TEST-TIME SCALING OF DIFFUSIONS WITH FLOW MAPS

Anonymous authors
Paper under double-blind review

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 a 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 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 lookahead 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.

Figure 1: Test-time search can overcome model biases and reliably sample from regions of the distribution (e.g., precise clock times) that baselines fail to capture.

1 INTRODUCTION

Large scale foundation models built out of diffusions (Ho et al., 2020; Song et al., 2020) or flow-based transport (Lipman et al., 2022; Albergo & Vanden-Eijnden, 2022; Albergo et al., 2023; Liu et al., 2022) have become highly successful tools across computer vision and scientific domains. In this paradigm, performing generation amounts to numerically solving an ordinary or stochastic differential equation (ODE/SDE), the coefficients of which are learned neural networks. An active area of current research is how to best *adapt* these dynamical equations at inference time to extract samples from the model that align well with a user-specified reward. For example, as shown in Figure 1, a user may want to generate an image of a clock with a precise time displayed on it, which is often generated inaccurately without suitable adaptation of the generative process. These approaches, often collectively referred to as *guidance*, do not require additional re-training and as a result are orthogonal to the class of fine-tuning methods, which instead attempt to adjust the model itself via an additional learning procedure 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 a one-step denoiser that can be derived from a learned score or flow-based model.

In this paper, we revisit the reward guidance problem from the perspective of flow maps, a recently-introduced methodology for flow-based generative modeling that learns the solution operator of a probability flow ODE directly rather than the associated drift (Boffi et al., 2024; 2025; Sabour et al., 2025). 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 predict the terminal point of a trajectory in a single or a few function evaluations, vastly improving the prediction relative to denoiser-based techniques 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 effectively uses their look-ahead capabilities to accurately incorporate learned and complex reward functions in Monte Carlo and search algorithms. (*ii*) Using conditions that characterize the flow map, we show that the importance weights for this Jarzynski/SMC scheme reduce to a remarkably simple formula. Our approach is theoretically grounded in controlling the *thermodynamic length* of the process over baselines, a measure of the efficiency of the guidance in sampling the tilted distribution. (*iii*) We empirically show that FMTT has favorable test-time scaling characteristics that outperform standard ways of embedding rewards into diffusion sampling setups. (*iv*) To our knowledge, we demonstrate the first successful use of pretrained vision-language models (VLMs) as reward functions for test-time scaling, allowing rewards to be specified entirely in natural language. We further show that the flow map is crucial for their success, substantially boosting the effectiveness of the search process when using these rewards.

1.1 Related work

Flows and diffusions. Diffusion models (Song et al., 2020; Ho et al., 2020) and flow models (Lipman et al., 2022; Albergo & Vanden-Eijnden, 2022; Liu et al., 2022) 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 map (Song et al., 2023; Kim et al., 2024; Boffi et al., 2024; Geng et al., 2025; Sabour et al., 2025; Boffi et al., 2025), 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 tradeoff compute at inference time to improve the performance of a model or align the generation with a user specified reward (Ma et al., 2025). Certain works use the denoiser associated with the score model to perform this look-ahead on the dynamics (Wu et al., 2024; Singhal et al., 2025; Zhang et al., 2025). 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 (Lee et al., 2025; Ramesh & Mardani, 2025), 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.

2 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 retrain-

ing) 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.

2.1 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) of the form

$$\dot{x}_t = b_t(x_t) \qquad x_{t=0} \sim \rho_{t=0}, \tag{1}$$

where $b_t : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ is a velocity field that governs the transport and is adjusted so that the solutions to the ODE (1) satisfy $x_{t=1} \sim \rho_1$. Since the time dependent PDF $\rho_t(x)$ of the solutions to (1) at time t satisfies the continuity equation

$$\partial_t \rho_t = -\nabla \cdot (b_t \rho_t) \qquad \rho_{t=0} = \rho_0 \tag{2}$$

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

$$X_{s,t}(x_s) = x_t \qquad \forall s, t \in [0,1]. \tag{3}$$

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

$$\partial_s X_{s,t}(x) + b_s(x) \cdot \nabla X_{s,t}(x) = 0, \tag{4}$$

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 the 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 (Albergo & Vanden-Eijnden, 2022; Albergo et al., 2023; Lipman et al., 2022; Liu et al., 2022), 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 for simplicity. Importantly, using these coefficients, the law of this process satisfies (2) with the velocity field

$$b_t(x) = \mathbb{E}[\dot{I}_t | I_t = x] = \mathbb{E}[x_1 | I_t] - \mathbb{E}[x_0 | I_t], \tag{5}$$

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

$$s_t(x) = (t b_t(x) - x)(1-t)^{-1}. \tag{6}$$

The velocity field $b_t(x)$ is also the minimizer of a simple quadratic objective (Lipman et al., 2022; Albergo & Vanden-Eijnden, 2022) 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

$$dx_t = [b_t(x_t) + \epsilon_t s_t(x_t)]\, dt + \sqrt{2\epsilon_t} dW_t \tag{7}$$

where $\epsilon_t \geq 0$ is an arbitrarily tunable diffusion coefficient and dW_t is an incremental Brownian motion (Albergo et al., 2023). 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

$$\partial_t \rho_t = -\nabla \cdot (b_t \rho_t) + \epsilon_t \nabla \cdot [-s_t \rho_t + \nabla \rho_t] \tag{8}$$

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 (Kim et al., 2024; Boffi et al., 2025)

$$\lim_{s \to t} \partial_t X_{s,t}(x) = b_t(x), \tag{9}$$

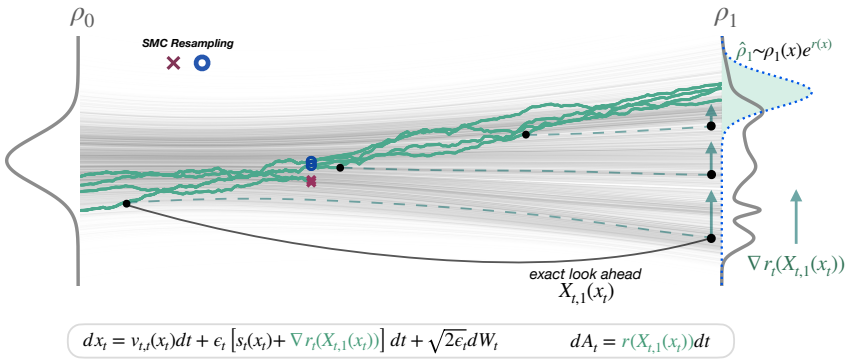

Figure 2: Schematic overview of test-time adaptation of diffusions with flow map tilting. Using the look-ahead map $X_{t,1}(x_t)$ in the diffusion inside the reward, reward information can be principly used through the tilted trajectories (green lines). This allows us to perform better ascent on the 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 $\hat{\rho}_t$.

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

$$X_{s,t}(x) = x + (t - s)v_{s,t}(x), \tag{10}$$

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)*:*

$$v_{t,t}(x) = b_t(x). \tag{11}$$

As such, training a flow map model instead of just a velocity model 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.

2.2 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 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 *track local maximizers* of $\hat{\rho}_t$ for example in image generation procedures. This is useful for ensuring 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

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

To implement this change in practice, however, we face 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 look-ahead:** This amounts to using e.g. $r_t(x) = tr(x)$. 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, instead of $r_t(x) = tr(x)$, we could instead use $r_t(x) = tr(\mathsf{D}(x))$. 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. **Flow map look-ahead:** Intuitively, the above dynamics are better if one works instead with the flow map defined in the previous section. Because the flow 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 $\lim_{s\to t} \partial_t X_{s,t}(x) = v_{t,t}(x)$, we can instead sample with the following SDE:

Figure 3: Qualitative results using VLM-based rewards. Prompts where the base model fails to generate aligned outputs are corrected by FMTT, with flowmap look-ahead producing the most reliable improvements.

