

IMPROVED TECHNIQUES FOR FINE-TUNING FLOW MODELS VIA ADJOINT MATCHING: A DETERMINISTIC CONTROL PIPELINE

ZHENGYI GUO, JIAYUAN SHENG, DAVID D. YAO, AND WENPIN TANG

ABSTRACT. We propose a deterministic adjoint matching framework that formulates human preference alignment for flow-based generative models as an optimal control problem over velocity fields. One can directly regress the control toward a value-gradient-induced target under the current policy, leading to a simple and stable training objective. Building on this perspective, we introduce a truncated adjoint scheme that focuses computation on the terminal portion of the trajectory, where reward-relevant signals concentrate, which yields substantial computational savings while preserving alignment quality. We further generalize the framework beyond standard KL-based regularization, allowing more flexible trade-offs between alignment strength and distributional preservation. Experiments on SiT-XL/2 and FLUX.2-Klein-4B demonstrate consistent gains across multiple alignment metrics, along with substantially improved diversity and mode preservation.



FIGURE 1. Prompts are “a sad muppet funeral in a rainy graveyard”; “a futuristic house on a floating island with waterfalls and moons”; “a monkey in a blue top hat painted in oil by Vincent van Gogh”.

1. INTRODUCTION

Flow matching models [1, 21, 24] are a class of generative models that train neural networks to predict a velocity field. This velocity field, described by an ordinary differential equation

Date: May 7, 2026.

(ODE), enables the transformation of simple probability distributions into complex data distributions along straighter, continuous-time trajectories. Due to their training and sampling efficiency, flow matching models have seen immense success in large-scale, high-fidelity image/video generation, powering state-of-the-art models such as Stable Diffusion v3 [9], Flux.2 [5] and WAN [38].

It is known that pretrained base models often fail to produce samples that align with human preferences for quality and prompt adherence [45]. This is observed even for state-of-the-art flow-based models such as Flux.2; see Figure 2. To align a base model with desired outputs without compromising its foundational strengths, reward-based post-training is necessary.



FIGURE 2. Prompts are “Two planes are placed next to each other.” and “Two planes sit together on the grass.” From left to right: (1) Base, (2) 4th ODE-AM-1, (3) 2nd ODE-AM-3, (4) DRaFT-1, and (5) ReFL-5 on **FLUX.2-Klein-4B**.

Reinforcement learning from human feedback (RLHF), a reward-based post-training technique, was first applied to fine-tune LLMs [27] and has since been extended to diffusion model post-training [4, 10]. It encompasses RL with zeroth-order reward such as DDPO [4], DPOK [11] and MixGRPO [20]; direct reward-feedback methods such as ReFL [47] and DRaFT [6]; and diffusion-DPO [42]. In another direction, stochastic control approaches were developed to fine-tune diffusion models [7, 37, 40]. Typically, the objective is to maximize

$$\text{reward}(\text{post-output}) - \beta D_{KL}(\text{post-output}, \text{pre-output}), \quad (1.1)$$

where post/pre-output is the post/pre-trained generation. One advantage of (1.1) is that it has a closed-form solution, which is an exponential tilting of the pre-output. The goal is to sample this exponential-tilted distribution by solving a stochastic control problem on the pretrained base model; see [39] for a review.

Recently, a series of works [7, 14, 22] proposed to sample a target distribution via *adjoint matching* in the context of stochastic control. The idea is to align the control field with a target reward by solving an adjoint ODE backward in time, which has its root in Pontryagin’s maximum principle [29, 49]. However, there are several drawbacks: (1) the objective of sampling the exponential-tilted distribution seems to be restrictive for diffusion post-training; (2) solving the adjoint ODE can be computationally expensive, especially for large foundation models where accurate adjoint integration requires finer time discretization.

The purpose of this paper is to improve upon adjoint matching techniques for flow models, enhancing both training flexibility and efficiency. The contributions of this paper are as follows:

- **Deterministic control pipeline:** As mentioned, most leading-edge base models have shifted to flow-based architectures. It is more natural to work within a deterministic control framework that aligns with the base model’s inherently efficient sampling scheme, though flow matching models do support stochastic samplers [12, 34, 51]. Here, we formulate post-training as a deterministic control problem that learns a velocity perturbation on top of a pretrained base model. We point out that adjoint matching under this deterministic control setting does not generate exactly the exponential-tilted distribution (i.e., the solution to (1.1)), which requires a stochastic sampler with memoryless noise schedule [7]. Nevertheless, it bypasses extremely large noise in the initial sampling phase with memoryless noise schedule, leading to much faster convergence.

- **Beyond KL regularization:** KL regularization constrains the fine-tuned model to remain close to the base model, mitigating reward hacking. In the stochastic control setting, it translates into a running cost equal to the squared norm of the control field. While most prior works [7, 11, 25, 54] adopted KL regularization in diffusion model alignment, recent studies [18, 37, 43] suggest that alternative regularizers may offer superior performance. Here, we propose a regularization term defined as an increasing function of the control norm. Particular examples include polynomial norms of the control field, which offer greater flexibility in fine-tuning flow-based models.

- **Adjoint matching with practical acceleration:** We derive simple adjoint matching objectives under the deterministic control setting. However, solving the adjoint ODE is computationally expensive. Here, we introduce a truncated adjoint scheme that allocates computation to the most influential terminal steps, following the observations in [7, 19, 47]. This truncated variant enhances the scalability of adjoint matching for large models: it yields a substantial speedup from 345s to 32s per update on FLUX.2-Klein, and meanwhile, the image quality is significantly improved with respect to the base model generation. Theoretically, we illustrate with a few examples where the control norm becomes large toward the terminal steps, thus justifying the truncated scheme.

