t(x) + \nabla r_t(x) \cdot s_t(x) \big) \big).
\end{aligned} \tag{87}$$

And observe that

$$\begin{aligned}
&(b_t(x) + |\chi_t| s_t(x)) \cdot \nabla e^{r_t(x)} + |\chi_t| \Delta e^{r_t(x)} \\
&= e^{r_t(x)} \big(b_t(x) \cdot \nabla r_t(x) + |\chi_t| \big(\|\nabla r_t(x)\|^2 + \Delta r_t(x) + \nabla r_t(x) \cdot s_t(x) \big) \big).
\end{aligned} \tag{88}$$

Plugging this into (87) yields

$$\partial_t e^{r_t(x)} = e^{r_t(x)} \mathcal{R}_t^\chi(x) - (b_t(x) + |\chi_t| s_t(x)) \cdot \nabla e^{r_t(x)} - |\chi_t| \Delta e^{r_t(x)}. \tag{89}$$

And plugging the expression of the infinitesimal generator in (51) into (89) yields

$$0 = e^{r_t(x)} \mathcal{R}_t^\chi(x) - [L_t^\chi e^{r_t}](x). \tag{90}$$

□

A.6 Flow Map Look-ahead is Optimal for Linear Rewards in a Gaussian Setting

In this section, we show that in a simplified Gaussian setting with a linear reward, the flow map is the unique scalar linear look-ahead that minimizes the accumulated variance of the SMC incremental log-weights. A linear reward simply corresponds to a mean shift of the terminal Gaussian distribution, so this setting is tractable and all relevant quantities have closed form solutions.

Formally, let $\rho_0(x) = \mathcal{N}(0, I)$ and $\rho_{data} = \rho_1(x) = \mathcal{N}(0, c^2 I)$. We consider the stochastic interpolant

$$x_t = (1-t)x_0 + tx_1, \qquad t \in [0,1], \tag{91}$$

where $x_0 \sim \rho_0, x_1 \sim \rho_1$ are independently drawn. The marginal distribution of x_t is

$$\rho_t = \mathcal{N}(0, \sigma_t^2 I_d), \qquad \sigma_t^2 = (1-t)^2 + c^2 t^2. \tag{92}$$

The score is therefore

$$s_t(x) = \nabla \log \rho_t(x) = -\frac{x}{\sigma_t^2}. \tag{93}$$

Using this, the flow matching field can be derived

$$b_t(x) = \mathbb{E}[x_1 - x_0 \mid x_t = x] = \left(\frac{c^2 t - (1-t)}{\sigma_t^2}\right) x = a_t x, \tag{94}$$

where

$$a_t = \frac{c^2 t - (1-t)}{\sigma_t^2}. \tag{95}$$

Equivalently,

$$a_t = \frac{\dot{\sigma}_t}{\sigma_t}. \tag{96}$$

Given this, we can write the probability-flow ODE as

$$\dot{x}_t = b_t(x_t) = a_t x_t. \tag{97}$$

Since $a_t = \dot{\sigma}_t / \sigma_t$, the flow map can be derived in closed form

$$X_{s,t}(x) = \frac{\sigma_t}{\sigma_s} x. \tag{98}$$

In particular, because $\sigma_1 = c$,

$$X_{t,1}(x) = \frac{c}{\sigma_t} x. \tag{99}$$

Similarly, the one-step denoiser can be derived as

$$D_t(x) = \mathbb{E}[X_1 \mid I_t = x] = \frac{c^2 t}{\sigma_t^2} x. \tag{100}$$

So, the three look-aheads are all linear functions of the form $L_t(x) = \ell_t \cdot x$ where

$$\ell_t^{\text{Flowmap}} = \frac{c}{\sigma_t}, \qquad \ell_t^{\text{Denoiser}} = \frac{c^2 t}{\sigma_t^2}, \qquad \ell_t^{\text{NoLookahead}} = 1. \tag{101}$$

Theorem A.7 (Optimality of the flow map for linear rewards)**.** *Let*

$$r(y) = \beta u^\top y, \qquad \|u\| = 1, \tag{102}$$

for some reward strength $\beta > 0$. Consider scalar linear look-aheads of the form

$$L_t(x) = \ell_t x, \tag{103}$$

where $\ell_1 = 1$, and define the time-dependent reward as

$$r_t^\ell(x) = tr(L_t(x)). \tag{104}$$

For the default Jarzynski/SMC correction with $\chi_t \equiv 0$, the ideal-case variance of the SMC incremental log-weights is

$$\mathcal{I}_\ell = \beta^2 \int_0^1 \left[\frac{d}{dt}\left(t\sigma_t \ell_t\right)\right]^2 dt. \tag{105}$$

Furthermore $\mathcal{I}_\ell \geq \beta^2 c^2$, with equality achieved if and only if using the flow map look-ahead.

Proof. Let $q_t = t\ell_t$, so that

$$r_t^\ell(x) = \beta q_t u^\top x. \tag{106}$$

For $\chi_t \equiv 0$, according to Proposition 3.1, the log-weight dynamics are

$$\frac{dA_t}{dt} = b_t(x_t) \cdot \nabla r_t^\ell(x_t) + \partial_t r_t^\ell(x_t). \tag{107}$$

Since

$$\nabla r_t^\ell(x) = \beta q_t u, \qquad \partial_t r_t^\ell(x) = \beta \dot{q}_t u^\top x, \tag{108}$$

and $b_t(x) = a_t x$ with $a_t = \dot{\sigma}_t/\sigma_t$, we have

$$\frac{dA_t}{dt} = \beta a_t q_t u^\top x_t + \beta \dot{q}_t u^\top x_t \tag{109}$$

$$= \beta(a_t q_t + \dot{q}_t) u^\top x_t \tag{110}$$

$$= \beta \left(\frac{\dot{\sigma}_t}{\sigma_t} q_t + \dot{q}_t \right) u^\top x_t \tag{111}$$

$$= \frac{\beta}{\sigma_t} \frac{d}{dt}(\sigma_t q_t) u^\top x_t. \tag{112}$$

Define

$$m_\ell(t) = \sigma_t q_t = t\sigma_t \ell_t. \tag{113}$$

Then

$$\frac{dA_t}{dt} = \beta \frac{\dot{m}_\ell(t)}{\sigma_t} u^\top x_t. \tag{114}$$

The tilted intermediate density is

$$\widehat{\rho}_t^\ell(x) \propto \rho_t(x) \exp(r_t^\ell(x)). \tag{115}$$

Because r_t^ℓ is linear and ρ_t is Gaussian, this tilt only changes the mean of ρ_t and leaves the covariance equal to $\sigma_t^2 I_d$. As a result,

$$\mathrm{Var}_{\widehat{\rho}_t^\ell}(u^\top X) = \sigma_t^2. \tag{116}$$

Therefore

$$\mathrm{Var}_{\widehat{\rho}_t^\ell}\left[\frac{dA_t}{dt} \right] = \beta^2 \frac{(\dot{m}_\ell(t))^2}{\sigma_t^2} \mathrm{Var}_{\widehat{\rho}_t^\ell}(u^\top X) \tag{117}$$

$$= \beta^2 (\dot{m}_\ell(t))^2. \tag{118}$$

So, the accumulated local variance is

$$\mathcal{I}_\ell = \beta^2 \int_0^1 (\dot{m}_\ell(t))^2 dt = \beta^2 \int_0^1 \left[\frac{d}{dt}(t\sigma_t \ell_t) \right]^2 dt. \tag{119}$$

Earlier, we assumed $\ell_1 = 1$, which implies

$$m_\ell(0) = 0, \qquad m_\ell(1) = \sigma_1 \ell_1 = c. \tag{120}$$

By Cauchy–Schwarz, we have

$$\int_0^1 (\dot{m}_\ell(t))^2 dt \geq \left(\int_0^1 \dot{m}_\ell(t) dt \right)^2 = (m_\ell(1) - m_\ell(0))^2 = c^2. \tag{121}$$

Equality holds if and only if $m'_\ell(t)$ is constant almost everywhere. Since $m_\ell(0) = 0$ and $m_\ell(1) = c$, this would mean $m_\ell(t) = ct$. Plugging this in to the definition of $m_\ell(t) = t\sigma_t \ell_t$ we have:

$$\ell_t = \frac{c}{\sigma_t}, \tag{122}$$

which is exactly the flow map look-ahead $X_{t,1}(x) = \frac{c}{\sigma_t} x$, completing the proof. □

Note that in the proof above, we assume an ideal case where x_t is distributed according to the tilted distribution $\widehat{\rho}_t^\ell(x) \propto \rho_t(x) \exp(r_t^\ell(x))$ for all t. In practice, with finite particles, this is only approximately true. Therefore, the value $\mathcal{I}_\ell$ should be viewed as an ideal asymptotic variance for the large-particle limit with sufficient resampling operations.

B Analyzing the Performance of Test-Time Sampling Algorithms

B.1 Simulating the Dynamics with SMC

The natural approach to handle the weights e^{A_t} in Proposition 3.2 and in Proposition A.1 is sequential Monte Carlo (SMC), which is implemented in Algorithm 1. Let $\mathcal{T} = (t_k)_{k=0}^K$ be an annealing schedule satisfying $0 = t_0 < \cdots < t_K = 1$. The SMC sampling procedure starts with N particles $\mathbf{X}_0 = (X_0^n)_{n\in[N]}$ drawn from the reference, $X_0^n \sim \rho_0$, and initial weights $\mathbf{w}_0 = (w_0^n)_{n\in[N]}$ with $w_0^n = 1$. For each subsequent iteration $k \in [K]$, we produce $\mathbf{X}_k$ and $\mathbf{w}_k$ by propagating, reweighting, and optionally resampling[2]. In what follows, we use a notation similar to the one of [36].