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

where $r_t(x) = tr(X_{t,1})(x)$ makes use of the *exact look-ahead* to properly evaluate the reward for any t, even $t = 0$. The above could be interpreted as a continuous deformation of how the 1-step flow map would evolve under ascent on the reward.

2.3 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 to fulfill this. This gives rise to the second issue, for any version of $r_t(x)$, with or without the 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 $\partial_t \hat{\rho}_t + \nabla \cdot (b_t \hat{\rho}_t)$:

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

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 can 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 (Jarzynski, 1997; Vaikuntanathan & Jarzynski, 2008):

Proposition 2.1 (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*

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

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 $\tilde{x} = (\tilde{x}_t)_{t \in [0,T]}$.

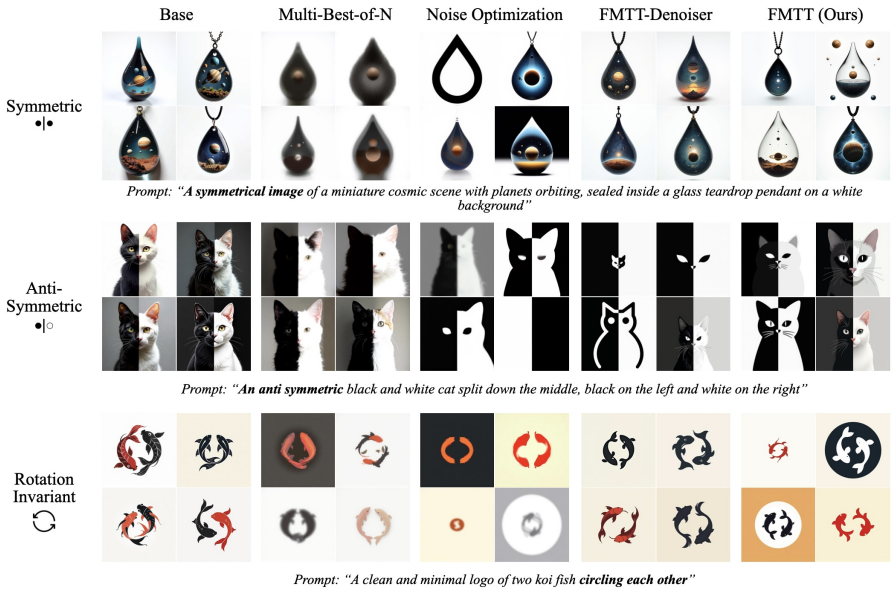

Figure 4: Qualitative comparison on three basic geometric rewards (symmetry, anti-symmetry, rotation invariance). The gradient-based methods that change the generative dynamics produce sharper images that satisfy the constraints more reliably than prior methods.

The proof of this statement in Appendix 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 2. 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 we use as $r_t(x)$ the reward composed with the flow map, as stated in the following proposition

Proposition 2.2 (Unbiased Flow Map Trajectory Tilting)**.** *Using the same notations as in Proposition 2.1, if $r_t(x) = t\, r(X_{t,1}(x))$, then the importance weights defined (17) reduce to*

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

This result is proven in Appendix A.2, and relies on a simple modification of the proof of Proposition 2.1 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 3.2 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

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

where $\eta_t = (1-t)/t$, and the weights with

$$d\tilde{A}_t = \left[\frac{r_t(\tilde{x}_t)}{t} + \eta_t\big(\|\nabla r_t\|^2 + \nabla r_t \cdot s_t\big)(\tilde{x}_t)\right] dt + \frac{\eta_t}{\sqrt{2\epsilon_t}} \nabla r_t(\tilde{x}_t) \cdot \big(\mathrm{d}^- W_t - \mathrm{d}W_t\big) \tag{21}$$

where $\mathrm{d}^- W_t$ is the backward Itô differential and $\tilde{A}_0 = 0$. It is proven in Appendix A.2 that these equations also provide an unbiased sampler of the tilted distribution by ensuring (18) holds. This further augments the sampling process toward the tilt and will prove useful in the experiments below.

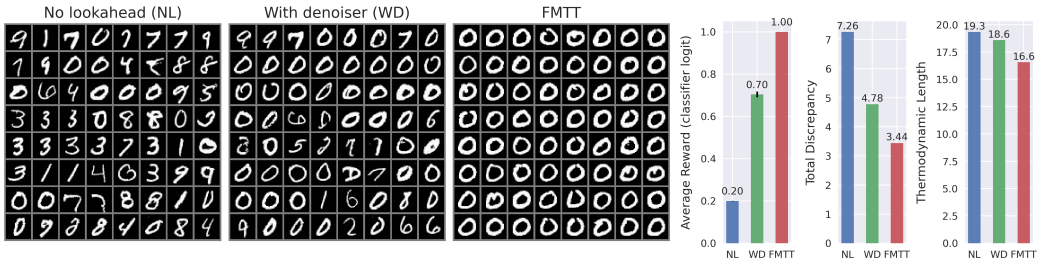

Figure 5: Comparison of MNIST tilted sampling to generate digits that would be classified as zeros. **Left**: using (20) with no look-ahead. **Center:** Doing the same with the denoiser composed with the reward. **Right**: Doing the same with the flow map i.e. our method FMTT. FMTT most consistently generates zeros, has the lowest total discrepancy, and the smallest thermodynamic length.

Characterizing the effectiveness of the flow map trajectory tilting As discussed above, the dynamics (13)-(19) and (20)-(21) are simulated via SMC, which involves a system of N particles, a time discretization $(t_k)_{k=1}^K$ and a (random) number of resampling steps $R \leq K$. SMC algorithms naturally yield an unbiased estimate $\hat{Z}_{\text{SMC}}$ of the normalization constant $\mathbb{E}[e^{A_t}]$, and a low variance $\text{Var}[\hat{Z}_{\text{SMC}}]$ is a proxy for an efficient sampling schedule, as it signals that the empirical distribution of the N particles is close to $\hat{\rho}_1$. However, $\text{Var}[\hat{Z}_{\text{SMC}}]$ often takes exponentially high values, making it hard to approximate, and it depends on N, K, and R. The following proposition introduces the **thermodynamic length**, a quantity related to $\text{Var}[\hat{Z}_{\text{SMC}}]$ which does not suffer from these issues.

Proposition 2.3 (Total discrepancy and thermodynamic length, informal)**.** *The variance* $\text{Var}[\hat{Z}_{\text{SMC}}]$ *can be expressed in terms of the number of particles* N*, discretization steps* K*, and resampling steps* R*, and the* total discrepancy $D(\mathcal{T})$ *which depends on discretization schedule* $\mathcal{T} = (t_k)_{k=0}^K$ *and is computable in practice. For optimal* $\mathcal{T}$*,* $K\sqrt{D(\mathcal{T})}$ *can be replaced by* thermodynamic length Λ*, which can be computed from the* $\tilde{A}_t$ *updates and that is agnostic to* $\mathcal{T}$*,* N *and* R*, and satisfies that* $\Lambda \leq K\sqrt{D(\mathcal{T})}$.