- **Empirical performance:** We show that our approach achieves a stronger balance between various reward improvements and distributional preservation on two pretrained backbones SiT-XL/2 and FLUX.2-Klein. On top of improving reward-related metrics such as Aesthetic Score, HPSv2, ImageReward, and PickScore, it also maintains diversity and reduces mode collapse as reflected by LPIPS, MS-SSIM, Coverage, and Recall.

Closely related to our paper is the work [25], where deterministic control was applied to fine-tune flow matching models. Different from our approach, they learn the value gradient by directly solving the corresponding partial differential equation.

Related Works: Flow Matching [21] trains a neural network to match a continuous-time velocity field, and Rectified Flow [24] learns straight transport paths between noise and data, enabling efficient ODE-based sampling. Early large-scale image generation systems mostly relied on U-Net [32] backbones, particularly in latent diffusion models (LDMs) and the Stable Diffusion [31]. More recently, transformer-based architectures have replaced U-Nets in high-capacity generative models. Diffusion Transformers [28] introduced ViT [8]-style

backbones for diffusion models, while SiT [26] studied scalable interpolant transformers under both diffusion and flow-based formulations. Modern text-to-image systems such as Stable Diffusion 3 [9] and FLUX [5] further combine rectified-flow-style objectives with transformer architectures, showing the scalability of flow-based modeling.

RLHF has recently been extended from LLMs to diffusion-based generative models. DDPO [4] and DPOK [11] cast the generation process as a stochastic policy and optimizes it via policy gradient methods. Continuous RL approach was developed in [53, 54] to fine-tune diffusion models by treating score matching as action. More recently, GRPO [20, 23, 48]-style methods extend group-relative preference optimization to diffusion models. Despite their flexibility, RL-based approaches often suffer from high variance in gradient estimation and require careful tuning of reward scaling and KL constraints. These challenges become more pronounced in large-scale text-to-image models, where long sampling trajectories amplify optimization instability.

Optimal control provides a principled framework for steering dynamical systems toward desired objectives under trajectory-level costs. More recently, it has been used in generative modeling and sampling [3, 7, 13, 25, 37, 41, 50]. These approaches provide a more principled alternative to reinforcement learning and direct reward optimization, but often suffer from high computational cost due to the need for solving auxiliary equations.

Organization of the paper: The rest of the paper is organized as follows. Section 2 reviews flow models and formulates our fine-tuning problem. Section 3 derives our adaptive deterministic-control pipeline. Section 4 provides theoretical examples that justify its efficiency. Section 5 demonstrates the robust empirical performance of our method. Section 6 concludes.

2. PRELIMINARIES OF FLOW MATCHING MODELS

Flow-based generative models. We present flow matching models, following [21, 24]. Let $X_0 \sim p_0 = \mathcal{N}(0, I)$ and $X_1 \sim p_{\text{data}}$. Define the reference flow

$$\bar{X}_t = \beta_t X_0 + \alpha_t X_1, \quad t \in [0, 1], \quad (2.1)$$

where $\alpha_0 = \beta_1 = 0$, $\alpha_1 = \beta_0 = 1$, and α'_t, β'_t denote the time derivatives of α_t, β_t . Let p_t be the marginal probability distribution of \bar{X}_t . The velocity field is $v_t^*(x) := \mathbb{E}[\beta'_t X_0 + \alpha'_t X_1 \mid \bar{X}_t = x]$, which is learned by solving $v^{\text{base}} := \arg \min_v \mathbb{E}_{t, X_0, X_1} [\|v(\bar{X}_t, t) - (\beta'_t X_0 + \alpha'_t X_1)\|^2]$. Then the flow-based sampler is the probability-flow ODE:

$$dX_t = v^{\text{base}}(X_t, t) dt. \quad (2.2)$$

When $v^{\text{base}} = v^*$, the solution to (2.2) has marginals $X_t \sim p_t$ for all $t \in [0, 1]$, and hence $X_1 \sim p_{\text{data}}$. The same marginal curve can also be generated by a family of stochastic dynamics:

$$dX_t = \left(v_t^*(X_t) + \frac{\sigma^2(t)}{2} \nabla \log p_t(X_t) \right) dt + \sigma(t) dB_t, \quad (2.3)$$

for any state-independent diffusion schedule $\sigma(t)$ (see e.g., [36, Section 5]).

Unified view of different parameterizations. Flow matching and diffusion models can be understood from a common marginal-preserving perspective. Denote

$$s_t(x) := \nabla \log p_t(x), \quad \kappa_t := \frac{\alpha'_t}{\alpha_t}, \quad \eta_t := \beta_t \left(\frac{\alpha'_t}{\alpha_t} \beta_t - \beta'_t \right).$$

By [17, Proposition 6.3.1], for the affine flow (2.1) with Gaussian initial noise, the velocity and score are related by $v_t^*(x) = \kappa_t x + \eta_t s_t(x)$. Equivalently, when $\eta_t \neq 0$, $s_t(x) = \frac{v_t^*(x) - \kappa_t x}{\eta_t}$. So different model parameterizations can be converted into the same velocity representation, see Table 1 for details. As a result, the stochastic and deterministic flow fine-tuning can be unified by holding the same learned velocity v_θ :

$$\begin{aligned} dX_t^{\text{ODE}} &= v_\theta(X_t^{\text{ODE}}, t) dt, \\ dX_t^{\text{SDE}} &= \left[v_\theta(X_t^{\text{SDE}}, t) + \frac{\sigma^2(t)}{2\eta_t} (v_\theta(X_t^{\text{SDE}}, t) - \kappa_t X_t^{\text{SDE}}) \right] dt + \sigma(t) dB_t. \end{aligned}$$