Propagate. Evolve $\mathbf{X}_{k-1}$ forward with the Markov transition kernel $M_{t_{k-1},t_k}(x_{k-1})$ to obtain $\mathbf{X}_k = (X_k^n)_{n\in[N]}$, where

$$X_k^n \sim M_{t_{k-1},t_k}(X_{k-1}^n). \tag{123}$$

For the dynamics of Proposition 3.2 and Proposition A.1, the Markov transition kernels are the Euler-Maruyama updates for the SDEs (13) and (20), respectively:

$$\begin{aligned} X_k^n &= X_{k-1}^n + [b_{t_{k-1}}(X_{k-1}^n) + \epsilon_{t_{k-1}}(s_{t_{k-1}}(X_{k-1}^n) + \nabla r_{t_{k-1}}(X_{k-1}^n))](t_k - t_{k-1}) \\ &\quad + \sqrt{2\epsilon_k(t_k - t_{k-1})}\xi_{k-1}^n, \quad \xi_{k-1}^n \sim N(0, \mathrm{I}), \end{aligned} \tag{124}$$

$$\begin{aligned} X_k^n &= X_{k-1}^n + [b_{t_{k-1}}(X_{k-1}^n) + \chi_{t_{k-1}}\nabla r_{t_{k-1}}(X_{k-1}^n) + \epsilon_{t_{k-1}}(s_{t_{k-1}}(X_{k-1}^n) + \nabla r_{t_{k-1}}(X_{k-1}^n))](t_k - t_{k-1}) \\ &\quad + \sqrt{2\epsilon_k(t_k - t_{k-1})}\xi_{k-1}^n, \quad \xi_{k-1}^n \sim N(0, \mathrm{I}), \end{aligned} \tag{125}$$

Reweight. Update the weights using the incremental weight function $g_{t_{k-1},t_k}(x_{k-1}, x_k)$:

$$\mathbf{w}_k = (w_k^n)_{n\in[N]}, \qquad w_k^n = w_{k-1}^n \, g_{t_{k-1},t_k}(X_{k-1}^n, X_k^n). \tag{126}$$

The incremental weight functions $g_{t_{k-1},t_k}(X_{k-1}^n, X_k^n)$ and weights w_k^n can be computed using the framework of Section A.4:

$$g_{t_{k-1},t_k}(X_{k-1}^n, X_k^n) = \exp\left(A_k^n - A_{k-1}^n\right), \qquad w_k^n = \exp\left(A_k^n\right). \tag{127}$$

Here, the difference $A_k^n - A_{k-1}^n$ is computed for each particle using one of the approaches described in Section A.4.

Resample (optional). On a (possibly random) subset of iterations $\mathcal{R} \subseteq [K]$ determined by a criterion depending on $(\mathbf{X}_k, \mathbf{w}_k)$, apply a resampling step

$$\mathbf{X}_k \leftarrow \mathrm{resample}(\mathbf{X}_k, \mathbf{w}_k),$$

to stabilize $\mathbf{w}_k$ and favor propagation of particles with higher relative weight. Concretely, set $X_k^n \leftarrow X_k^{a_k^n}$, where $a_k = (a_k^n)_{n\in[N]}$ is a random ancestor index vector with $a_k^n \in [N]$ and

$$\mathbb{P}(a_k^n = m \,|\, \mathbf{w}_k) = \frac{w_k^m}{\sum_{j\in[N]} w_k^j}, \qquad m \in [N].$$

After resampling, reset the weights via $w_k^n \leftarrow 1$ for all n. See [10] for annealed-SMC-specific resampling schemes, and [12] for a recent review of SMC samplers.

The SMC procedure yields an unbiased estimate of the expectation $\int_{\mathbb{R}^d} h(x)\hat{\rho}_{t_k}(x)dx$ for any test function h and any $k \in [K]$. Namely, if we let $\bar{\rho}_t$ be the unnormalized density as defined in Section A.1, recall that $\mathcal{R} \cap [k]$ denotes the subset of iterations where a resampling step happens, and for $r \in \mathcal{R}$ we let w_r be the weight prior to resetting to 1, we have that

$$\int_{\mathbb{R}^d} h(x)\bar{\rho}_{t_k}(x)dx = \mathbb{E}\left[\left(\prod_{r\in\mathcal{R}\cap[k-1]} \frac{1}{N}\sum_{n=1}^N w_r^n\right)\frac{1}{N}\sum_{n=1}^N w_k^n h(X_k^n)\right], \tag{128}$$

[2]While in Algorithm 1 we allow for a number of clones C greater than one, for simplicity, the analysis we perform in this section is with $C = 1$.

Setting $h \equiv 1$ and recalling that $\hat{\rho}_{t_k}(x) = \bar{\rho}_{t_k}(x)/\int_{\mathbb{R}^d} \bar{\rho}_{t_k}(x)dx$, we obtain that

$$\int_{\mathbb{R}^d} h(x)\hat{\rho}_{t_k}(x)dx = \frac{\mathbb{E}\left[\hat{Z}^{(k)}_{\mathrm{SMC}} \frac{\frac{1}{N}\sum_{n=1}^N w^n_{t_k} h(X^n_{t_k})}{\frac{1}{N}\sum_{n=1}^N w^n_{t_k}}\right]}{\mathbb{E}\left[\hat{Z}^{(k)}_{\mathrm{SMC}}\right]}, \quad \hat{Z}^{(k)}_{\mathrm{SMC}} = \left(\prod_{r\in\mathcal{R}\cap[k-1]} \frac{1}{N}\sum_{n=1}^N w^n_r\right) \times \left(\frac{1}{N}\sum_{n=1}^N w^n_k\right). \tag{129}$$

If we set $k = K$, we obtain that

$$\int_{\mathbb{R}^d} h(x)\hat{\rho}_1(x)dx = \frac{\mathbb{E}\left[\hat{Z}_{\mathrm{SMC}} \frac{\frac{1}{N}\sum_{n=1}^N w^n_K h(X^n_K)}{\frac{1}{N}\sum_{n=1}^N w^n_K}\right]}{\mathbb{E}\left[\hat{Z}_{\mathrm{SMC}}\right]}, \qquad \text{where } \hat{Z}_{\mathrm{SMC}} = \prod_{r\in\mathcal{R}} \frac{1}{N}\sum_{n=1}^N w^n_r. \tag{130}$$

$\hat{Z}_{\mathrm{SMC}}$ is known as the SMC normalization constant. Observe that $\mathbb{E}[\hat{Z}_{\mathrm{SMC}}] = \int_{\mathbb{R}^d} \bar{\rho}_1(x)dx := Z$. Thus, low $\mathrm{Var}[\hat{Z}_{\mathrm{SMC}}/Z]$ is a proxy for good performance of the SMC procedure. In Section B.3 we reproduce a result by [36] that expresses $\mathrm{Var}[\hat{Z}_{\mathrm{SMC}}/Z]$ in terms of the parameters of the SMC procedure, using the concept of total discrepancy from Section B.2.

B.2 The Incremental and Total Discrepancies

Following [36], we define the normalized incremental weight function $G_{t,t'}$ as

$$G_{t,t'}(x, x') = \frac{g_{t,t'}(x, x')}{\mathbb{E}_{(X,X')\sim\hat{\rho}_t\otimes M_{t,t'}}[g_{t,t'}(X, X')]} \tag{131}$$

where $g_{t,t'}$ is the unnormalized incremental weight function defined in (126) and $M_{t,t'}$ is the Markov transition kernel defined in (123).

Given $G_{t,t'}$ from time t to time t', the *incremental discrepancy* $D(t, t')$ is defined as

$$D(t, t') = \log\left(1 + \mathrm{Var}_{(X,X')\sim\hat{\rho}_t\otimes M_{t,t'}}\left(G_{t,t'}(X, X')\right)\right). \tag{132}$$

Given a sequence of timesteps $\mathcal{T} = (t_k)_{k=0}^K$ with $t_0 = 0$, $t_K = 1$, tor $k \le k'$, define $D(\mathcal{T}, t_k, t_{k'})$ as the *accumulated discrepancy* between iterations k and k' and $D(\mathcal{T})$ as the *total discrepancy*:

$$D(\mathcal{T}, t_k, t_{k'}) = \sum_{k''=k+1}^{k'} D(t_{k''-1}, t_{k''}), \qquad D(\mathcal{T}) = D(\mathcal{T}, 0, 1). \tag{133}$$

Observe that the incremental discrepancy can be expressed in terms of the first and second moments of $g_{t,t'}(X, X')$:

$$\begin{aligned} D(t, t') &= \log\left(1 + \mathrm{Var}_{(X,X')\sim\hat{\rho}_t\otimes M_{t,t'}}\left[\frac{g_{t,t'}(X, X')}{\mathbb{E}_{(X'',X''')\sim\hat{\rho}_t\otimes M_{t,t'}}[g_{t,t'}(X'', X''')]}\right]\right) \\ &= \log\left(1 + \frac{\mathbb{E}_{(X,X')\sim\hat{\rho}_t\otimes M_{t,t'}}\left[g_{t,t'}(X, X')^2\right]}{\mathbb{E}_{(X'',X''')\sim\hat{\rho}_t\otimes M_{t,t'}}[g_{t,t'}(X'', X''')]^2} - \frac{\mathbb{E}_{(X,X')\sim\hat{\rho}_t\otimes M_{t,t'}}\left[g_{t,t'}(X, X')\right]^2}{\mathbb{E}_{(X'',X''')\sim\hat{\rho}_t\otimes M_{t,t'}}[g_{t,t'}(X'', X''')]^2}\right) \\ &= \log \mathbb{E}_{(X,X')\sim\hat{\rho}_t\otimes M_{t,t'}}\left[g_{t,t'}(X, X')^2\right] - 2\log \mathbb{E}_{(X,X')\sim\hat{\rho}_t\otimes M_{t,t'}}\left[g_{t,t'}(X, X')\right] \end{aligned} \tag{134}$$

We want to obtain a consistent estimator for $D(t, t')$. Following Sec 5.1 of [36], if we are using a single SMC run with N particles, and we let $g^n_k = g_{t_{k-1},t_k}(X^n_{k-1}, X^n_k)$, then the following estimator is consistent:

$$\hat{D}_k = \log \hat{g}_{k,2} - 2\log \hat{g}_{k,1} - \log \hat{g}_{k,0}, \quad \text{where for } i \in \{0, 1, 2\}, \quad \hat{g}_{k,i} = \sum_{n\in[N]} w^n_{k-1}\left(g^n_k\right)^i. \tag{135}$$

To get a consistent estimator using N_R SMC runs, each with N particles, we compute the normalization constant $\hat{Z}_{\mathrm{SMC}}^{(k,j)}$ in (129) for each SMC run $j = 1 : N_R$, and define $\hat{D}_k$ as in (135), but where $\hat{g}_{k,i}$ takes the form

$$\hat{g}_{k,i} = \sum_{j=1}^{N_R} \hat{Z}_{\mathrm{SMC}}^{(k,j)} \frac{\sum_{n\in[N]} w_{k-1}^{(n,j)} \left(g_k^{(n,j)}\right)^i}{\sum_{n\in[N]} w_{k-1}^{(n,j)}}, \qquad \text{where} \quad g_k^{(n,j)} = g_{t_{k-1},t_k}(X_{k-1}^{(n,j)}, X_k^{(n,j)}), \tag{136}$$

and $X_k^{(n,j)}$, $w_k^{(n,j)}$ is the n-th particle and weight of j-th SMC run at iteration k.