From sampling to search: making the most of rewards in practice. Notably, the importance weights defined in either (19) or (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 we'd usually do in SMC. Unlike (Ma et al., 2025), this use of top-n still makes use of the gradient of the reward along the trajectory, while also making use of better signal thanks to the flow map.

3 Numerical Experiments

Algorithm 1 details all the inference-time adaption techniques that we introduce: The base algorithm described in Proposition 2.2 corresponds to the choice $\epsilon_k = \epsilon_{t_k}$ and $\eta_k = 0$ for all $k = 0 : K$, and Sampling $=$ True.The reward-modified algorithm described in Proposition A.1 amounts to setting $\epsilon_k = \epsilon_{t_k}$ and $\eta_k = \eta_{t_k} = \beta_{t_k}(\frac{\dot{\alpha}_{t_k}}{\alpha_{t_k}}\beta_{t_k} - \dot{\beta}_{t_k})$ for all $k = 0 : K$, and Sampling $=$ True.

3.1 Thermodynamic length control on MNIST

To demonstrate that the thermodynamic length is a meaningful diagnostic of the performance of the tilt, we compute the thermodynamic length on the problem of tilting an unconditional image generation model to a class conditional one. This will allow us to show that the process driven by FMTT more efficiently samples the tilted distribution of interest. For sake of expediency to make the computation of the thermodynamic discrepancies and length calculable, we measure and compare these quantities on an MNIST experiment where the reward model is for a classifier to assign high likelihood to the unconditionally generated image being a zero. In Figure 5 we compare the three setups specified below (12) to show that higher performance on the classification task aligns with lower total discrepancy and lower thermodynamic length. This is precisely what we see, showing that FMTT, our flow map method using (13) and the weights (19) in an SMC procedure, achieves perfect classification accuracy along with the lowest total discrepancy and thermodynamics length.

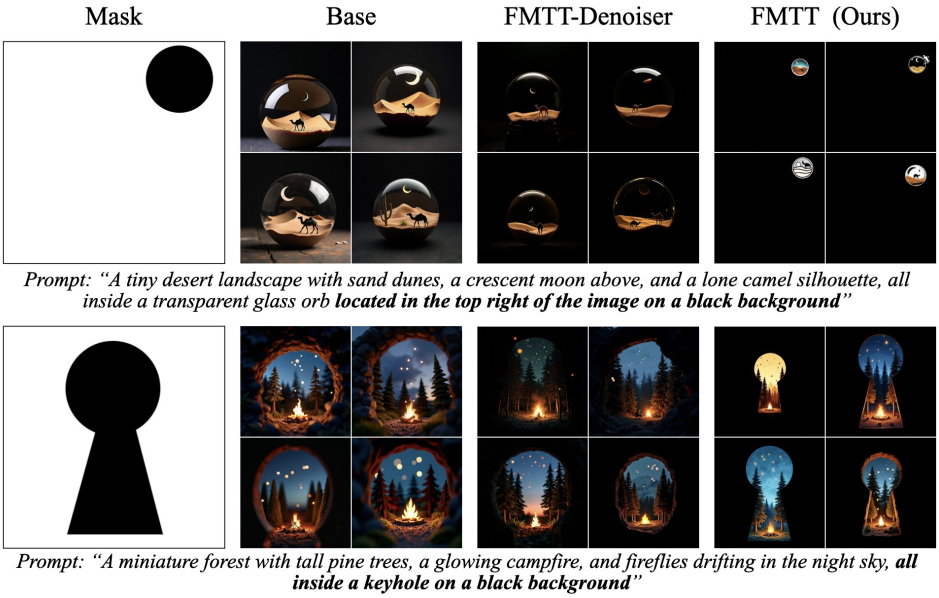

Figure 6: Qualitative comparison on masked rewards. Only our flow map-based FMTT reliably satisfies the constraints, concentrating content in the unmasked regions.

3.2 Text-to-Image Experiments

We evaluate our approach on text-to-image generation using the 4-step distilled flow map from Align Your Flow (Sabour et al., 2025), trained by distilling the open-source FLUX.1-dev model (Labs, 2024). We consider three categories of reward functions: 1) *Human preference rewards* capturing visual quality and text alignment, 2) *Geometric rewards* enforcing structural constraints such as symmetry or rotation invariance, 3) *VLM-based rewards* defined through natural language queries.

As baselines, we compare against *gradient-free* and *gradient-based* methods. Gradient-free approaches such as Best-of-N (Chatterjee & Diaconis, 2018), Multi-Best-of-N (Lee et al., 2025), and beam search (Fernandes et al., 2025) 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 (Eyring et al., 2024) performs gradient ascent in the initial noise latent space, keeping samples tied to the base distribution, whereas our FMTT algorithm (and its ablations) use the gradient to modify the generative process itself, enabling exploration beyond the model's support and generation of out-of-distribution samples. Notably, the gradient of the reward used in FMTT efficiently gives meaningful signal for the whole trajectory thanks to the flow map, enably OOD sample generation for highly nuanced rewards.

Human Preference Rewards. To quantitatively benchmark FMTT, we follow prior work (Eyring et al., 2024) and use a linear combination of PickScore (Kirstain et al., 2023), HPSv2 (Wu et al., 2023), ImageReward (Xu et al., 2023), and CLIPScore (Radford et al., 2021) as the reward and perform evaluation on GenEval (Ghosh et al., 2023), which consists of ≈550 object-centric prompts and measures the quality of generated images using a pre-trained object detector. Results in Table 1.

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 flowmap slightly reduces performance but significantly accelerates generation. A simple best-of-N search on top of the flowmap 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. It is important to note that the base FLUX model has already been post-trained with human preference data. As a result, optimizing for the same types of reward during inference cannot substantially shift its output distribution, which explains why improvements across methods remain limited.

Finally, we ablate the use of the 4-step flowmap look-ahead by comparing FMTT against variants using either a 1-step denoiser or a 4-step diffusion sampler. The flowmap look-ahead consistently performs best, in line with our earlier findings.

Table 1: Quantitative results on GenEval.

Method	Mean ↑	Single Obj. ↑	Two Obj. ↑	Counting ↑	Colors ↑	Position ↑	Attr. Binding ↑	NFE
Diffusions + Flowmaps								
FLUX.1 [dev]	0.65	0.99	0.78	0.70	0.78	0.18	0.45	180
Flowmap	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
Flowmap + Best-of-N	0.73	1.00	0.88	0.82	0.85	0.25	0.59	128
Flowmap + Multi-best-of-N	0.76	1.00	0.95	0.84	0.85	0.26	0.69	1280
Flowmap + Beam Search	0.75	1.00	0.92	0.86	0.85	0.29	0.58	1200
Gradient-Based Search								
Flowmap + 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 lookahead	0.75	0.99	0.90	0.87	0.87	0.26	0.59	350
FMTT - 4-step diffusion lookahead	0.75	0.99	0.93	0.86	0.89	0.27	0.57	1400

Geometric Transformation Rewards. Recall that FLUX.1-dev has already been trained on human preference data, so its output distribution is already biased toward high preference rewards. This explains why much of the improvement in the previous experiments could be achieved with a simple best-of-N search, with a slight additional boost being obtained when using gradient-based methods. This changes when the reward function is more specialized and achieves high values only in the long tails of the base model's output distribution. An example is a reward that enforces invariance under simple geometric transformations, defined as $r(x) = -d(x, T(x))$ where $T(x) : \mathbb{R}^d \to \mathbb{R}^d$ is a transformation function and $d(\cdot, \cdot)$ is a distance metric. For example, if T is a masking operator, this reward incentivizes blackening the masked regions which can be used as a way to position elements in the scene. Similar rewards can be defined for symmetry, anti-symmetry, rotation, and so on.