Parameterization	Predicted quantity	Conversion to velocity
Velocity field	$v_\theta(x, t)$	$v_\theta(x, t)$
Score	$\mathbf{s}_\theta(x, t) \approx \nabla \log p_t(x)$	$v_\theta(x, t) = \kappa_t x + \eta_t \mathbf{s}_\theta(x, t)$
Noise	$\boldsymbol{\epsilon}_\theta(x, t) \approx \mathbb{E}[X_0 \mid X_t = x]$	$v_\theta(x, t) = \kappa_t x - (\kappa_t \beta_t - \beta'_t) \boldsymbol{\epsilon}_\theta(x, t)$
Clean data	$\mathbf{x}_\theta(x, t) \approx \mathbb{E}[X_1 \mid X_t = x]$	$v_\theta(x, t) = \frac{\beta'_t}{\beta_t} x - \left(\frac{\beta'_t}{\beta_t} \alpha_t - \alpha'_t \right) \mathbf{x}_\theta(x, t)$

TABLE 1. Equivalent parameterizations under (2.1).

3. FINE-TUNE FLOW MODELS WITH ADJOINT MATCHING

3.1. Deterministic Optimal Control. Consider the control problem [2, 29, 35]:

$$\min_u \mathcal{L}(u; \mathbf{X}^u) = \int_0^1 f(\|u(X_t, t)\|) dt + g(X_1), \quad X_0 = x, \quad (3.1)$$

subject to the deterministic control-affine dynamics

$$\dot{X}_t = v^{base}(X_t, t) + u(X_t, t), \quad t \in [0, 1]. \quad (3.2)$$

Here $X_t \in \mathbb{R}^d$ denotes the state of the system, $u : \mathbb{R}^d \times [0, 1] \rightarrow \mathbb{R}^d$ is the control, v^{base} is the pre-trained velocity field, $f : \mathbb{R}_+ \rightarrow \mathbb{R}$ is a regularization on the control magnitude, and $g : \mathbb{R}^d \rightarrow \mathbb{R}$ is the terminal cost.

Define the cost-to-go functional starting from state x at time t under control u :

$$J(u; x, t) := \int_t^1 f(\|u(X_s, s)\|) ds + g(X_1), \quad \text{where } X_t = x.$$

The corresponding value function is:

$$V(x, t) := \min_u J(u; x, t) = J(u^*; x, t), \quad (3.3)$$

where $u^*(x, t)$ denotes the pointwise optimal control. Under regularity assumptions, the equality of $V(x, t)$ and the optimal control at $u = u^*$ is given by (See Appendix A.1&A.2):

$$\nabla_u f(\|u\|) \Big|_{u=u^*} + \nabla_x V(x, t) = 0. \quad (3.4)$$

Motivation for general regularization: As mentioned in the introduction, the common choice is $f(r) = \frac{1}{2}r^2$, which corresponds to the KL-regularized control cost. However, this quadratic choice is not indispensable: *more general* control penalties f can be used in the the adjoint-matching target.

3.2. Adjoint Matching(AM). To compute the gradient of $\mathcal{L}(u; \mathbf{X}^u)$ in (3.1) with respect to control parameters, we rely on the adjoint method. As shown in [7, Appendix E.2], this continuous-adjoint gradient can be learned by a *matching* objective. Define

$$a(t; \mathbf{X}^u, u) := \nabla_{X_t} \left(\int_t^1 f(\|u(X_s, s)\|) ds + g(X_1) \right),$$

representing the sensitivity of the cost-to-go under current control u . We have $a(t; \mathbf{X}^u, u)|_{X_t=x} = \nabla_x J(u; x, t)$. It can be shown that the adjoint state satisfies the backward ODE:

$$\frac{d}{dt} a(t; \mathbf{X}^u, u) = - \left[a^\top \nabla_{X_t} \left(v^{base}(X_t, t) + u(X_t, t) \right) + \nabla_{X_t} f(\|u(X_t, t)\|) \right], \quad (3.5)$$

with the terminal condition $a(1; \mathbf{X}^u, u) = \nabla_{X_1} g(X_1)$. Note that at the optimum $u(x, t) = u^*(x, t)$, $a = \nabla_x V(x, t)$. Multiplying (3.4) by $\nabla_{X_t} u(X_t, t)$, we have:

$$a^\top \nabla_{X_t} u(X_t, t) + \nabla_{X_t} f(\|u(X_t, t)\|) = 0.$$

Substituting it into (3.5) yields a simplified “lean” adjoint:

$$\frac{d}{dt} \tilde{a}(t; \mathbf{X}^u, u) = - \left[\tilde{a}(t; \mathbf{X}^u, u)^\top \nabla_{X_t} v^{base}(X_t, t) \right], \quad \tilde{a}(1; \mathbf{X}^u, u) = \nabla_{X_1} g(X_1). \quad (3.6)$$

Motivation for truncated time-steps: Theoretical and empirical analysis (Section 4 and Appendix C) on the backward ODE (3.6) shows that the last few denoising steps are the most important. The parameter $\tau \in [0, 1)$ allows us to start control from an intermediate denoising time. Since the dynamics are deterministic, the distribution of X_1 is induced solely by the distribution X_τ and the chosen control u .