B.3 The Variance of the SMC Normalization Constant

The total discrepancy defined in equation (133) is related to the variance of the SMC normalization constant Z_{SMC} via the following result, which was proven by [36] as a generalization of a result of [12]:

Theorem B.1 (Theorem 1, [36])**.** *Suppose that the following assumptions on the normalized incremental weights $G_k^n = G_{t_{k-1},t_k}(X_{k-1}^n, X_k^n)$ defined in (131) hold:*

- Assumption 1 (Integrability). *For all $n \in [N]$, $k \in [K]$, G_k^n has finite variance with respect to $\hat{\rho}_{t_k} \otimes M_{t_{k-1},t_k}$.*
- Assumption 2 (Temporal indep.). *For $n \in [N]$, $(G_k^n)_{t\in[T]}$ are independent.*
- Assumption 3 (Particle indep.). *For $k \in [K]$, $(G_k^n)_{n\in[N]}$ are independent.*
- Assumption 4 (Efficient local moves). *For each $n \in [N]$ and $k \in [k]$,*

$$G_k^n \overset{d}{=} G_{t_{k-1},t_k}(X_{k-1}, X_k), \qquad (X_{k-1}, X_k) \sim \pi_{t_{k-1}} \otimes M_{t_{k-1},t_k}.$$

Assume also that $N > 1$, $D(\mathcal{T}) > 0$. For every resample schedule $\mathcal{T}_R = (t_r)_{r=0}^R$, there exists a unique $1 \le R_{\mathrm{eff}} \le \mathbb{E}[R]$ such that

$$\mathrm{Var}\left(\frac{\hat{Z}_{\mathrm{SMC}}}{Z}\right) = \frac{1}{N}\left(\exp\left(\frac{D(\mathcal{T})}{R_{\mathrm{eff}}}\right) - 1\right) R_{\mathrm{eff}} - 1. \tag{137}$$

Moreover, $R_{\mathrm{eff}} = 1$ if and only if $D(\mathcal{T}, t_{r-1}, t_r) \overset{\text{a.s.}}{=} D(\mathcal{T})$ for some $r \in [R]$, and $R_{\mathrm{eff}} = \mathbb{E}[R]$ if and only if R is a.s. constant and $D(\mathcal{T}, t_{r-1}, t_r) \overset{\text{a.s.}}{=} D(\mathcal{T})/R$ for some $r \in [R]$.

Remark B.2. As remarked by [36], Assumptions 1–4 constitute an idealized model similar to the one considered by other works in the area [12, 17]. While Assumption 1 is weak, Assumptions 2–4 are not. Assumption 3 only holds when no resampling is performed, and Assumptions 2–4 only hold (approximately) when a number of MCMC steps are interleaved with the SMC updates. However, Syed et al. [36, Sec. 6.1] show that empirically, the scaling (137) is consistent with empirical observation.

B.4 Optimizing the Annealing Schedule to Minimize the Total Discrepancy

As defined in (133), the total discrepancy depends not only on the continuous time dynamics for positions and weights, but also on the specific annealing schedule $\mathcal{T} = (t_k)_{k=0}^K$. For a fixed K, it is possible to characterize and find the annealing schedule that minimizes the total discrepancy.

Under technical regularity assumptions (see Sec 4.1 of [36]), the incremental discrepancy admits the asymptotic expansion

$$G_{t,t+\Delta t} = 1 + S_t\,\Delta t + o(\Delta t),$$

and hence the local changes and variance of the incremental discrepancy $G_{t,t+\Delta t}$ are encoded in S_t and its variance $\delta(t)$, defined as

$$S_t = \left|\frac{\partial}{\partial t'} G_{t,t'}\right|_{t'=t}, \qquad \delta(t) = \mathrm{Var}_{t,t}[S_t].$$

Using this expansion, we can expand the incremental discrepancy as follows:

$$D(t, t+\Delta t) = \delta(t)\Delta t + O(\Delta t^3). \tag{138}$$

Scheduler generators A *schedule generator* is a continuously twice-differentiable function $\varphi : [0,1] \to [0,1]$ such that $\varphi(0) = 0$, $\varphi(1) = 1$, and $\dot{\varphi}(u) = \frac{d}{du}\varphi(u) > 0$. Given $K \in \mathbb{N}$, φ generates an annealing schedule $\mathcal{T} = (t_k)_{k=0}^K$ where

$$t_k = \varphi(u_k), \qquad u_k = \frac{k}{K}.$$

In the following, without loss of generality, we restrict our attention to schedules generated by schedule generator.

By the mean value theorem, we have

$$t_k - t_{k-1} \approx \frac{\dot{\varphi}(u_k)}{K}.$$

Combining this with equation (138), we obtain

$$D(t_{k-1}, t_k) \approx \frac{\delta(\varphi(u_k))\,\dot{\varphi}(u_k)^2}{K^2}. \tag{139}$$

By summing over k and using Riemann approximations, we can approximate $D(\mathcal{T}, t_k, t_{k'})$ and $D(\mathcal{T})$ in terms of $E(\varphi, u_{t_k}, u_{t_{k'}})$ and $E(\varphi)$, defined as the integral of $\delta(\varphi(u))\,\dot{\varphi}(u)^2$,

$$E(\varphi, u, u') = \int_u^{u'} \delta(\varphi(v))\,\dot{\varphi}(v)^2\,dv, \qquad E(\varphi) = E(\varphi, 0, 1).$$

Proposition B.3 (Proposition 1, [36])**.** *Suppose Assumptions 5 to 8 hold. There exists $C_D(\varphi) > 0$ such that, for $k, k' \in [K]$,*

$$\left| D(\mathcal{T}, t_k, t_{k'}) - \frac{1}{K} E(\varphi, u_k, u_{k'}) \right| \leq \frac{C_D(\varphi)\,|t_{k'} - t_k|}{T^3}.$$

An immediate consequence of Proposition 1 is that, in the dense schedule limit as $K \to \infty$, the total discrepancy $D(\mathcal{T})$ is asymptotically equivalent to $\frac{E(\varphi)}{K}$. Hence, for a fixed K, optimizing $D(\mathcal{T})$ with respect to $\mathcal{T}$ is asymptotically equivalent to the following problem:

$$\min_{\varphi:[0,1]\to[0,1]} \int_0^1 \delta(\varphi(u))\,\dot{\varphi}(u)^2\,du, \quad \text{s.t.} \quad \int_0^1 \dot{\varphi}(u)\,du = 1. \tag{140}$$

Jensen's inequality implies that

$$\int_0^1 \delta(\varphi(u))\,\dot{\varphi}(u)^2\,du \geq \left(\int_0^1 \sqrt{\delta(\varphi(u))}\,\dot{\varphi}(u)\,du \right)^2, \tag{141}$$

with equality if and only if there exists a constant $\Lambda > 0$ such that $\sqrt{\delta(\varphi^\star(u))}\,\dot{\varphi^\star}(u) = \Lambda$ for u a.e. in $[0,1]$. Defining

$$\Lambda(t) = \int_0^t \sqrt{\delta(u)}\,du, \tag{142}$$

by the chain rule, we have equivalently that

$$\Lambda = \Lambda'(\varphi^\star(t))\,\dot{\varphi^\star}(t) = \frac{d}{dt}\Lambda(\varphi^\star(t)) \implies \Lambda(\varphi^\star(t)) = \Lambda t \implies \varphi^\star(t) = \Lambda^{-1}(\Lambda t). \tag{143}$$

Setting $t = 1$ in $\Lambda(\varphi^\star(t)) = \Lambda t$ also implies that $\Lambda = \Lambda(\varphi^\star(1)) = \Lambda(1) = \int_0^1 \sqrt{\delta(u)}\,du$. [36] refer to $\Lambda(t)$ and Λ as the *local barrier* and the *global barrier* associated to the SMC algorithm. We refer to Λ as the **thermodynamic length** associated to the algorithm.

A change of the integration variable implies that for any schedule generator φ,

$$\begin{aligned}
&\Lambda(\varphi(t)) = \int_0^{\varphi(t)} \sqrt{\delta(u)}\,du = \int_0^t \sqrt{\delta(\varphi(u))}\dot{\varphi}(u)\,du, \\
&\implies \Lambda(t_k) = \Lambda(\varphi(u_k)) = \int_0^{u_k} \sqrt{\delta(\varphi(u))}\dot{\varphi}(u)\,du \approx \sum_{k'=1}^{k} \frac{\sqrt{\delta(\varphi(u_k))}\,\dot{\varphi}(u_k)}{K} = \sum_{k'=1}^{k} \sqrt{D(t_{k-1}, t_k)}, \\
&\Lambda = \int_0^1 \sqrt{\delta(\varphi(u))}\dot{\varphi}(u)\,du \approx \sum_{k=1}^{K} \sqrt{D(t_{k-1}, t_k)}.
\end{aligned} \tag{144}$$

where the last equality holds by (139). This allows us to approximate the local and global barriers using (134) to compute the incremental discrepancy $D(t_{k-1}, t_k)$. Once we have an estimate $\hat{\Lambda}$ of the barrier, [36] propose to iteratively refine the annealing schedule by resetting $t_k \leftarrow \hat{\Lambda}^{-1}(\hat{\Lambda}k/K)$.

Observe that given the quantities $\sum_{k'=1}^{k} D(t_{k'-1}, t_{k'})$ and $\sum_{k'=1}^{k} \sqrt{D(t_{k'-1}, t_{k'})}$, we have that

$$\sum_{k'=1}^{k} D(t_{k'-1}, t_{k'}) = D(\mathcal{T}, 0, t_{k'}) \approx \frac{1}{K} E(\varphi, u_0, u_{k'}) \geq \frac{1}{K} \Lambda(t_k)^2 \approx \frac{1}{K} \left(\sum_{k'=1}^{k} \sqrt{D(t_{k'-1}, t_{k'})} \right)^2, \tag{145}$$

Thus, the quantity

$$\frac{\left(\sum_{k'=1}^{k} \sqrt{D(t_{k'-1}, t_{k'})} \right)^2}{K \sum_{k'=1}^{k} D(t_{k'-1}, t_{k'})} \tag{146}$$

should fall within $[0, 1]$ and close to 1 when the annealing schedule $\mathcal{T}$ is close to the optimal one.

C Implementation Details

The complete pseudocode of FMTT is given in Algorithm 1.