Figure 4 shows that the base model roughly aligns with these objectives but does not fully satisfy them (the small planets break symmetry, the cats eyes aren't anti-symmetric, and the koi fish have different colors so is not rotation invariant). Prior methods such as multi-best-of-N (Lee et al., 2025) and ReNO (Eyring et al., 2024) also fail, either breaking constraints or producing blurry images. In contrast, our gradient-based variants directly modify the dynamics, producing sharper outputs that more reliably satisfy the constraints. For a harder case, we evaluate the masked reward in Figure 6. FMTT with a denoiser look-ahead produces darker images with higher rewards than the base model, but fails to move all content to the unmasked region. Using a flowmap look-ahead, however, successfully maximizes the reward, generating images that fully satisfy the constraint.

VLMs as a Judge. We explore using pretrained VLMs to judge our images. The setup is straightforward: we provide the generated image along with a binary yes/no question, and define the reward as the difference between the log-probabilities of the answers "Yes" and "No". This formulation allows rewards to be expressed entirely in natural language. 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 (Wang et al., 2025) for single-image settings and Qwen2.5-VL-7B-Instruct (Bai et al., 2025) for multi-image applications.

Qualitative results are shown in Figure 3. The figure highlights two prompts where the base model fails to produce text-aligned outputs. When the VLM is used as a reward to judge whether the prompt is a correct caption for the image, our FMTT algorithm generates outputs that match the input text much more accurately. While FMTT with a denoiser look-ahead improves text alignment in some cases, its success rate is low (only 1/4 images match the prompt). By contrast, FMTT with a flowmap look-ahead consistently produces better-aligned images, such as correctly repeating characters in the book example, and does so with a higher success rate and average final reward. For additional VLM-based experiments, including multi-image settings, please see Appendix D.

The reward here is based on the question: *"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)."*

One caveat is the possibility of reward hacking (Amodei et al., 2016), as the search procedure explicitly maximizes the VLM reward. To mitigate this, the yes/no questions must be written with enough detail to prevent the algorithm from exploiting loopholes. Discussion and examples in Appendix E.

Conclusions. We have presented FMTT, using flow maps for improved test-time scaling of diffusions. We envision that FMTT can overcome the limitations of common image generation systems when nuanced control is required or challenging rewards are given, for instance by VLMs.

REFERENCES

Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *The Eleventh International Conference on Learning Representations*, 2022.

Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

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

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.

Nicholas M. Boffi, Michael S. Albergo, and Eric Vanden-Eijnden. Flow Map Matching: A unifying framework for consistency models. *arXiv:2406.07507*, June 2024.

Nicholas M. Boffi, Michael S. Albergo, and Eric Vanden-Eijnden. How to build a consistency model: Learning flow maps via self-distillation, 2025. URL https://arxiv.org/abs/2505.18825.

Sourav Chatterjee and Persi Diaconis. The sample size required in importance sampling. *The Annals of Applied Probability*, 28(2):1099–1135, 2018.

Nicolas Chopin, Sumeetpal S. Singh, Tomás Soto, and Matti Vihola. On resampling schemes for particle filters with weakly informative observations. *The Annals of Statistics*, 50:3197–3222, 2022.

Chenguang Dai, Jeremy Heng, Pierre E. Jacob, and Nick Whiteley. An invitation to sequential monte carlo samplers. *Journal of the American Statistical Association*, 117:1587 – 1600, 2020.

Luca Eyring, Shyamgopal Karthik, Karsten Roth, Alexey Dosovitskiy, and Zeynep Akata. Reno: Enhancing one-step text-to-image models through reward-based noise optimization. *Neural Information Processing Systems (NeurIPS)*, 2024.

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.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling, 2025. URL https://arxiv.org/abs/2505.13447.

Dhruba Ghosh, Hanna Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *ArXiv*, abs/2310.11513, 2023. URL https://api.semanticscholar.org/CorpusID:264288728.

Roger B Grosse, Chris J Maddison, and Russ R Salakhutdinov. Annealing between distributions by averaging moments. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

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

C. Jarzynski. Nonequilibrium equality for free energy differences. *Phys. Rev. Lett.*, 78:2690–2693, Apr 1997. doi: 10.1103/PhysRevLett.78.2690. URL https://link.aps.org/doi/10.1103/PhysRevLett.78.2690.

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. *arXiv:2310.02279*, 2024.

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. 2023.

Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024.

Gyubin Lee, Truong Nhat Nguyen Bao, Jaesik Yoon, Dongwoo Lee, Minsu Kim, Yoshua Bengio, and Sungjin Ahn. Adaptive inference-time scaling via cyclic diffusion search. 2025. URL https://api.semanticscholar.org/CorpusID:278769684.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2022.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2022.

Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, et al. Inference-time scaling for diffusion models beyond scaling denoising steps. *arXiv preprint arXiv:2501.09732*, 2025.

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*, pp. 8748–8763. PmLR, 2021.

Vignav Ramesh and Morteza Mardani. Test-time scaling of diffusion models via noise trajectory search. *arXiv preprint arXiv:2506.03164*, 2025.

Amirmojtaba Sabour, Sanja Fidler, and Karsten Kreis. Align your flow: Scaling continuous-time flow map distillation. *ArXiv*, abs/2506.14603, 2025. URL https://api.semanticscholar.org/CorpusID:279410235.

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, 2025. URL https://arxiv.org/abs/2501.06848.

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

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency Models. *arXiv:2303.01469*, 2023.

Saifuddin Syed, Alexandre Bouchard-Côté, Kevin Chern, and Arnaud Doucet. Optimised annealed sequential monte carlo samplers. *arXiv:2408.12057*, 2024.

Suriyanarayanan Vaikuntanathan and Christopher Jarzynski. Escorted free energy simulations: Improving convergence by reducing dissipation. *Phys. Rev. Lett.*, 100:190601, May 2008. doi: 10.1103/PhysRevLett.100.190601. URL https://link.aps.org/doi/10.1103/PhysRevLett.100.190601.

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.

Luhuan Wu, Brian L. Trippe, Christian A. Naesseth, David M. Blei, and John P. Cunningham. Practical and asymptotically exact conditional sampling in diffusion models, 2024. URL https://arxiv.org/abs/2306.17775.

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.

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 *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pp. 15903–15935, 2023.

Tao Zhang, Jia-Shu Pan, Ruiqi Feng, and Tailin Wu. Vfscale: Intrinsic reasoning through verifier-free test-time scalable diffusion model. *arXiv preprint arXiv:2502.01989*, 2025.

A PROOFS

A.1 PROOF OF PROPOSITION 2.1

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

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

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 (**??**) 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 2.2

When $r_t(x) = tr(X_{t,1}(x))$, 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

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 (6). Since the score of the tilted distribution at time $t = 1$ is $\hat{s}_1(x) = \nabla \log \hat{\rho}_1(x) = s_1(x) + \nabla r_1(x)$, we obtain that the vector field $\hat{v}_{1,1}(x)$ for the tilted distribution reads

$$\hat{v}_{1,1}(x) = \alpha_1\Big(\frac{\dot{\beta}_1}{\beta_1}\alpha_1 - \dot{\alpha}_1\Big)\hat{s}_1(x) + \frac{\dot{\beta}_1}{\beta_1} x = v_{1,1}(x) + \alpha_1\Big(\frac{\dot{\beta}_1}{\beta_1}\alpha_1 - \dot{\alpha}_1\Big)\nabla r_1(x), \tag{35}$$

and we can define $\hat{v}_{t,t}$ analogously replacing 1 by t. As we show in the following proposition, we can substitute in $\hat{v}_{t,t}$ for $v_{t,t}$ in the SDE (13), and adjust the dynamics of $A^\star$ appropriately such that the same result holds.