Observe that Equation (3.4) is a fixed-point problem. As long as f is differentiable and invertible, we can solve for $u^*(x, t)$ as a function of the (optimal) lean adjoint $\tilde{a}(x, t)$ from this equation. This provides a surrogate target to train our control iteratively.

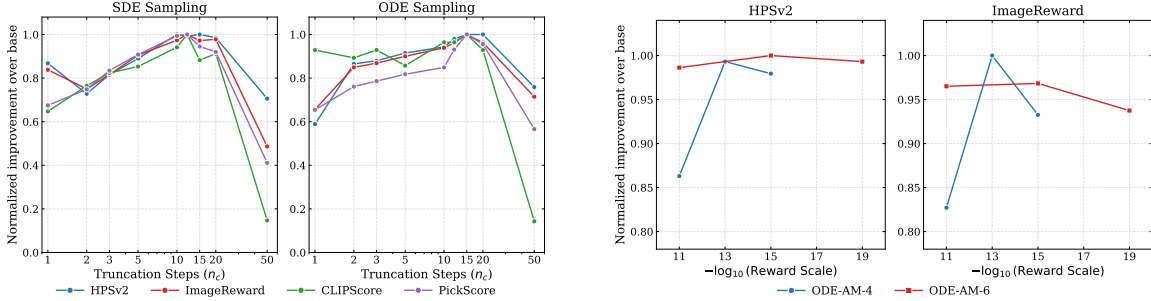
Theorem 3.1. *For the deterministic controlled problem (3.1) on the time horizon $[\tau, 1]$ subject to (3.2), if $f : \mathbb{R}_+ \rightarrow \mathbb{R}$ is continuously differentiable, strictly convex and increasing, then there is an optimal control u^* given by*

$$u_t^* = - \frac{(f')^{-1}(\|a_t\|)}{\|a_t\|} a_t, \quad t \in [\tau, 1]. \quad (3.7)$$

Example: Typically, we can choose **polynomial** regularization $f(x) = \frac{1}{p\lambda} x^p$. Then the optimal control is $u_t^* = -\lambda^{\frac{1}{p-1}} \|a_t\|^{\frac{2-p}{p-1}} a_t$.

Now we propose our Deterministic Adjoint Matching (AM) objective:

$$\mathcal{L}_{\text{Adj-Match}}(u; X) := \frac{1}{2} \int_\tau^1 \|u(X_t, t) - u^*(\tilde{a}(t, X_t))\|^2 dt,$$



(A) Normalized reward improvement for **Truncated AM**. $n_{\text{truncate}} = 10, 12, 15$ achieves best performances.

(B) Normalized reward improvement for **HPSv2** (left) and **ImageReward** (right) under **ODE-AM-4** and **-6** at $n_{\text{truncate}} = 10$.

FIGURE 3. Ablations on **SiT-XL/2**: (a) truncation horizon and (b) reward scale across higher-order ODE adjoint matching variants. See Table 6 and Table 5 in Appendix D for full results

where \tilde{a}_t satisfies ODE (3.6) and the stochastic process $X \sim p^{\tilde{u}}$ with $\tilde{u} = \text{stopgrad}(u)$. See Algorithm 1 for the empirical implementation pipelines of our adaptive methods, where our adaptations are marked in **red** as well.

Stochastic adjoint matching algorithm can also be leveraged in our finetuning tasks, see Appendix B for detailed explanations on stochastic AM and corresponding algorithm.

4. THEORY AND ALGORITHMS OF DETERMINISTIC ADJOINT MATCHING

Section 3 introduced two additional components of deterministic adjoint matching: a truncated time horizon τ and a general control regularizer $f(\cdot)$. Empirically, control intensity is highly non-uniform: [19] showed that successful first-order methods assign more weight to high-SNR steps. Two toy models with *Variance Exploding* and *Variance Preserving* dynamics also show the same trend (Figure 6 in Appendix C). Moreover, integrating the full-trajectory adjoint ODE is expensive for large models and prone to error accumulation.

Quadratic regularization overemphasizes steps with maximal adjoints. Higher-order regularization mitigates this issue (see Appendix D.1). Here we give some simple calculations motivating why combining time truncation with higher-order regularization can significantly reduce computational cost while improving performance. Proofs are deferred to Appendix C.

4.1. Time Horizon. Consider the following elementary setting: $X_0 \perp X_1$ and the rectified flow $X_t = (1-t)X_0 + tX_1$, $t \in [0, 1]$. Let $v_t(x) := \mathbb{E}[X_1 - X_0 \mid X_t = x]$ be the corresponding velocity. We analyze the controlled dynamics $\dot{X}_t^u = v_t(X_t^u) + u_t$ with terminal adjoint $a(1) \neq 0$ and optimal control u_t^* given by Theorem 3.1. Proposition 4.1 shows that, when $\sigma > 1$, the control maximizer lies in the later stage of the denoising trajectory.

Proposition 4.1 (Unique later-stage control maximum). *Suppose $X_0 \sim \mathcal{N}(\mu, \sigma^2)$ and $X_1 \sim \mathcal{N}(0, 1)$ are independent, the control intensity takes a unique maximum at $t^* = \frac{\sigma^2}{1+\sigma^2}$.*

Relative intensity is independent of the specific fine-tuned target. Also, the reward scale λ changes only the absolute, rather than relative, control intensity.