Algorithm 1 Inference-time adaptation of flow maps

Input: # simulation steps K, # resampling steps R, # particles N, # particle clones C, time sequence $(t_k)_{k=0:K}$, sequences $(\epsilon_k)_{k=0:K}$ and $(\chi_k)_{k=0:K}$
for $i = 1 : N$, initialize $x_i^0 = N(0, \mathrm{I})$ i.i.d. Let $X^0 = (x_i^0)_{i=1:N}$.
Clone the particles: $\bar{X}^0 = (x_{ij}^0)_{i=1:N, j=1:C}$, where $(x_{ij}^0)^{j=1:C}$ are equal copies of x_i^0.
if Sampling **then for** $i = 1 : N$ and $j = 1 : C$, initialize $A_{ij}^0 = 0$.
for $k = 0 : K - 1$ **do**
  $\Delta t \leftarrow t_{k+1} - t_k$
  for $i = 1 : N$ and $j = 1 : C$ **do** ▷ Update particles
    $x_{ij}^{k+1} = x_{ij}^k + [v_{t_k, t_k}(x_{ij}^k) + \chi_k \nabla r_{t_k}(x_{ij}^k) + \epsilon_k (s_{t_k}(x_{ij}^k) + \nabla r_{t_k}(x_{ij}^k))]\Delta t + \sqrt{2\epsilon_k \Delta t}\xi_{ij}^k, \quad \xi_{ij}^k \sim N(0, \mathrm{I})$.
    if Sampling **then**
      Compute A_{ij}^{k+1} from A_{ij}^k using approach *(i)* (66), approach *(ii)* (68) or approach *(iii)* (70)-(72) from Section A.4. Note that simplifications occur when $\chi_k \equiv 0$ (Remark A.3).
    end if
  end for
  if $k = 0 \pmod{K/R}$ and $k > 0$ **then** ▷ Resample / select particles
    if Sampling **then**
      Define probabilities $p^k = \mathrm{softmax}(A^k) = \left(\exp(A_{ij}^k) / \sum_{i'j'} \exp(A_{i'j'}^k) \right)_{i'=1:N, j'=1:C}$
      Resample $X^k = (x_i^k)_{i=1:N} \sim \sum_{i'=1}^{n} \sum_{j'=1}^{C} p_{i'j'}^k \delta_{x_{i'j'}^k}$ i.i.d., or using Quasi-Monte Carlo
      Set $A_{ij}^k = 0$.
    else if Searching **then**
      Select $X^k = (x_i^k)_{i=1:N}$ as the top-n samples among $\bar{X}^k$ with respect to $r_{t_k}(x_{ij}^k)$.
    end if
    Clone the particles: $\bar{X}^k \leftarrow (x_{ij}^k)_{i=1:N}^{j=1:C}$, where $(x_{ij}^k)^{j=1:C}$ are equal copies of x_i^k.
  end if
end for
return x

D Additional Experiments

D.1 Toy 2D Experiments

In this section, we provide additional experiments for our 2D toy data setting. We consider a ring-shaped base distribution and define a reward function r that has a high value for samples within the base distribution that lie on a zigzag-shaped pattern, and low values elsewhere. The reward along with the base and tilted distributions can be seen in Figure 9. We choose this reward since it mimics the behavior of complex VLM-based rewards, due to the fact that its values are relatively flat away from the data manifold resulting in unhelpful gradients.

To model the base distribution, we train a simple continuous-time flow matching model to learn the velocity field, followed by distilling into a flow map. We train both models for 100,000 steps with a batch size of 8092. After training, we run SMC algorithm defined by Equations (13) and (17) for 50, 100, 200, and 500 sampling steps with 10,000 particles, and perform resampling when the effective sample size falls beneath 0.9 using a stochasticity factor of $\epsilon_t = 0.1 \times (1 - t)$. We calculate Maximum Mean Discrepancy (MMD) and Sliced Wasserstein-2 (SW2) against samples obtained via rejection sampling, and summarize the results in Table 3. Qualitative results can be seen in Figure 10.

As can be seen from the results, both naive- and denoiser- look-aheads fail to capture the true tilted distribution, with their outputs degenerating into small blob-like subsets. This is also reflected quantitatively: despite achieving high mean rewards, these methods have much larger MMD and SW2 distances, indicating that they over-optimize the reward while losing coverage of the tilted target. In contrast, flow map look-ahead consistently achieves much lower MMD and SW2 across all step counts, while requiring far fewer resampling steps. This suggests that the flow map provides a much more reliable look-ahead for SMC, allowing the particle system to track the tilted distribution without collapsing. Furthermore, the lower number of resampling steps suggests more stable SMC behavior with smaller importance weights variance.

Table 3: **Quantitative results for the 2D toy experiment.** We compare no-lookahead, denoiser-lookahead, and flow map-lookahead across different numbers of SMC steps. Lower Maximum Mean Discrepancy (MMD) and Sliced Wasserstein-2 (SW2) indicate better agreement with the tilted target distribution.

Steps	Method	MMD ↓	SW2 ↓	Mean Reward ↑	# of Resamples ↓
50	No lookahead	0.996	0.578	0.911	46
	Denoiser	3.973	0.988	0.917	47
	Flow map	**0.043**	**0.060**	0.621	5
100	No lookahead	0.460	0.443	0.958	64
	Denoiser	4.131	0.979	0.949	81
	Flow map	**0.065**	**0.083**	0.925	10
200	No lookahead	4.332	0.902	0.977	84
	Denoiser	3.016	0.942	0.964	135
	Flow map	**0.065**	**0.126**	0.967	9
500	No lookahead	3.459	0.857	0.972	155
	Denoiser	3.717	0.977	0.976	198
	Flow map	**0.031**	**0.089**	0.971	6

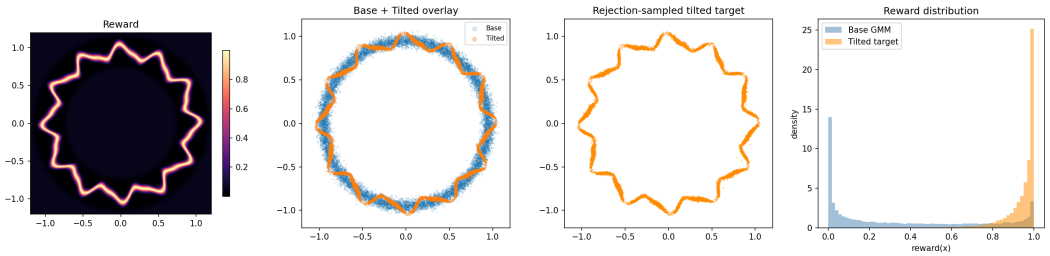
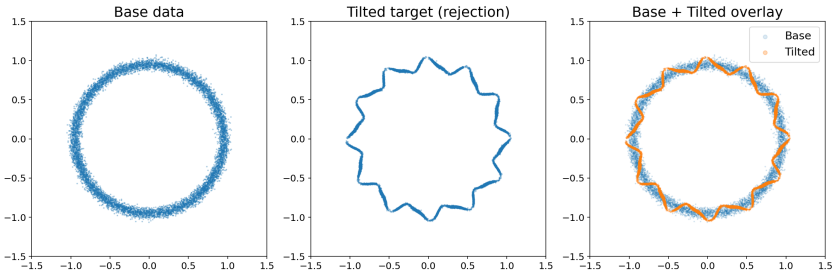
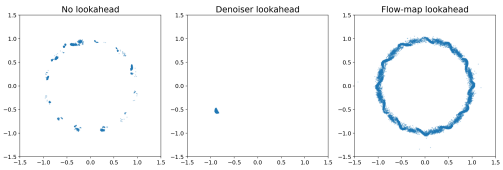


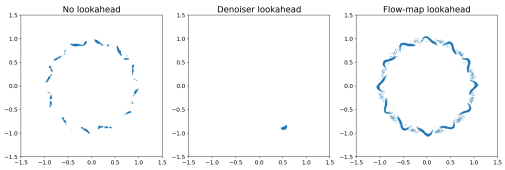
Figure 9: **Tilting a 2D toy distribution**. We train a diffusion model on the ring-shaped distribution, and choose a reward that tilts it toward the target zigzag shape. Running SMC with 500 steps, no-lookahead and denoiser-lookahead fail to recover the target, while flow map- lookahead can effectively capture it. See Section D.1 for details.



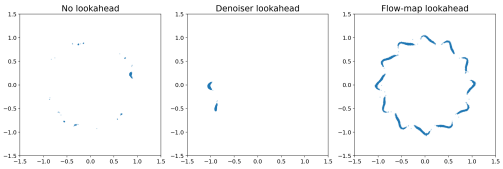
(a) Base and reward-tilted distributions.



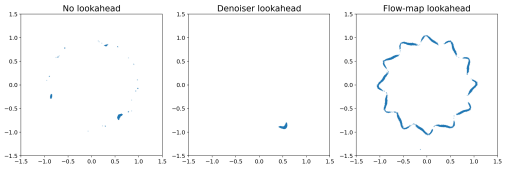
(b) 50 SMC steps



(c) 100 SMC steps



(d) 200 SMC steps



(e) 500 SMC steps

Figure 10: **Tilting a 2D toy distribution.** We run SMC with varying numbers of steps. Across step counts, no-lookahead and denoiser-lookahead fail to recover the target, while flow map-lookahead captures it much more effectively.

D.2 MNIST Experiments

We ran MNIST experiments using a flow map model that we trained from scratch, with stochastic interpolants $\alpha_t = 1 - t$, $\beta_t = t$. We use the unconditional flow map, which samples the distribution ρ_1 of all MNIST digits. We run Algorithm 1 with $N = 128$ particles and $K = 200$ simulation steps.

We ablate the following look-ahead approaches introduced in Section 3.2:

- Naive look-ahead / no look-ahead (NL): $r_t(x) = t \cdot r(x)$.
- Denoiser look-ahead (WD): $r_t(x) = t \cdot r(\mathsf{D}_t(x))$.
- Flow map look-ahead (FMTT): $r_t(x) = t \cdot r(X_{t,1}(x))$.

We set $\epsilon_t = \alpha_t = 1 - t$. We consider the four choices of χ_t described in Section A.3:

- Default: $\chi_t = 0$.
- Tilted score dynamics: $\chi_t = \eta_t := \alpha_t(\frac{\dot\beta_t}{\beta_t}\alpha_t - \dot\alpha_t) = (1-t)\big(\frac{1-t}{t}+1\big) = \frac{1-t}{t}$. In practice, we add an offset of 0.05 to β_t in the denominator: $\chi_t = \alpha_t(\frac{\dot\beta_t}{\beta_t+0.05}\alpha_t - \dot\alpha_t) = (1-t)\big(\frac{1-t}{t+0.05}+1\big) = \frac{1.05\times(1-t)}{t+0.05}$.
- Local tilt dynamics: $\chi_t = \epsilon_t = 1-t$.
- Base dynamics: $\chi_t = -\epsilon_t = -1+t$.