Proposition A.1 (Unbiased Map Tilting with reward-modified vector field). *Let $\eta_t = \alpha_t\big(\frac{\dot{\beta}_t}{\beta_t}\alpha_t - \dot{\alpha}_t\big)$ and $r_t(x) = tr(X_{t,1}(x))$, and $\tilde{x}_t, \tilde{A}_t$ be the solution to (20) and (21) respectively. Then for all $t \in [0,1]$ and any test function $h : \mathbb{R}^d \to \mathbb{R}$, we have*

$$\int_{\mathbb{R}^d} h(x)\hat{\rho}_t(x) dx = \frac{\mathbb{E}[e^{\tilde{A}_t} h(\tilde{x}_t)]}{\mathbb{E}[e^{\tilde{A}_t}]}, \tag{36}$$

where the expectations at the right-hand side are taken over the law of $\tilde{x} = (\tilde{x}_t)_{t\in[0,1]}$.

When we replace $b_t(x)$ by $\tilde{b}_t(x) = b_t(x) + \eta_t \nabla r_t(x)$, the analog of equation (15) is:

$$\begin{aligned}\partial_t\hat{\rho}_t + \nabla\cdot(\tilde{b}_t\hat{\rho}_t) &= \partial_t(e^{r_t+\hat{F}_t}\rho_t) + \nabla\cdot(\tilde{b}_t e^{r_t+\hat{F}_t}\rho_t)\\ &= (\partial_t r_t + \partial_t\hat{F}_t)\hat{\rho}_t + e^{r_t+\hat{F}_t}\partial_t\rho_t + \big(b_t+\eta_t\nabla r_t\big)\cdot\nabla r_t\hat{\rho}_t + e^{r_t+\hat{F}_t}\nabla\cdot((b_t+\eta_t\nabla r_t)\rho_t)\\ &= (\partial_t r_t + \partial_t\hat{F}_t)\hat{\rho}_t + b_t\cdot\nabla r_t\hat{\rho}_t + \eta_t\|\nabla r_t\|^2\hat{\rho}_t + \eta_t e^{r_t+\hat{F}_t}\nabla\cdot(\nabla r_t\rho_t)\\ &= \big(b_t\cdot\nabla r_t + \partial_t r_t + \eta_t\big(\|\nabla r_t\|^2 + \Delta r_t + \langle\nabla r_t, s_t\rangle\big) + \partial_t\hat{F}_t\big)\hat{\rho}_t\end{aligned} \tag{37}$$

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 $b_t\cdot\nabla r_t + \partial_t r_t + \eta_t\big(\|\nabla r_t\|^2 + \Delta r_t + \langle\nabla r_t, s_t\rangle\big)$.

And in analogy with Proposition 2.1, if $(\tilde{x}_t, A_t)$ is a solution of

$$d\tilde{x}_t = \big(\tilde{b}_t(\tilde{x}_t) + \epsilon_t s_t(\tilde{x}_t) + \epsilon_t\nabla r_t(\tilde{x}_t)\big)dt + \sqrt{2\epsilon_t}dW_t, \qquad \tilde{x}_0\sim\rho_0, \tag{38}$$

$$dA_t = \big(b_t(\tilde{x}_t)\cdot\nabla r_t(\tilde{x}_t) + \partial_t r_t(\tilde{x}_t) + \eta_t\big(\|\nabla r_t(\tilde{x}_t)\|^2 + \Delta r_t(\tilde{x}_t) + \langle\nabla r_t(\tilde{x}_t), s_t(\tilde{x}_t)\rangle\big)\big)\,dt, \qquad A_0 = 0, \tag{39}$$

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

$$\int_{\mathbb{R}^d} h(x)\hat{\rho}_t(x)dx = \frac{\mathbb{E}[e^{A_t}h(\tilde{x}_t)]}{\mathbb{E}[e^{A_t}]}. \tag{40}$$

We omit a full proof of this statement, as it is analogous to the proof of Proposition 2.1 in Appendix A.1, simply replacing b_t by $\tilde{b}_t$, and $b_t\cdot\nabla r_t + \partial_t r_t$ by $b_t\cdot\nabla r_t + \partial_t r_t + \eta_t\big(\|\nabla r_t\|^2 + \Delta r_t + \langle\nabla r_t, s_t\rangle\big)$.

It only remains to show that the ODE (39) is equal to the ODE (21) in the statement of the result. Since $r_t(x) = tr(X_{t,1}(x))$ as in Proposition 2.2, using the argument in Appendix A.2 yields

$$b_t(x)\cdot\nabla r_t(x) + \partial_t r_t(x) = r(X_{t,1}(x)) = \frac{r_t(x)}{t}. \tag{41}$$

Thus, the solution of equation (39) is

$$A_t = \int_0^t\left(\frac{r_\tau(\tilde{x}_\tau)}{\tau} + \eta_\tau\big(\|\nabla r_\tau(\tilde{x}_\tau)\|^2 + \Delta r_\tau(\tilde{x}_\tau) + \langle\nabla r_\tau(\tilde{x}_\tau), s_\tau(\tilde{x}_\tau)\rangle\big)\right)d\tau. \tag{42}$$

Next, we handle the term ∇r_t. Applying Lemma A.2, we obtain that

$$\begin{aligned}\int_0^t \eta_\tau\Delta r_\tau(\tilde{x}_\tau)\,d\tau &= \int_0^t \tau\eta_\tau\Delta(r\circ X_{\tau,1})(\tilde{x}_\tau)\,\mathrm{d}\tau\\ &= \int_t^{t'}\frac{\tau\eta_\tau}{\sqrt{2\epsilon_\tau}}\nabla(r\circ X_{\tau,1})(\tilde{x}_\tau)\cdot\mathrm{d}^-W_\tau - \int_t^{t'}\frac{\tau\eta_\tau}{\sqrt{2\epsilon_\tau}}\nabla(r\circ X_{\tau,1})(\tilde{x}_\tau)\cdot\mathrm{d}W_\tau,\end{aligned} \tag{43}$$

And plugging this back into (42) concludes the proof:

$$dA_t = \left[\frac{r_t(\tilde{x}_t)}{t} + \eta_t\big(\|\nabla r_t\|^2 + \nabla r_t\cdot s_t\big)(\tilde{x}_t)\right]dt + \frac{\eta_t}{\sqrt{2\epsilon_t}}\nabla r_t(\tilde{x}_t)\cdot\mathrm{d}^-W_t - \frac{\eta_t}{\sqrt{2\epsilon_t}}\nabla r_t(\tilde{x}_t)\cdot\mathrm{d}W_t. \tag{44}$$