4.2. Regularization. Proposition 4.2 below shows that higher-order polynomial regularization preserves the same peak location but smooths the normalized control curve (see Figure 7 in Appendix C). Therefore, a broader later-stage region remains close to the maximal control strength, which helps truncated adjoint matching retain neighboring reward-relevant signals. This toy example aligns with our empirical observations in Appendix D.1.

Proposition 4.2 (Relative Control strength). *Under the same setup as Proposition 4.1, for p -th order polynomial regularization, the normalized relative control strength is given by*

$$R_p(t) := \frac{\|u_t^*\|}{\max_{s \in [0,1]} \|u_s^*\|} = \left(\frac{\sigma^2}{((1-t)^2\sigma^2 + t^2)(1 + \sigma^2)} \right)^{1/(2p-2)}.$$

Algorithm 1 Deterministic Adjoint Matching with polynomial regularization

Require: Pretrained velocity field v^{base} , parametrized velocity field v_θ , reward model r , exponent $p > 1$, cost weight $\lambda > 0$, step size h , number of control steps T , number of finetune iterations N , batch size m .

- 1: Initialize $v_\theta \leftarrow v^{base}$.
- 2: **for** $n = 0, \dots, N - 1$ **do**
- 3: Generate m trajectories using current model at $t = 0, \dots, 1 - h$:

$$X_{t+h} = X_t + hv_\theta(X_t, t), \quad X_0 \sim \mathcal{N}(0, I).$$

- 4: Integrate the lean adjoint backwards in $t = 1, 1 - h, \dots, 1 - (T - 1)h$:

$$\tilde{a}_{t-h} = \tilde{a}_t + h\tilde{a}_t^\top \nabla_{X_t} v^{base}(X_t, t) \quad \tilde{a}_1 = -\nabla_{X_1} r(X_1).$$

Note that both X_t and \tilde{a}_t should be computed without gradients, *i.e.* Stop gradients: $X_t = \text{stopgrad}(X_t)$, $\tilde{a}_t = \text{stopgrad}(\tilde{a}_t)$.

- 5: Compute target-matching loss for each trajectory:

$$L(\theta) = \frac{1}{mT} \sum_{t \in \{1, \dots, 1-Th\}} \left\| \left(v_\theta(X_t, t) - v^{base}(x_t, t) \right) - \left(-\lambda^{\frac{1}{p-1}} \|\tilde{a}_t\|^{(2-p)/(p-1)} \tilde{a}_t \right) \right\|^2.$$

- 6: Compute the gradient $\nabla_\theta L$ and update θ .
 - 7: **end for**
-

5. NUMERICAL RESULTS

In this section, we test our algorithms against two flow matching baselines: **SiT-XL/2** (675M Params., [26]) and **FLUX.2-Klein** (4B Params., [5]). For both models, we finetune with HPSv2 reward [46]. For evaluation, we also use other preference and alignment metrics: ImageReward [47], Aesthetic Score [33], CLIPScore [30], and PickScore [16].

We present our method by “Regularization order” + “ODE/SDE” + “AM” + “Truncate steps”. For example, **4th ODE-AM-3** means deterministic (ODE) adjoint matching with 4th order polynomial regularization and truncate adjoint ODE to the last 3 steps.

5.1. SiT-XL/2 on ImageNet. SiT achieves strong ImageNet generation quality with small parameter sizes, ideal for detailed ablation studies. To evaluate this class-conditional model, we convert each of the 1000 ImageNet classes into a descriptive class prompt and generate

Method	n_{truncate}	Regularization	HPSv2 \uparrow	ImageReward \uparrow	CLIPScore \uparrow	PickScore \uparrow	Iter. Time (s) \downarrow
Base Model	–	–	0.183 \pm 0.038	-0.658 \pm 0.915	0.239 \pm 0.049	0.190 \pm 0.011	–
DRaFT-1	–	–	0.296 \pm 0.041	0.266 \pm 0.909	0.260 \pm 0.041	0.200 \pm 0.011	<u>26.7</u>
ReFL-10	–	–	0.297 \pm 0.034	0.282 \pm 0.885	0.265 \pm 0.038	0.200 \pm 0.011	26.4
SDE-AM	10	2nd	0.319 \pm 0.038	0.491 \pm 0.876	<u>0.271</u> \pm 0.038	0.206 \pm 0.011	47.6
SDE-AM	12	2nd	0.318 \pm 0.038	0.522 \pm 0.867	0.273 \pm 0.038	<u>0.206</u> \pm 0.011	49.6
SDE-AM	50(Full)	2nd	0.279 \pm 0.038	-0.084 \pm 0.989	0.244 \pm 0.052	<u>0.197</u> \pm 0.012	83.8
ODE-AM	10	2nd	0.316 \pm 0.040	0.450 \pm 0.903	0.266 \pm 0.040	0.204 \pm 0.011	47.3
ODE-AM	10	4th	<u>0.328</u> \pm 0.038	0.574 \pm 0.834	0.269 \pm 0.040	0.206 \pm 0.011	47.3
ODE-AM	10	6th	0.329 \pm 0.039	<u>0.535</u> \pm 0.862	0.269 \pm 0.039	0.207 \pm 0.011	47.7
ODE-AM	12	2nd	0.321 \pm 0.039	0.477 \pm 0.860	0.266 \pm 0.039	0.205 \pm 0.011	49.3
ODE-AM	15	2nd	0.324 \pm 0.039	0.522 \pm 0.877	0.267 \pm 0.040	0.206 \pm 0.011	52.0
ODE-AM	50(Full)	2nd	0.290 \pm 0.037	0.185 \pm 0.967	0.243 \pm 0.049	0.199 \pm 0.012	83.8

TABLE 2. Comparison on reward metrics with **SiT-XL/2**. Best values are shown in **bold**, second-best are underlined.