We run Sampling and Searching experiments.

- For Sampling experiments (Figure 11), we set $r(x) = 0.1 \times \log p_\theta(0|x)$, where p_θ is a pre-trained MNIST classifier and $\log p_\theta(0|x)$ is the probability that the classifier assigns to the digit 0 for the image x. Following Section 3, the target distribution that we aim to sample is the tilted probability distribution is $\hat\rho_1(x) \propto \rho_1(x)\exp\big(0.1 \times \log p_\theta(0|x)\big)$. We run the sampling experiments with $C = 1$ clone. We perform the log-weight update in Line 10 using approach *(iii)* (see Section A.4.1 for a discussion on the different approaches). We remark again that the choice $\chi_t = 0$ is much faster computationally, as it does not require estimating expectations. For the other three choices of χ_t, we use $M = 400$ noisy samples when computing (70) or (72). Instead of resampling every R steps with R fixed, we resample when the effective sample size, which is computed as $\sum_{i=1}^N \exp(A_i^k)^2 / \big(\sum_{i=1}^N \exp(A_i^k)\big)^2$, falls below $0.85 \times N$.
- For Searching experiments (Figure 12), $r(x) = 0.05 \times \log p_\theta(0|x)$. Note that in this case, there does not exist an explicit form for the target probability distribution, which is different for each algorithm. We run the simulation with $C = 2$ clones, and do one resampling step after the 100th simulation step (in the middle of the generation).

Sampling results In Figure 11, we show samples generated using each algorithm and plot four metrics. For each algorithm, the metrics were computed from 16 runs, each with $N = 128$ particles, and the bars show the average value plus/minus the standard error.

- Average reward: this is the expected value of $\log p_\theta(0|x)$ for the samples generated with the algorithm (without the multiplier 0.1). We also print the ground truth value, which we compute by sampling $400 \times 128 = 51200$ independent samples from the unconditional model, and reweighting them by the factor $\exp\big(0.1 \times \log p_\theta(0|x)\big)$ upon normalization.
- Class entropy: this is the average value of the entropy of the probability distribution $(p_\theta(i|x))_{i=0}^9$ induced by the classifier p_θ for each generated sample x, i.e. $-\sum_{i=0}^9 \log p_\theta(i|x)$. We also print the ground truth value, which we compute by sampling 51200 samples and reweighting them accordingly.
- Total discrepancy: this is the quantity $D(\mathcal{T}) = \sum_{k=1}^K D(t_{k-1}, t_k)$ defined in equation (133), where $D(t_{k-1}, t_k)$ are computed using the procedure described in Section B.2.
- Thermodynamic length: this is the $\Lambda = \sum_{k=1}^K \sqrt{D(t_{k-1}, t_k)}$ defined in (144) (strictly speaking, this is an approximation of the actual thermodynamic length).

We observe that the only χ_t choice for which the ground truth values for the average reward and the class entropy fall within the error bars is $\chi_t = 0$. For this choice, the look-ahead approach that results in lower total discrepancy and thermodynamic length is FMTT. To understand why this is the case, recall that the evolution of the log-weights for $\chi_t = 0$ is given by the ODE $\frac{\mathrm{d}A_t}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}s} r_s(X_s^0)\big|_{X_t^0 = \tilde x_t, s=t}$. Observe that when $r_t(x) = r_t^\star(x) := r(X_{t,1}(x))$, then $\frac{\mathrm{d}A_t}{\mathrm{d}t} = 0$, which implies that the total discrepancy and the thermodynamic length are zero, and it is easy to see that this is the only choice of r_t for which they are zero[3]. Thus, the total discrepancy and the thermodynamic length values can be regarded as a measure of how close a given r_t is to $r_t^\star := r \circ X_{t,1}$. From their values in the first row of Figure 11, we deduce that the flow map look-ahead reward $r_t(x) = t \cdot r(X_{t,1}(x))$ is the closest one to $r_t^\star(x) = r(X_{t,1}(x))$, which is unsurprising because they only differ by a factor t.

[3] Observe that $r_t(x) = r_t^\star(x) = r(X_{t,1}(x))$ cannot be used in practice because we require that $r_0(x)$ is constant so that we can sample from $\hat\rho_0$

For all other χ_t choices, the ground truth class entropy values fall outside of the error bars of the model. Moreover, for these other χ_t choices, the total discrepancy and thermodynamic length are substantially higher for FMTT than for NL and WD, and are lowest for WD. To understand why this is the case, observe that the evolution of the log-weights for $\chi_t = \eta_t$ and $\chi_t = \epsilon_t$ is given by the ODE $\frac{\mathrm{d}A_t}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}s}\mathbb{E}[e^{r_s(X_s^\chi) - r_t(\tilde{x}_t)} \mid X_t^\chi = \tilde{x}_t]\big|_{s=t}$. Note that when r_t is the optimal time-dependent reward in the stochastic optimal control sense, i.e., $r_t(x) = r_t^\star(x) := \log \mathbb{E}[\exp(r_1(X_1^\chi))|X_t^\chi = \tilde{x}_t]$[4], we have that $\mathbb{E}[e^{r_s(X_s^\chi) - r_t(\tilde{x}_t)} \mid X_t^\chi = \tilde{x}_t] = 0$ for all $0 \leq t \leq s \leq 1$. Hence, for this choice of r_t, the total discrepancy and the thermodynamic length are zero, and it is the only choice for which they are zero. Like before, the total discrepancy and the thermodynamic length values can be regarded as a measure of how close a given r_t is to $r_t^\star$. From their values in the second and third row of Figure 11, we deduce that the denoiser look-ahead reward $r_t(x) = t \cdot r(\mathsf{D}_t(x))$ is the closest one to $r_t^\star(x)$, and $r_t(x) = t \cdot r(X_{t,1}(x))$ is the farthest apart.

Searching results In Figure 12, we show samples generated using each algorithm and plot the average reward and average class entropy methods, which are also computed from 16 runs with $N = 128$ particles each, as in the sampling case. In this case, there are no ground truth values to compare to. We observe that all else equal, FMTT with $\chi_t = \eta_t$ is the most effective choice in that it achieves the highest average reward and the lowest class entropy, hence sampling the digit 0 more often. Accordingly, FMTT with $\chi_t = \eta_t$ is the choice we make for our high dimensional search experiments throughout the paper.

[4]This holds by the path integral characterization of stochastic optimal control.

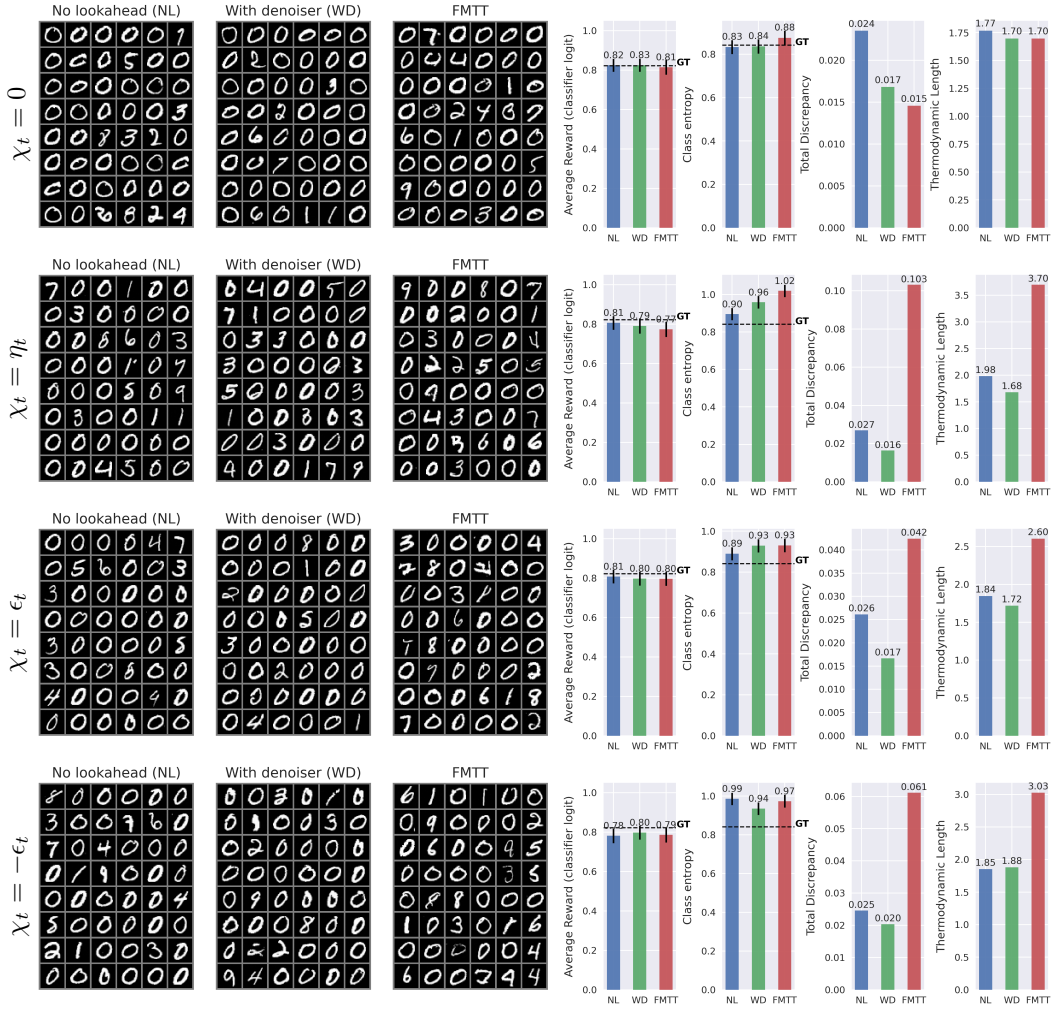


Figure 11: Comparison of MNIST tilted sampling to generate digits that would be classified as zeros: the reward is $r(x) = 0.1 \times \log p(0|x)$. The four rows correspond to the dynamics with different χ_t choices described in Section A.3.