Lemma A.2. *Assume that $(\tilde{x}_t)_{t\in[0,1]}$ satisfies the SDE (38). We have that*

$$\int_t^{t'}\tau\eta_\tau\Delta(r\circ X_{\tau,1})(\tilde{x}_\tau)\,\mathrm{d}\tau = \int_t^{t'}\frac{\tau\eta_\tau}{\sqrt{2\epsilon_\tau}}\nabla(r\circ X_{\tau,1})(\tilde{x}_\tau)\cdot\mathrm{d}^-W_\tau - \int_t^{t'}\frac{\tau\eta_\tau}{\sqrt{2\epsilon_\tau}}\nabla(r\circ X_{\tau,1})(\tilde{x}_\tau)\cdot\mathrm{d}W_\tau, \tag{45}$$

where the forward and backward Itô integrals are defined respectively as

$$\int_t^{t'} H_\tau\cdot\mathrm{d}W_\tau \;:=\; \lim_{|\pi|\to0}\sum_{k=0}^{n-1} H_{t_k}\cdot\big(W_{t_{k+1}} - W_{t_k}\big), \tag{46}$$

$$\int_t^{t'} H_\tau\cdot\mathrm{d}^-W_\tau \;:=\; \lim_{|\pi|\to0}\sum_{k=0}^{n-1} H_{t_{k+1}}\cdot\big(W_{t_{k+1}} - W_{t_k}\big). \tag{47}$$

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,

$$\int_0^t H_\tau \, d^- W_\tau \; - \; \int_0^t H_\tau \, dW_\tau = \lim_{|\pi|\to 0} \sum_{k=0}^{n-1} \big(H_{t_{k+1}} - H_{t_k}\big) \cdot \big(W_{t_{k+1}} - W_{t_k}\big) := [H, W]_t, \quad (48)$$

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

$$\begin{aligned} &d(\gamma_\tau \cdot \nabla(r \circ X_{\tau,1}))(\tilde{x}_\tau) \\ &= \gamma_t \Big(\partial_t \nabla(r \circ X_{t,1})(\tilde{x}_\tau) + \nabla^2(r \circ X_{t,1})(\tilde{x}_\tau) \cdot \big(\tilde{b}_t + \epsilon_t s_t - \epsilon_t \nabla r_t\big)(\tilde{x}_\tau) + \epsilon_t \nabla \cdot \nabla^2(r \circ X_{t,1})(\tilde{x}_\tau) \Big)\, \mathrm{d}t \\ &\quad + \dot{\gamma}_t \nabla(r \circ X_{t,1})(\tilde{x}_\tau)\, \mathrm{d}t + \sqrt{2\epsilon_t}\gamma_t \nabla^2(r \circ X_{t,1})(\tilde{x}_\tau) dW_t. \end{aligned} \quad (49)$$

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

$$\begin{aligned} [H, W]_t &= \lim_{|\pi|\to 0} \sum_{k=0}^{n-1} \sqrt{2\epsilon_{t_k}}\gamma_{t_k} \langle \nabla^2(r \circ X_{t_k,1})(\tilde{x}_{t_k})(W_{t_{k+1}} - W_{t_k}), W_{t_{k+1}} - W_{t_k} \rangle \\ &= \lim_{|\pi|\to 0} \sum_{k=0}^{n-1} \sqrt{2\epsilon_{t_k}}\gamma_{t_k} \mathrm{Tr}\big(\nabla^2(r \circ X_{t_k,1})(\tilde{x}_{t_k})\big)(t_{k+1} - t_k) = \int_0^t \sqrt{2\epsilon_\tau}\gamma_\tau \Delta(r \circ X_{\tau,1})(\tilde{x}_\tau)\, d\tau \\ &= \int_0^t \tau\eta_\tau \Delta(r \circ X_{\tau,1})(\tilde{x}_\tau)\, d\tau, \end{aligned} \quad (50)$$

which concludes the proof. □

Remark A.3 (On the factor η_t)**.** *Observe that the proof of Proposition A.1 would also go through if we had defined $\tilde{b}_t$ a different factor multiplying ∇r_t instead of η_t. The rationale for choosing η_t is that the flow matching vector field $b_t(x)$ can be written in terms of the score function $s_t(x)$ as follows:*

$$b_t(x) = \frac{\dot{\beta}_t}{\beta_t} x + \alpha_t\big(\frac{\dot{\beta}_t}{\beta_t}\alpha_t - \dot{\alpha}_t\big) s_t(x) = \frac{\dot{\beta}_t}{\beta_t} x + \eta_t s_t(x). \quad (51)$$

Thus,

$$\tilde{b}_t(x) = \frac{\dot{\alpha}_t}{\alpha_t} x + \eta_t\big(s_t(x) + \nabla r_t(x)\big). \quad (52)$$

Hence, in replacing $b_t(x)$ by $\tilde{b}_t(x)$ we are substituting the score of the base process by the score of the tilted process.

Remark A.4 (Computing Itô integrals vs. the Laplacian Δr_t)**.** *The log-weight A_t could be computed by solving the ODE (39), but that would require approximating the Laplacian ∇r_t using the Hutchinson trace estimator, which would increase variance and add substantial computational cost. By rewriting the integral of ∇r_t in terms of forward and backward Itô integrals, we are able to obtain low error ODE solutions without additional overhead.*

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 2.2 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*. In what follows, we use a notation similar to the one of Syed et al. (2024).

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

$$X_k^n \sim M_{t_{k-1},t_k}(X_{k-1}^n). \tag{53}$$

For the dynamics of Proposition 2.2 and Proposition A.1, the Markov transition kernels are the Euler-Maruyama updates for the SDEs (13) and (20), respectively:

$$\begin{aligned} X_k^n &= X_{k-1}^n + [b_{t_{k-1}}(X_{k-1}^n) + \epsilon_{t_{k-1}}(s_{t_{k-1}}(X_{k-1}^n) + \nabla r_{t_{k-1}}(X_{k-1}^n))](t_k - t_{k-1}) \\ &\quad + \sqrt{2\epsilon_k(t_k - t_{k-1})}\xi_{k-1}^n, \quad \xi_{k-1}^n \sim N(0, \mathrm{I}), \end{aligned} \tag{54}$$

$$\begin{aligned} X_k^n &= X_{k-1}^n + [b_{t_{k-1}}(X_{k-1}^n) + \eta_{t_{k-1}}\nabla r_{t_{k-1}}(X_{k-1}^n) + \epsilon_{t_{k-1}}(s_{t_{k-1}}(X_{k-1}^n) + \nabla r_{t_{k-1}}(X_{k-1}^n))](t_k - t_{k-1}) \\ &\quad + \sqrt{2\epsilon_k(t_k - t_{k-1})}\xi_{k-1}^n, \quad \xi_{k-1}^n \sim N(0, \mathrm{I}), \end{aligned} \tag{55}$$

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

$$\mathbf{w}_k = (w_k^n)_{n\in[N]}, \qquad w_k^n = w_{k-1}^n \, g_{t_{k-1},t_k}(X_{k-1}^n, X_k^n), \tag{56}$$

where for the dynamics of Proposition A.1 and Proposition A.1, the expression of g_{t_{k-1},t_k} is, respectively,

$$g_{t_{k-1},t_k}(X_{k-1}^n, X_k^n) = \exp\left((t_k - t_{k-1})r(X_{t_{k-1},1}(X_{k-1}^n))\right) + o(|t_k - t_{k-1}|), \tag{57}$$

$$\begin{aligned} g_{t_{k-1},t_k}(X_{k-1}^n, X_k^n) = \exp\Bigg(&\left[\frac{r_{t_{k-1}}(X_{k-1}^n)}{t_{k-1}} + \eta_{t_{k-1}}\left(\|\nabla r_{t_{k-1}}\|^2 + \nabla r_{t_{k-1}} \cdot s_{t_{k-1}}\right)(X_{k-1}^n)\right](t_k - t_{k-1}) \\ &+ \eta_{t_k}\sqrt{\frac{t_k - t_{k-1}}{2\epsilon_{t_k}}}\nabla r_{t_k}(X_k^n) \cdot \xi_{k-1}^n - \eta_{t_{k-1}}\sqrt{\frac{t_k - t_{k-1}}{2\epsilon_{t_{k-1}}}}\nabla r_{t_{k-1}}(X_{k-1}^n) \cdot \xi_{k-1}^n\Bigg) \\ &+ o(|t_k - t_{k-1}|), \end{aligned} \tag{58}$$

The terms $o(|t_k - t_{k-1}|)$ account for the higher-order numerical errors that we incur by simulating the coupled SDE-ODE using the Euler-Maruyama and Euler schemes. In practice, we disregard them, but they must be included for a rigorous theoretical treatment.