FIGURE 4. Above: **SiT-XL/2** base model. Below: **SiT-XL/2** with **6th ODE-AM-10** finetune for 150 steps. ImageNet classes, from left to right *Shetland sheepdog*, *tiger cat*, *esser panda (red panda)*, *castle*, *forklift* and *computer keyboard*.

images conditioned on the corresponding class labels. We investigate the effects of trajectory stochasticity, truncate steps and regularization mode with four reward/alignment metrics, see Table 2. The results show our pipelines improve over the base SiT model across all four fidelity metrics, while 10-step truncation with 6th order regularization achieves the overall best performance. See Figure 3 for ablations and Figure 4 for generated images.

SDE-AM-Full and **ODE-AM-Full** already improve over the base SiT model, but they are less efficient, as discussed in Section 4.1. In particular, truncated AM consistently improves the four reward metrics while reducing the per-iteration wall-clock time from 83.8s for **ODE-AM-Full** to 39 ~ 57s. Detailed ablation studies are provided in Appendix D.3.

5.2. FLUX.2-Klein-4B Text to Image. We fine-tune FLUX.2-Klein-4B with 20 sampling steps and 500 iterations, using prompt set from the dataset HPDv2 in HPSv2. For evaluation, other than the fidelity scores, we also report within-prompt diversity metrics LPIPS [52], 1-MS-SSIM [44], and distribution-preservation metrics Recall and Recovery. Detailed evaluation protocols are deferred to Appendix D.4.

From Table 3, our truncated AM variants (even with 1 step) improve the HPSv2 reward dramatically compared to the base model. While the baselines DRaFT and ReFL achieve slightly stronger gains on some fidelity metrics, our methods preserve much better within-prompt diversity and the base-model distribution, as shown in Table 4. In addition, with qualitative examples provided in Figure 5, we see clearly that truncated AM is less prone to reward hacking and mode collapse.



FIGURE 5. Prompts are “A portrait of a cat wearing a samurai helmet.” and “A snowy lake in Sweden captured in a vibrant, cinematic style with intense detail and raytracing technology showcased on Artstation.” Left to right: (1) Base, (2) 2nd ODE-AM-1, (3) 2nd ODE-AM-3, (4) DRaFT-1, and (5) ReFL-5.

To put it more clearly, some methods such as **6th ODE-AM-1** and **ReFL-5**, appear to score higher on training reward HPSv2. However, when it comes to image diversity, methods such as **2nd ODE-AM-3**, **2nd ODE-AM-1** and **6th ODE-AM-3** outperforms, as they have high LPIPS and 1-MS-SSIM. Overall, our approach is better at preserving diversity, which is consistent with the observation in [25] that methods based on direct differentiation tend not to align with the target distribution, but only favor some modes of it. To examine whether fine-tuning leads to mode collapse, we also compute Recall and Coverage to measure how much the fine-tuned distribution deviates from the original base distribution. We observe that **2nd ODE-AM-1** and **6th ODE-AM-3** preserve the distribution significantly better than others. One reason is that high-reward training methods such as **DRaFT** and **ReFL** tend to produce overly bright or visually flashy images, often at the cost of semantic alignment. Through various level of truncated steps and order of regularization, our methods managed to balance the tradeoff between achieving high image fidelity and preventing reward hacking.

Finally, we observe significant acceleration with truncated Adjoint Matching compared to full adjoint matching during training: wall-clock time decreased from 345s/iter. in ODE-AM-Full to 32s/iter. in ODE-AM-1. For detailed analysis, see Appendix D.3.

6. CONCLUSION AND FUTURE DIRECTIONS

We introduced a deterministic AM pipeline for fine-tuning flow models, combining general control regularization with truncated adjoint computation. The method improves reward alignment while preserving diversity and substantially reducing training cost.

In the future, we will extend deterministic adjoint matching to video generation or discrete diffusion models. In addition, our adaptive regularization is currently developed only for the

Method	n_{truncate}	Regularization	Aesthetic \uparrow	HPSv2 \uparrow	ImageReward \uparrow	PickScore \uparrow	Iter. Time (s) \downarrow
Base Model	–	–	5.549 \pm 0.665	0.290 \pm 0.039	1.074 \pm 0.645	0.221 \pm 0.014	–
DRaFT-1	–	–	6.440 \pm 0.385	0.458 \pm 0.045	<u>1.556</u> \pm 0.455	<u>0.235</u> \pm 0.017	24.7
ReFL-5	–	–	6.336 \pm 0.356	<u>0.466</u> \pm 0.046	1.473 \pm 0.521	0.234 \pm 0.018	24.1
ODE-AM	1	2nd	6.297 \pm 0.544	0.413 \pm 0.047	1.470 \pm 0.503	<u>0.235</u> \pm 0.017	32.2
ODE-AM	1	4th	6.307 \pm 0.427	0.449 \pm 0.043	1.522 \pm 0.503	<u>0.235</u> \pm 0.017	32.2
ODE-AM	1	6th	6.204 \pm 0.400	0.470 \pm 0.046	1.531 \pm 0.514	<u>0.235</u> \pm 0.018	32.2
ODE-AM	3	2nd	6.297 \pm 0.550	0.422 \pm 0.047	1.505 \pm 0.490	0.236 \pm 0.017	65.3
ODE-AM	3	4th	6.313 \pm 0.440	0.449 \pm 0.043	1.557 \pm 0.475	<u>0.235</u> \pm 0.017	65.3
ODE-AM	3	6th	6.309 \pm 0.530	0.422 \pm 0.043	1.502 \pm 0.524	0.234 \pm 0.017	65.3
ODE-AM	20(Full)	2nd	<u>6.437</u> \pm 0.520	0.382 \pm 0.044	1.160 \pm 0.679	0.227 \pm 0.017	345.5