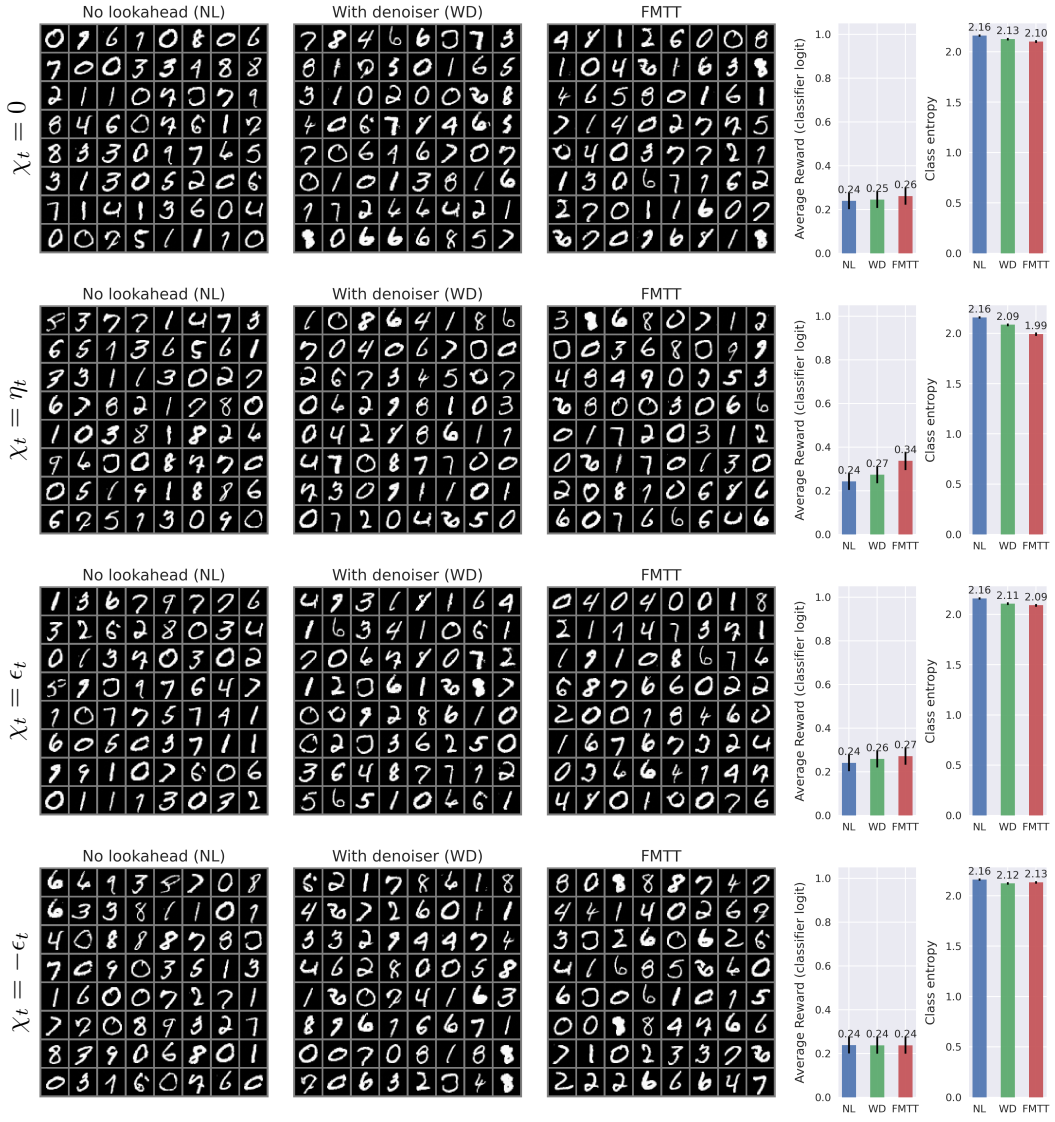


Figure 12: Comparison of MNIST greedy search to generate digits that would be classified as zeros: the reward is $r(x) = 0.05 \times \log p(0|x)$. The four rows correspond to the dynamics with different χ_t choices described in Section A.3.

D.3 Text-to-Image Experiments

D.3.1 Additional Geometric Reward Examples

Additional qualitative comparisons when using a geometric reward can be seen in Figures 13 to 15.

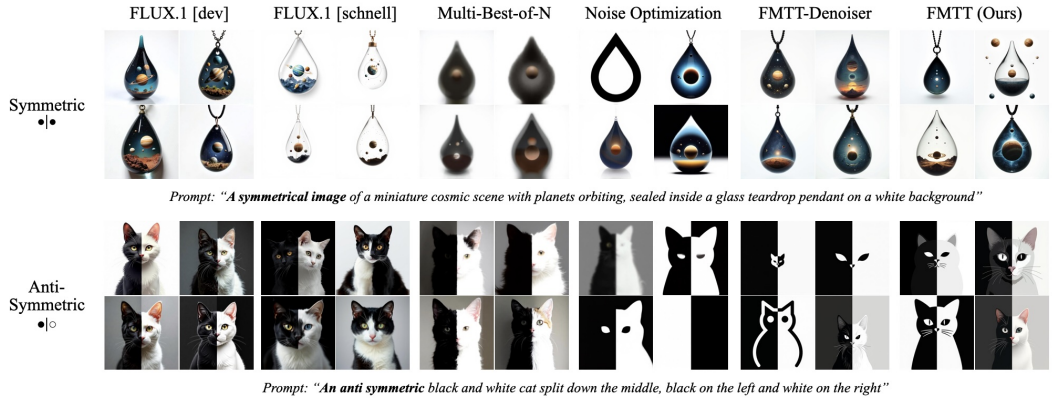


Figure 13: Qualitative comparison on two basic geometric rewards (symmetry, anti-symmetry). The gradient-based methods that change the generative dynamics produce sharper images that satisfy the constraints more reliably than prior methods. Note how Multi-best-of-N and ReNO-style noise optimization result in blurry images while FMTT methods produce clear results that properly satisfy the constraints. See Section D.3.1 for additional examples.

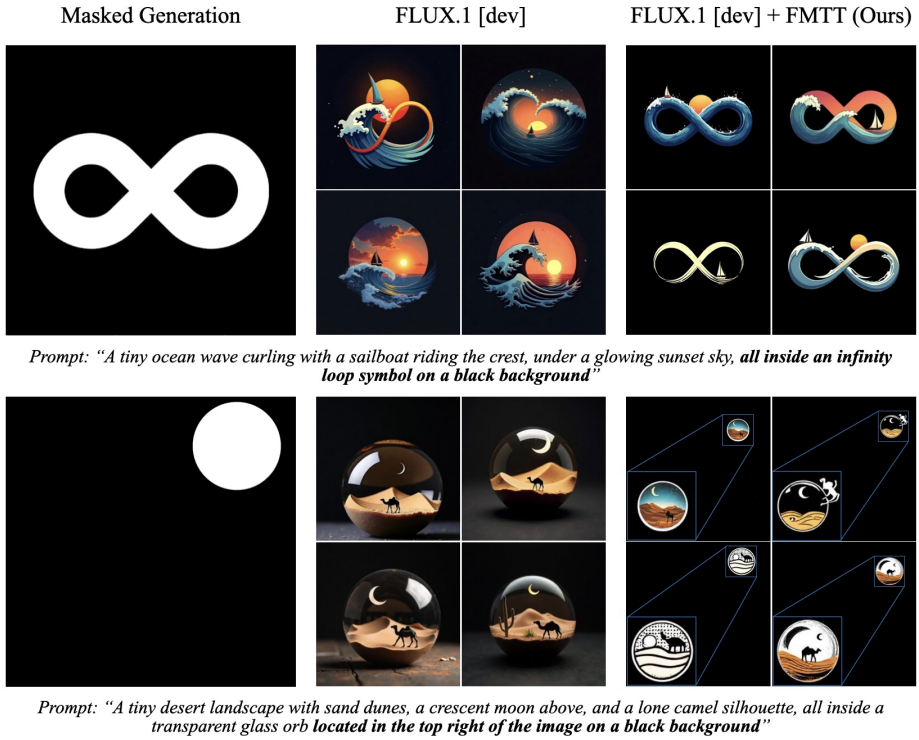


Figure 14: Additional qualitative comparisons using masked rewards. Note the zoom-ins in the second example for FMTT.



Figure 15: Additional qualitative comparison using rotation invariant rewards.

D.3.2 Additional VLM-as-judge Examples

Additional qualitative results with VLM-as-judge rewards can be seen in Figure 16.

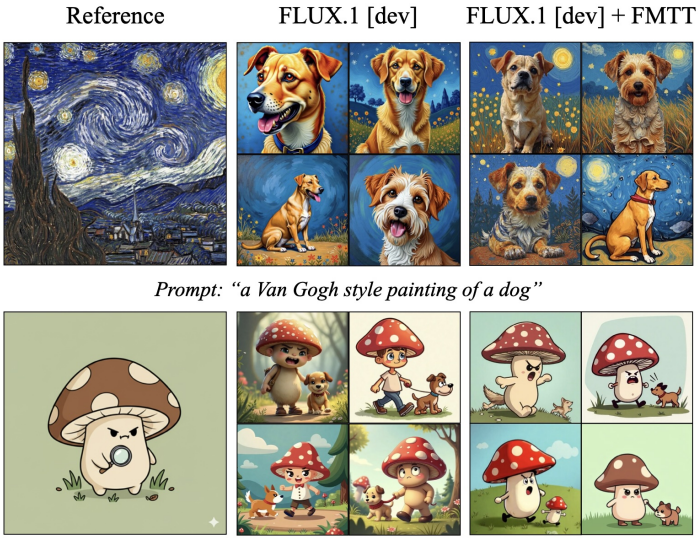


Figure 16: Additional qualitative comparison using VLM-as-judge rewards.

As described in the paper, we also use Qwen2.5-VL-7B-Instruct to define rewards expressed as yes/no questions over one or more context images. This makes it possible to cast diverse objectives as test-time search problems, including style consistency and character consistency. Here, we demonstrate the style consistency case. The VLM receives both a reference image and a generated image and is asked whether they share the same art style. FMTT then optimizes this reward, producing generations more closely aligned with the reference style. The qualitative results are shown in Figure 17.

D.3.3 VLM Reward Hacking

As discussed in the paper, a challenge of using VLMs (or any non-verifiable reward model) is the risk of the search process exploiting loopholes. This happens when the algorithm produces images that either act as adversarial examples for the VLM or satisfy the literal question without achieving the intended effect. Figure 18 shows such a case.



Prompt: "a Van Gogh style painting of a dog"

Prompt: "a cute grumpy cartoon mushroom character walking his dog"

Figure 17: Style consistency via VLM-based rewards. Given a reference image, FMTT produces images that better match its art style than the base model.



Prompt: "An analog clock showing exactly 4:45"
VLM question: "Is the analog clock showing 4:45?"

Figure 18: VLM reward hacking. Instead of the clock being at 4:45, the search process finds a way to "cheat" by writing the text `4:45` on the clock face and achieving high rewards from the VLM.