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 (Chopin et al., 2022) for annealed-SMC-specific resampling schemes, and (Dai et al., 2020) 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

*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 $C = 1$.

Appendix 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

$$\int_{\mathbb{R}^d} h(x)\bar{\rho}_{t_k}(x)dx = \mathbb{E}\left[\left(\prod_{r\in\mathcal{R}\cap[k-1]} \frac{1}{N}\sum_{n=1}^N w_r^n\right)\frac{1}{N}\sum_{n=1}^N w_k^n h(X_k^n)\right], \tag{59}$$

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

$$\int_{\mathbb{R}^d} h(x)\hat{\rho}_{t_k}(x)dx = \frac{\mathbb{E}\left[\hat{Z}_{\mathrm{SMC}}^{(k)}\frac{\frac{1}{N}\sum_{n=1}^N w_{t_k}^n h(X_{t_k}^n)}{\frac{1}{N}\sum_{n=1}^N w_{t_k}^n}\right]}{\mathbb{E}\left[\hat{Z}_{\mathrm{SMC}}^{(k)}\right]}, \quad \hat{Z}_{\mathrm{SMC}}^{(k)} = \left(\prod_{r\in\mathcal{R}\cap[k-1]}\frac{1}{N}\sum_{n=1}^N w_r^n\right)\times\left(\frac{1}{N}\sum_{n=1}^N w_k^n\right). \tag{60}$$

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

$$\int_{\mathbb{R}^d} h(x)\hat{\rho}_1(x)dx = \frac{\mathbb{E}\left[\hat{Z}_{\mathrm{SMC}}\frac{\frac{1}{N}\sum_{n=1}^N w_K^n h(X_K^n)}{\frac{1}{N}\sum_{n=1}^N w_K^n}\right]}{\mathbb{E}\left[\hat{Z}_{\mathrm{SMC}}\right]}, \qquad \text{where } \hat{Z}_{\mathrm{SMC}} = \prod_{r\in\mathcal{R}}\frac{1}{N}\sum_{n=1}^N w_r^n. \tag{61}$$

$\hat{Z}_{\mathrm{SMC}}$ 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 Appendix B.3 we reproduce a result by Syed et al. (2024) 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 Appendix B.2.

B.2 The incremental and total discrepancies

Following Syed et al. (2024), we define the normalized incremental weight function $G_{t,t'}$ as

$$G_{t,t'}(x,x') = \frac{g_{t,t'}(x,x')}{\mathbb{E}_{(X,X')\sim\hat{\rho}_t\otimes M_{t,t'}}[g_{t,t'}(X,X')]} \tag{62}$$

where $g_{t,t'}$ is the unnormalized incremental weight function defined in (56) and $M_{t,t'}$ is the Markov transition kernel defined in (53).

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

$$D(t,t') = \log\left(1 + \mathrm{Var}_{(X,X')\sim\hat{\rho}_t\otimes M_{t,t'}}\left(G_{t,t'}(X,X')\right)\right). \tag{63}$$

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*:

$$D(\mathcal{T}, t_k, t_{k'}) = \sum_{k''=k+1}^{k'} D(t_{k''-1}, t_{k''}), \qquad D(\mathcal{T}) = D(\mathcal{T}, 0, 1). \tag{64}$$

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

$$\begin{aligned} D(t,t') &= \log\left(1 + \mathrm{Var}_{(X,X')\sim\hat{\rho}_t\otimes M_{t,t'}}\left[\frac{g_{t,t'}(X,X')}{\mathbb{E}_{(X'',X''')\sim\hat{\rho}_t\otimes M_{t,t'}}[g_{t,t'}(X'',X''')]}\right]\right) \\ &= \log\left(1 + \frac{\mathbb{E}_{(X,X')\sim\hat{\rho}_t\otimes M_{t,t'}}\left[g_{t,t'}(X,X')^2\right]}{\mathbb{E}_{(X'',X''')\sim\hat{\rho}_t\otimes M_{t,t'}}[g_{t,t'}(X'',X''')]^2} - \frac{\mathbb{E}_{(X,X')\sim\hat{\rho}_t\otimes M_{t,t'}}\left[g_{t,t'}(X,X')\right]^2}{\mathbb{E}_{(X'',X''')\sim\hat{\rho}_t\otimes M_{t,t'}}[g_{t,t'}(X'',X''')]^2}\right) \\ &= \log\mathbb{E}_{(X,X')\sim\hat{\rho}_t\otimes M_{t,t'}}\left[g_{t,t'}(X,X')^2\right] - 2\log\mathbb{E}_{(X,X')\sim\hat{\rho}_t\otimes M_{t,t'}}\left[g_{t,t'}(X,X')\right] \end{aligned} \tag{65}$$

We want to obtain a consistent estimator for $D(t,t')$. Following (Syed et al., 2024, Sec. 5.1), if we are using a single SMC run with N particles, and we let $g_k^n = g_{t_{k-1},t_k}(X_{k-1}^n, X_k^n)$, then the following estimator is consistent:

$$\hat{D}_k = \log\hat{g}_{k,2} - 2\log\hat{g}_{k,1} - \log\hat{g}_{k,0}, \quad \text{where for } i\in\{0,1,2\}, \quad \hat{g}_{k,i} = \sum_{n\in[N]} w_{k-1}^n\left(g_k^n\right)^i. \tag{66}$$

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 (60) for each SMC run $j = 1 : N_R$, and define $\hat{D}_k$ as in (66), but where $\hat{g}_{k,i}$ takes the form

$$\hat{g}_{k,i} = \sum_{j=1}^{N_R} \hat{Z}_{\mathrm{SMC}}^{(k,j)} \frac{\sum_{n\in[N]} w_{k-1}^n \left(g_k^{(n,j)}\right)^i}{\sum_{n\in[N]} w_{k-1}^n}, \qquad \text{where} \quad g_k^{(n,j)} = g_{t_{k-1},t_k}(X_{k-1}^{(n,j)}, X_k^{(n,j)}), \tag{67}$$

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 (64) is related to the variance of the SMC normalization constant Z_{SMC} via the following result, which was proven by Syed et al. (2024) as a generalization of a result of Dai et al. (2020):

Theorem B.1 (Theorem 1, Syed et al. (2024))**.** *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 (62) 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

$$\mathrm{Var}\left(\frac{\hat{Z}_{\mathrm{SMC}}}{Z}\right) = \frac{1}{N}\left(\exp\left(\frac{D(\mathcal{T})}{R_{\mathrm{eff}}}\right) - 1\right) R_{\mathrm{eff}} - 1. \tag{3}$$