TABLE 3. Comparison on fidelity/reward/time metrics with **FLUX.2-Klein-4B**. Best values are in **bold**, second-best are underlined.

Method	n_{truncate}	Regularization	LPIPS \uparrow	1-MS-SSIM \uparrow	Coverage \uparrow	Recall \uparrow
Base Model	–	–	0.505 \pm 0.073	0.857 \pm 0.113	–	–
DRaFT-1	–	–	0.401 \pm 0.069	0.757 \pm 0.102	0.080	0.010
ReFL-5	–	–	0.431 \pm 0.070	0.779 \pm 0.103	0.047	0.002
ODE-AM	1	2nd	0.448 \pm 0.078	<u>0.832</u> \pm 0.111	<u>0.376</u>	0.129
ODE-AM	1	4th	0.420 \pm 0.074	0.801 \pm 0.109	0.204	0.037
ODE-AM	1	6th	0.424 \pm 0.073	0.804 \pm 0.102	0.139	0.012
ODE-AM	3	2nd	0.454 \pm 0.074	0.843 \pm 0.104	0.306	0.057
ODE-AM	3	4th	0.432 \pm 0.075	0.810 \pm 0.111	0.224	0.042
ODE-AM	3	6th	<u>0.450</u> \pm 0.073	0.813 \pm 0.115	0.396	0.083
ODE-AM	20(Full)	2nd	0.449 \pm 0.065	0.817 \pm 0.098	0.262	<u>0.106</u>

TABLE 4. Comparison on diversity and distribution metrics with **FLUX.2-Klein-4B**. Best values are in **bold**, second-best are underlined (excluding the base model).

deterministic dynamics; extending it under the stochastic maximum principle would provide a more complete theory. Finally, a principled recipe for optimal truncation, potentially through online estimation of control intensity, is left for future work.

Acknowledgment: Wenpin Tang is supported by NSF CAREER Award DMS-2538791 and the Tang Family Assistant Professorship. Jiayuan Sheng and David D. Yao are part of a Columbia-CityU/HK collaborative project that is supported by InnoHK Initiative, The Government of the HKSAR and the AIFT Lab.

REFERENCES

- [1] M. S. Albergo, N. M. Boffi, and E. Vanden-Eijnden. Stochastic interpolants: a unifying framework for flows and diffusions. *J. Mach. Learn. Res.*, 26:Paper No. [209], 80, 2025.
- [2] R. Bellman. *Dynamic Programming*. Princeton Landmarks in Mathematics. Princeton University Press, Princeton, NJ, 1957. Reprinted in 2010 by Princeton University Press.
- [3] J. Berner, L. Richter, and K. Ullrich. An optimal control perspective on diffusion-based generative modeling. *Trans. Mach. Learn. Res.*, 2024.
- [4] K. Black, M. Janner, Y. Du, I. Kostrikov, and S. Levine. Training diffusion models with reinforcement learning. In *ICLR*, 2024.
- [5] Black Forest Labs. FLUX.2: Frontier Visual Intelligence. <https://bf1.ai/blog/flux-2>, 2025.