We explored two solutions. The first is to craft the VLM prompt to be as verbose and unambiguous as possible, explicitly discouraging potential "cheats". This works when only a few edge cases exist, but becomes brittle when many (4+) conditions are needed, at which point the reward model grows opaque and the search converges to local maxima. The second approach is to decompose the binary question into several simpler sub-questions and define the reward as their sum. While this adds computational overhead by requiring multiple VLM inferences, it proved more robust in practice.

For reference, to achieve the results in Figure 1, we used the following three questions:

- Is the hour hand pointing between 4 and 5?
- Is the minute hand pointing at 9?
- Is the second hand pointing at 12?

D.3.4 Complete UniGenBench++ Results

Table 4 reports the complete per-dimension scores used by UniGenBench++, including entity layout, text rendering, world knowledge, spatial reasoning, and all other evaluation categories.

Table 4: Quantitative results on UniGenBench++. Entries show the percentage of images judged by the evaluation model to satisfy each criterion. The standard deviation is computed across the 4 images generated for every prompt. N represents the number of distinct samples/particles used during generation.

Method	Mean	Style	World Knowledge	Attribute	Action	Relationship	Compound	Grammar	Logical Reasoning	Entity Layout	Text Generation	NFE
Diffusions												
FLUX.1 [dev]	61.78 ± 3.58	85.20 ± 2.30	86.23 ± 2.62	64.53 ± 2.82	60.46 ± 3.80	63.07 ± 5.46	41.24 ± 5.33	60.96 ± 2.95	24.08 ± 3.63	67.16 ± 4.32	30.17 ± 5.77	180
Flow Map	58.16 ± 0.66	79.70 ± 2.42	80.22 ± 1.04	59.51 ± 1.26	57.22 ± 1.44	62.44 ± 0.95	42.01 ± 2.20	57.62 ± 1.33	25.92 ± 2.37	70.90 ± 1.75	3.74 ± 0.50	16
Gradient-Free Search												
FLUX.1 [dev] + Best-of-N												
- $N = 8$	65.40 ± 2.56	87.40 ± 2.09	89.72 ± 1.75	68.27 ± 2.50	62.26 ± 3.64	68.02 ± 4.38	47.29 ± 1.80	63.37 ± 2.16	29.36 ± 3.24	71.08 ± 71.08	35.06 ± 5.66	360
- $N = 16$	67.57 ± 1.53	88.60 ± 1.40	90.82 ± 1.30	69.44 ± 1.37	64.92 ± 2.33	70.81 ± 3.10	52.71 ± 2.48	64.97 ± 1.10	31.19 ± 2.34	72.95 ± 2.26	36.49 ± 4.01	720
- $N = 32$	69.18 ± 0.87	89.80 ± 1.40	90.82 ± 1.30	72.33 ± 1.22	66.06 ± 1.30	72.97 ± 1.98	54.64 ± 1.59	65.24 ± 1.65	31.88 ± 1.00	76.49 ± 1.54	39.37 ± 4.55	1440
- $N = 64$	70.93 ± 1.38	90.50 ± 0.59	90.98 ± 0.82	74.15 ± 1.37	67.97 ± 0.73	75.00 ± 0.66	56.96 ± 4.13	65.24 ± 3.00	38.30 ± 4.12	77.99 ± 1.71	42.82 ± 3.29	2880
- $N = 128$	72.77 ± 1.26	91.40 ± 0.66	91.77 ± 0.45	75.00 ± 1.56	70.15 ± 1.38	74.24 ± 2.36	60.95 ± 2.20	68.85 ± 2.49	41.97 ± 2.86	80.60 ± 1.40	44.54 ± 5.71	5760
- $N = 256$	73.75 ± 1.06	91.10 ± 0.59	92.88 ± 0.52	76.39 ± 0.82	71.58 ± 1.83	75.51 ± 1.54	61.86 ± 3.44	69.52 ± 0.53	45.18 ± 3.14	80.97 ± 1.54	44.83 ± 3.54	11520
FLUX.1 [schnell] + Best-of-N												
- $N = 4$	67.37 ± 3.74	89.40 ± 2.27	89.40 ± 2.91	67.31 ± 5.10	65.97 ± 3.12	70.69 ± 3.97	52.06 ± 6.57	63.37 ± 3.24	35.09 ± 4.27	78.92 ± 3.35	26.44 ± 7.67	16
- $N = 8$	71.25 ± 1.87	91.30 ± 2.37	92.56 ± 0.69	72.44 ± 2.43	69.77 ± 3.83	75.00 ± 0.66	57.86 ± 3.04	66.58 ± 2.22	39.68 ± 1.64	82.09 ± 2.74	30.46 ± 1.72	32
- $N = 16$	73.59 ± 1.83	91.90 ± 1.34	93.83 ± 0.94	74.79 ± 3.11	73.95 ± 2.83	76.52 ± 2.77	61.73 ± 2.17	67.91 ± 1.96	43.35 ± 2.70	85.63 ± 1.33	31.32 ± 2.49	64
- $N = 32$	75.18 ± 1.13	92.70 ± 1.58	94.62 ± 1.05	77.35 ± 1.09	74.52 ± 1.61	79.95 ± 2.05	63.92 ± 1.96	69.65 ± 1.28	46.10 ± 0.76	85.26 ± 1.62	32.76 ± 4.49	128
- $N = 64$	75.88 ± 0.92	93.50 ± 0.91	95.41 ± 0.52	77.46 ± 2.08	**75.10 ± 0.95**	81.22 ± 2.46	66.11 ± 1.56	69.39 ± 2.12	45.64 ± 4.07	85.45 ± 1.94	34.77 ± 2.62	256
- $N = 128$	76.89 ± 0.51	**94.50 ± 0.89**	94.62 ± 1.14	78.42 ± 1.26	**75.10 ± 1.97**	82.36 ± 1.50	67.78 ± 1.73	71.39 ± 1.23	50.00 ± 2.64	86.94 ± 2.01	33.33 ± 3.35	512
- $N = 256$	**77.41 ± 1.26**	93.40 ± 1.73	**94.94 ± 1.73**	**79.17 ± 1.67**	75.00 ± 1.24	**83.63 ± 1.31**	**68.56 ± 0.36**	71.79 ± 2.52	51.15 ± 4.22	**88.62 ± 2.26**	35.63 ± 3.54	1024
Flow Map + Multi-best-of-N (10 rounds)												
- $N = 8$	67.58 ± 0.56	86.70 ± 0.77	90.66 ± 1.22	70.41 ± 1.06	67.40 ± 1.87	71.32 ± 2.05	52.32 ± 1.39	64.71 ± 2.11	38.30 ± 3.46	76.49 ± 1.63	18.39 ± 0.81	320
- $N = 16$	68.01 ± 1.05	85.90 ± 1.80	88.29 ± 1.05	72.33 ± 1.97	67.97 ± 1.24	71.57 ± 1.52	52.84 ± 2.30	67.38 ± 2.30	38.76 ± 3.14	77.05 ± 1.43	18.10 ± 2.35	640
- $N = 32$	70.53 ± 0.80	89.20 ± 0.49	91.14 ± 1.85	74.04 ± 1.67	70.06 ± 2.99	73.73 ± 2.57	55.93 ± 0.85	70.32 ± 1.79	39.22 ± 1.19	79.29 ± 2.44	22.99 ± 2.93	1280
- $N = 64$	70.90 ± 0.77	88.80 ± 1.67	92.56 ± 2.21	73.18 ± 1.49	70.63 ± 1.53	73.98 ± 2.34	57.73 ± 2.80	69.79 ± 3.04	41.74 ± 2.47	79.29 ± 3.23	23.28 ± 2.21	2560
- $N = 128$	72.60 ± 0.63	90.40 ± 1.50	91.46 ± 0.32	75.75 ± 1.33	70.91 ± 0.87	79.70 ± 2.06	60.57 ± 1.73	71.93 ± 1.79	42.89 ± 3.00	78.73 ± 2.87	23.85 ± 1.49	5120
- $N = 256$	73.47 ± 0.36	90.00 ± 1.47	93.04 ± 1.17	77.35 ± 1.24	75.00 ± 2.55	76.27 ± 0.92	61.73 ± 2.16	72.46 ± 2.16	46.56 ± 3.00	79.66 ± 2.14	21.55 ± 2.97	10240
FK-Steering - $N = 64$												
- 1-step denoiser look-ahead	66.13 ± 0.45	87.70 ± 0.91	87.82 ± 0.94	66.88 ± 1.83	63.69 ± 0.99	70.05 ± 1.87	48.97 ± 1.50	63.77 ± 1.79	32.34 ± 2.28	71.83 ± 2.00	38.22 ± 2.86	2880
- 1-step flow map look-ahead	68.81 ± 0.21	88.80 ± 1.20	89.24 ± 1.34	71.26 ± 2.17	68.35 ± 1.48	71.19 ± 1.81	52.96 ± 0.99	67.11 ± 3.22	32.80 ± 3.00	75.93 ± 1.62	36.78 ± 1.82	3264
Gradient-Based Search												
DAS - $N = 64$												
- 1-step denoiser look-ahead	66.42 ± 1.02	86.90 ± 1.34	87.66 ± 0.95	68.38 ± 0.91	64.35 ± 2.05	69.29 ± 1.37	52.32 ± 3.02	63.90 ± 2.52	33.72 ± 0.76	72.01 ± 2.66	32.76 ± 1.72	2880
- 1-step flow map look-ahead	68.14 ± 0.86	88.50 ± 0.91	89.72 ± 0.82	70.09 ± 1.38	66.25 ± 1.83	70.05 ± 2.41	54.16 ± 1.88	63.77 ± 1.03	33.26 ± 2.63	73.51 ± 1.24	34.77 ± 2.49	1920
FMTT - 1-step denoiser look-ahead												
- $N = 4$	62.00 ± 4.93	85.00 ± 3.26	84.02 ± 4.79	66.88 ± 4.63	60.08 ± 4.79	63.96 ± 7.57	42.91 ± 5.79	58.96 ± 3.99	24.77 ± 5.31	69.59 ± 7.13	28.16 ± 6.32	180
- $N = 8$	65.73 ± 1.83	87.70 ± 1.21	88.29 ± 3.30	69.12 ± 2.27	62.83 ± 1.33	69.04 ± 2.24	47.81 ± 3.54	62.83 ± 3.04	30.05 ± 2.78	72.57 ± 0.97	34.20 ± 1.70	360
- $N = 16$	67.75 ± 1.82	88.20 ± 0.87	88.92 ± 1.14	70.09 ± 2.16	65.49 ± 1.33	71.70 ± 2.92	49.23 ± 3.45	64.57 ± 3.77	32.57 ± 3.52	76.49 ± 3.05	40.80 ± 4.34	720
- $N = 32$	69.79 ± 1.06	90.00 ± 1.30	90.51 ± 1.55	71.90 ± 2.35	67.30 ± 1.17	73.35 ± 1.16	52.71 ± 1.56	67.38 ± 1.81	37.84 ± 5.44	77.05 ± 2.07	39.94 ± 1.49	1440
- $N = 64$	70.89 ± 1.32	90.20 ± 1.54	90.66 ± 1.51	73.72 ± 0.88	68.92 ± 2.89	73.73 ± 2.60	56.06 ± 1.80	66.58 ± 1.58	37.61 ± 4.05	78.36 ± 3.03	43.97 ± 2.21	2880
- $N = 128$	72.52 ± 1.42	91.60 ± 0.85	92.56 ± 0.27	73.40 ± 2.82	69.49 ± 2.75	76.27 ± 1.58	58.12 ± 2.92	69.52 ± 2.30	42.20 ± 5.07	81.53 ± 2.55	42.24 ± 0.95	5760
- $N = 256$	72.90 ± 0.89	91.00 ± 1.31	90.51 ± 1.85	73.93 ± 1.98	70.72 ± 1.57	78.30 ± 0.91	60.31 ± 0.82	69.65 ± 3.37	42.89 ± 4.12	81.53 ± 1.70	39.94 ± 1.88	11520
FMTT - 1-step flow map look-ahead												
- $N = 16$	63.28 ± 0.77	85.10 ± 0.95	85.28 ± 1.57	67.09 ± 2.58	60.93 ± 1.55	67.13 ± 1.31	45.36 ± 3.75	61.90 ± 1.33	27.06 ± 1.38	69.22 ± 1.43	27.87 ± 3.39	240
- $N = 32$	67.55 ± 0.86	88.60 ± 0.82	88.92 ± 1.05	68.06 ± 1.69	65.97 ± 1.46	70.94 ± 0.83	51.93 ± 1.17	65.37 ± 0.95	31.65 ± 3.70	75.37 ± 3.34	36.49 ± 4.33	480
- $N = 64$	71.13 ± 0.62	89.60 ± 0.63	91.14 ± 0.45	73.40 ± 2.17	67.87 ± 2.26	75.25 ± 1.98	57.35 ± 1.52	68.72 ± 1.34	37.39 ± 3.39	77.05 ± 2.66	45.11 ± 3.39	960
- $N = 128$	73.32 ± 0.68	91.90 ± 0.52	90.98 ± 0.94	76.39 ± 2.01	71.29 ± 1.46	75.63 ± 1.61	59.79 ± 1.67	69.65 ± 0.95	42.43 ± 1.76	80.04 ± 1.93	46.84 ± 2.86	1920
- $N = 256$	74.86 ± 0.50	91.90 ± 0.77	92.72 ± 1.30	77.78 ± 1.89	71.86 ± 0.60	77.41 ± 2.17	62.50 ± 1.76	72.59 ± 1.53	46.33 ± 1.38	81.72 ± 2.14	46.55 ± 4.10	3840
FMTT - 1-step Schnell look-ahead												
- $N = 256$	77.00 ± 0.29	92.40 ± 0.49	93.67 ± 1.00	78.95 ± 0.70	74.43 ± 0.87	80.20 ± 0.88	66.11 ± 2.04	**74.87 ± 1.77**	**51.38 ± 4.35**	84.14 ± 1.33	47.70 ± 4.63	3840
FMTT - 2-step flow map look-ahead												
- $N = 16$	64.73 ± 0.74	87.40 ± 2.09	87.34 ± 2.15	65.60 ± 2.35	61.79 ± 0.83	68.27 ± 1.81	47.94 ± 1.63	61.90 ± 1.38	30.50 ± 0.76	71.27 ± 3.10	33.33 ± 2.15	360
- $N = 32$	68.04 ± 1.34	89.60 ± 0.57	90.35 ± 0.82	70.19 ± 2.55	64.83 ± 4.10	69.80 ± 1.16	51.80 ± 1.06	66.84 ± 2.33	33.26 ± 4.27	73.69 ± 2.50	39.08 ± 3.45	840
- $N = 64$	71.46 ± 0.40	90.10 ± 1.07	91.46 ± 2.12	74.15 ± 1.11	68.25 ± 2.49	74.49 ± 1.50	56.70 ± 2.03	69.65 ± 1.28	38.76 ± 1.88	77.99 ± 2.87	44.83 ± 1.63	1680
- $N = 128$	74.31 ± 0.78	91.20 ± 1.20	92.56 ± 0.82	74.79 ± 2.28	71.48 ± 2.81	78.05 ± 0.55	62.89 ± 1.26	72.46 ± 1.26	46.33 ± 2.00	80.78 ± 2.32	45.98 ± 2.44	3360
- $N = 256$	75.55 ± 0.53	91.90 ± 0.77	92.88 ± 1.51	78.21 ± 1.09	72.34 ± 1.12	79.19 ± 1.87	65.08 ± 1.98	72.73 ± 0.85	46.79 ± 1.45	81.53 ± 1.10	47.70 ± 3.09	6720
FMTT - 4-step flow map look-ahead												
- $N = 16$	65.90 ± 0.45	86.90 ± 1.21	84.34 ± 1.64	68.27 ± 1.55	63.97 ± 1.43	69.92 ± 1.81	49.23 ± 1.65	63.90 ± 0.96	31.42 ± 1.76	75.37 ± 0.53	32.47 ± 1.25	600
- $N = 32$	68.12 ± 0.62	89.00 ± 1.00	87.03 ± 2.22	70.51 ± 1.17	65.02 ± 2.56	71.07 ± 1.83	52.19 ± 1.33	66.84 ± 0.38	34.17 ± 1.00	74.81 ± 2.14	40.52 ± 4.25	1560
- $N = 64$	71.48 ± 0.29	89.80 ± 0.72	90.82 ± 0.78	72.86 ± 0.48	69.39 ± 0.69	74.49 ± 2.19	56.83 ± 2.20	69.12 ± 0.95	40.14 ± 2.78	78.92 ± 0.97	43.97 ± 2.05	3120
- $N = 128$	74.32 ± 0.32	90.90 ± 0.71	90.55 ± 0.78	78.31 ± 0.56	70.63 ± 0.78	76.65 ± 1.19	60.44 ± 1.72	74.20 ± 1.38	46.56 ± 3.00	80.22 ± 0.83	49.43 ± 2.82	6240
- $N = 256$	75.14 ± 0.80	91.40 ± 1.18	92.41 ± 1.18	77.03 ± 1.72	71.39 ± 2.16	79.19 ± 1.48	65.08 ± 3.29	72.46 ± 2.32	45.87 ± 2.25	80.41 ± 1.10	**50.86 ± 2.62**	12480