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 Syed et al. (2024), Assumptions 1–4 constitute an idealized model similar to the one considered by other works in the area (Grosse et al., 2013; Dai et al., 2020). 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. (2024, Sec. 6.1) show that empirically, the scaling (3) is consistent with empirical observation.*

B.4 Optimizing the annealing schedule to minimize the total discrepancy

As defined in (64), 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 (Syed et al., 2024, Sec. 4.1)), 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

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

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

$$D(t, t+\Delta t) = \delta(t)\Delta t + O(\Delta t^3). \tag{68}$$

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 (68), we obtain

$$D(t_{k-1}, t_k) \approx \frac{\delta(\varphi(u_k))\,\dot{\varphi}(u_k)^2}{K^2}. \tag{69}$$

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, Syed et al. (2024))**.** *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:

$$\min_{\varphi:[0,1]\to[0,1]} \int_0^1 \delta(\varphi(u))\,\dot{\varphi}(u)^2\,du, \quad \text{s.t.} \quad \int_0^1 \dot{\varphi}(u)\,du = 1. \tag{70}$$

Jensen's inequality implies that

$$\int_0^1 \delta(\varphi(u))\,\dot{\varphi}(u)^2\,du \geq \left(\int_0^1 \sqrt{\delta(\varphi(u))}\,\dot{\varphi}(u)\,du \right)^2, \tag{71}$$

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

$$\Lambda(t) = \int_0^t \sqrt{\delta(u)}\,du, \tag{72}$$

by the chain rule, we have equivalently that

$$\Lambda = \Lambda'(\varphi^\star(t))\,\dot{\varphi^\star}(t) = \frac{d}{dt}\Lambda(\varphi^\star(t)) \implies \Lambda(\varphi^\star(t)) = \Lambda t \implies \varphi^\star(t) = \Lambda^{-1}(\Lambda t). \tag{73}$$

Setting $t = 1$ in $\Lambda(\varphi^\star(t)) = \Lambda t$ also implies that $\Lambda = \Lambda(\varphi^\star(1)) = \Lambda(1) = \int_0^1 \sqrt{\delta(u)}\,du$. Syed et al. (2024) refer to $\Lambda(t)$ and Λ as the *local barrier* and the *global barrier* associated to the SMC algorithm. We refer to it as the thermodynamic length.

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

$$\begin{aligned} &\Lambda(\varphi(t)) = \int_0^{\varphi(t)} \sqrt{\delta(u)}\,du = \int_0^t \sqrt{\delta(\varphi(u))}\dot{\varphi}(u)\,du, \\ &\implies \Lambda(t_k) = \Lambda(\varphi(u_k)) = \int_0^{u_k} \sqrt{\delta(\varphi(u))}\dot{\varphi}(u)\,du \approx \sum_{k'=1}^k \frac{\sqrt{\delta(\varphi(u_k))}\,\dot{\varphi}(u_k)}{K} = \sum_{k'=1}^k \sqrt{D(t_{k-1}, t_k)}, \end{aligned} \tag{74}$$

where the last equality holds by (69). This allows us to approximate the local and global barriers using (65) to compute the incremental discrepancy. Once we have an estimate $\hat{\Lambda}$ of the barrier, Syed et al. (2024) 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

$$\sum_{k'=1}^{k} D(t_{k-1}, t_k) = D(\mathcal{T}, 0, t_{k'}) \approx \frac{1}{K} E(\varphi, u_k, u_{k'}) \geq \frac{1}{K} \Lambda(t_k)^2 \approx \frac{1}{K} \left(\sum_{k'=1}^{k} \sqrt{D(t_{k-1}, t_k)} \right)^2, \tag{75}$$

Thus, the quantity

$$\frac{\left(\sum_{k'=1}^{k} \sqrt{D(t_{k-1}, t_k)} \right)^2}{K \sum_{k'=1}^{k} D(t_{k-1}, t_k)} \tag{76}$$

should fall in $[0, 1]$ 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 $(\eta_k)_{k=0:K}$.
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
    $x_{ij}^{k+1} = x_{ij}^k + [v_{t_k,t_k}(x_{ij}^k) + \eta_k \nabla r_{t_k}(x_{ij}^k) + \epsilon_k(s_{t_k}(x_{ij}^k) + \nabla r_{t_k}(x_{ij}^k))]\Delta t + \sqrt{2\epsilon_k \Delta t}\xi_{ij}^k, \quad \xi_{ij}^k \sim N(0, \mathrm{I})$.
    if Sampling **then**
      $A_{ij}^{k+1} = A_{ij}^k + [\frac{r_{t_k}(x_{ij}^k)}{t_k} + \eta_k \big(\|\nabla r_{t_k}\|^2 + \nabla r_{t_k} \cdot s_{t_k}\big)(x_{ij}^k)]\Delta t + \big(\frac{\eta_{k+1} \nabla r_{t_{k+1}}(x_{ij}^{k+1})}{\sqrt{2\epsilon_{k+1}}} - \frac{\eta_k \nabla r_{t_k}(x_{ij}^k)}{\sqrt{2\epsilon_k}}\big) \cdot \sqrt{\Delta t}\xi_{ij}^k$
    end if
  end for
  if $k = 0 \pmod{K/R}$ and $k > 0$ **then** ▷ Resample / select particles
    if Sampling **then**
      Define the probabilities $p^k = \mathrm{softmax}(A^k) = \big(\exp(A_{ij}^k)/\sum_{i'j'} \exp(A_{i'j'}^k)\big)_{i'=1:N, j'=1:C}$
      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 VLM-AS-JUDGE EXAMPLES

As described in the paper, we 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, character consistency, and multi-subject generation.

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. Qualitative results are shown in Figure 7.

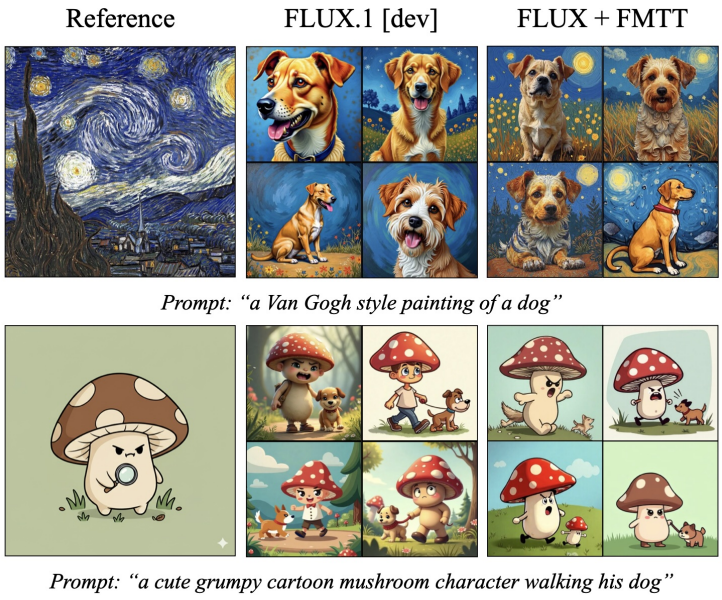

Figure 7: Style consistency via VLM-based rewards. Given a reference image, FMTT produces images that better match its art style than the base model.

E 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 8 shows such a case.

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

Figure 8: 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?

LLM USAGE

In preparing this paper, we used large language models (LLMs) as assistive tools. Specifically, LLMs were used for (i) editing and polishing the text for clarity and readability, and (ii) generating some reference images that appear in some figures. All research ideas, experiments, and analysis were conducted by the authors. The authors take full responsibility for the content of this paper.