E Broader Impact

Flow Map Trajectory Tilting contributes to reward-aligned generation with diffusion models, enabling stronger results without retraining and potentially reducing the computational cost of adapting models to new objectives. While we focus on text-to-image generation, the framework is broadly applicable to other modalities wherever suitable reward feedback can be defined. However, generative models and reward-guided methods can be misused for producing deceptive imagery or optimizing toward harmful objectives, and may inherit biases from reward models. Such techniques should be deployed with appropriate safeguards and caution.

F Limitations

As discussed in the main paper, FMTT is a gradient-based method and therefore requires a differentiable reward function. As a result, it can be slower than purely gradient-free methods under the same hyperparameters, and it is not directly applicable to non-differentiable rewards. In addition, flow map look-ahead requires access to a pretrained flow map or a few-step distilled model, which may introduce an additional distillation cost when such a model is not already available. Finally, in text-to-image experiments, FMTT can require up to several thousand function evaluations, which is orders of magnitude more expensive than sampling from the base flow map model alone. A promising direction for future work is to combine FMTT with online RL-based diffusion fine-tuning methods, which could amortize this test-time cost.

G Licenses

Our models and code are built upon the following codebases and datasets:

- **Diffusers** (https://github.com/huggingface/diffusers): Used for our text-to-image experiments. Licensed under the Apache License 2.0. We build on top of `FLUX.1-dev`, which is under a proprietary non-commercial license (https://huggingface.co/black-forest-labs/FLUX.1-dev/blob/main/LICENSE.md).
- **GenEval** (https://github.com/djghosh13/geneval): Used as a validation benchmark. Licensed under the MIT license.
- **UniGenBench++** (https://github.com/CodeGoat24/UniGenBench): Used as a validation benchmark. Licensed under the CC BY 4.0 license.
- **Skywork-VL** (https://huggingface.co/Skywork/Skywork-VL-Reward-7B): Used as a VLM-based reward model. Licensed under the MIT license.
- **Qwen2.5-VL** (https://huggingface.co/Qwen/Qwen2.5-VL-7B-Instruct): Used as a VLM-based reward model. Licensed under the Apache 2.0 license.
- **Pickscore** (https://github.com/yuvalkirstain/pickscore): Used as a human preference reward model. Licensed under the MIT license.
- **HPSv2** (https://github.com/tgxs002/HPSv2): Used as a human preference reward model. Licensed under the Apache 2.0 license.
- **ImageReward** (https://github.com/zai-org/ImageReward): Used as a human preference reward model. Licensed under the Apache 2.0 license.
- **CLIP** (https://huggingface.co/laion/CLIP-ViT-H-14-laion2B-s32B-b79K): Used as a text-image alignment reward model. Licensed under the MIT license.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We summarize the contributions of our work in the introduction and the abstract. We conduct comprehensive experiments and analytical derivations to support all claims and proofs.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss the limitations of our approach in a dedicated section in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: We give complete proofs for our theoretical results in the appendix.

Guidelines:

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

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We share all implementation details needed to replicate our results in the method section and the appendix.

   Guidelines:

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

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We don't plan to release the code.

Guidelines:

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

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: We describe all training and sampling details, including all hyperparameters, in our method and experiments sections as well as the appendix.

Guidelines:

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

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We follow standard practice of the field of reporting accuracy scores in a table format which do not involve error bars.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide our hardware specifications and report our number of function evaluations (NFE) in our experiments section and the Appendix.

   Guidelines:

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

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our work fully confirms with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We include a discussion on broader impact in the Appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [No]

    Justification: Our paper builds upon publicly released text-to-image models, which come with their own set of safeguards.

    Guidelines:

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

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification:We cite all research works be build on and use the code of others according to their licenses. We list all licenses in the Appendix.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: We plan to release our code for our ImageNet experiments after publication. This will include a complete README with detailed instructions and license.

    Guidelines:

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

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: We don't perform any experiments with human subjects.

    Guidelines:

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

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: We de not perform any research on human subjects that may have any potential risks.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: We do not make use of LLMs for developing our core methods and results.

    Guidelines:

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