- [6] K. Clark, P. Vicol, K. Swersky, and D. J. Fleet. Directly fine-tuning diffusion models on differentiable rewards. 2023. arXiv:2309.17400.
- [7] C. Domingo-Enrich, M. Drozdal, B. Karrer, and R. T. Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. In *ICLR*, 2025.
- [8] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Mindere, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.
- [9] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, and F. Boesel. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, pages 12606–12633, 2024.
- [10] Y. Fan and K. Lee. Optimizing DDPM sampling with shortcut fine-tuning. 2023. arXiv:2301.13362.
- [11] Y. Fan, O. Watkins, Y. Du, H. Liu, M. Ryu, C. Boutilier, P. Abbeel, M. Ghavamzadeh, K. Lee, and K. Lee. DPOK: Reinforcement learning for fine-tuning text-to-image diffusion models. In *Neurips*, 2023.
- [12] R. Gao, E. Hoogeboom, J. Heek, V. De Bortoli, K. P. Murphy, and T. Salimans. Diffusion meets flow matching: Two sides of the same coin. 2024. <https://diffusionflow.github.io>.
- [13] Y. Han, M. Razaviyayn, and R. Xu. Stochastic control for fine-tuning diffusion models: Optimality, regularity, and convergence. In *ICML*, pages 21844 – 21870, 2025.
- [14] A. Havens, B. K. Miller, B. Yan, C. Domingo-Enrich, A. Sriram, B. Wood, D. Levine, B. Hu, B. Amos, B. Karrer, X. Fu, G.-H. Liu, and R. T. Q. Chen. Adjoint sampling: Highly scalable diffusion samplers via adjoint matching. 2025. arXiv:2504.11713.
- [15] J. Ho and T. Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.
- [16] Y. Kirstain, A. Polyak, U. Singer, S. Matiana, J. Penna, and O. Levy. Pick-a-Pic: An open dataset of user preferences for text-to-image generation. 2023. arXiv:2305.01569.
- [17] C.-H. Lai, Y. Song, D. Kim, Y. Mitsufuji, and S. Ermon. The principles of diffusion models, 2025. arXiv:2510.21890.
- [18] C. Laidlaw, S. Singhal, and A. Dragan. Correlated proxies: A new definition and improved mitigation for reward hacking. In *ICLR*, 2025.
- [19] J. Lee, J. Chang, J. Kim, and J. C. Ye. Reward score matching: Unifying reward-based fine-tuning for flow and diffusion models. 2026. arXiv:2604.17415.
- [20] J. Li, Y. Cui, T. Huang, Y. Ma, C. Fan, Y. Cheng, M. Yang, Z. Zhong, and L. Bo. MixGRPO: Unlocking flow-based GRPO efficiency with mixed ODE-SDE. 2025. arXiv:2507.21802.
- [21] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. In *ICLR*, 2023.
- [22] G.-H. Liu, J. Choi, Y. Chen, B. K. Miller, and R. T. Chen. Adjoint Schrödinger bridge sampler. In *Neurips*, 2025.
- [23] J. Liu, G. Liu, J. Liang, Y. Li, J. Liu, X. Wang, P. Wan, D. Zhang, and W. Ouyang. Flow-GRPO: Training flow matching models via online RL. 2025. arXiv:2505.05470.
- [24] X. Liu and C. Gong. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023.
- [25] Z. Liu, T. Z. Xiao, C. Domingo-Enrich, W. Liu, and D. Zhang. Value gradient guidance for flow matching alignment. In *Neurips*, 2025.
- [26] N. Ma, M. Goldstein, M. S. Albergo, N. M. Boffi, E. Vanden-Eijnden, and S. Xie. SiT: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *ECCV*, 2024.
- [27] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, and A. Ray. Training language models to follow instructions with human feedback. In *Neurips*, pages 27730–27744, 2022.
- [28] W. Peebles and S. Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.
- [29] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko. *The Mathematical Theory of Optimal Processes*. Interscience Publishers John Wiley & Sons, Inc., New York-London, 1962.
- [30] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763, 2021.

- [31] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, June.
- [32] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, volume 9351 of *Lecture Notes in Computer Science*, pages 234–241, 2015.
- [33] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, P. Schramowski, S. Kundurthy, K. Crowson, L. Schmidt, R. Kaczmarczyk, and J. Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models. 2022. arXiv:2210.08402.
- [34] J. Sheng, H. Zhao, H. Chen, D. D. Yao, and W. Tang. Understanding sampler stochasticity in training diffusion models for RLHF. 2025. arXiv:2510.10767.
- [35] W. Tang, H. V. Tran, and Y. P. Zhang. Policy iteration for the deterministic control problems: a viscosity approach. *SIAM J. Control. Optim.*, 63, 2025.
- [36] W. Tang and H. Zhao. Score-based diffusion models via stochastic differential equations. *Stat. Surv.*, 19:28–64, 2025.
- [37] W. Tang and F. Zhou. Fine-tuning of diffusion models via stochastic control: entropy regularization and beyond. 2026. To appear in *ACC*.
- [38] Team WAN. WAN: Open and advanced large-scale video generative models. 2025. arXiv:2503.20314.
- [39] M. Uehara, Y. Zhao, T. Biancalani, and S. Levine. Understanding reinforcement learning-based fine-tuning of diffusion models: A tutorial and review. 2024. arXiv:2407.13734.
- [40] M. Uehara, Y. Zhao, K. Black, E. Hajiramezanali, G. Scalia, N. L. Diamant, A. M. Tseng, T. Biancalani, and S. Levine. Fine-tuning of continuous-time diffusion models as entropy-regularized control. 2024. arXiv:2402.15194.
- [41] M. Uehara, Y. Zhao, K. Black, E. Hajiramezanali, G. Scalia, N. L. Diamant, A. M. Tseng, T. Biancalani, and S. Levine. Fine-tuning of continuous-time diffusion models as entropy-regularized control. 2024. arXiv:2402.15194.
- [42] B. Wallace, M. Dang, R. Rafailov, L. Zhou, A. Lou, S. Purushwalkam, S. Ermon, C. Xiong, S. Joty, and N. Naik. Diffusion model alignment using direct preference optimization. In *CVPR*, pages 8228–8238, 2024.
- [43] C. Wang, Y. Jiang, C. Yang, H. Liu, and Y. Chen. Beyond reverse KL: Generalizing direct preference optimization with diverse divergence constraints. In *ICLR*, 2024.
- [44] Z. Wang, E. P. Simoncelli, and A. C. Bovik. Multi-scale structural similarity for image quality assessment. In *Conf. Rec. Asilomar Conf. Signals Syst. Comput.*, 2003.
- [45] G. I. Winata, H. Zhao, A. Das, W. Tang, D. D. Yao, S.-X. Zhang, and S. Sahu. Preference tuning with human feedback on language, speech, and vision tasks: a survey. *J. Artificial Intelligence Res.*, 82:2595–2661, 2025.
- [46] X. Wu, Y. Hao, K. Sun, Y. Chen, F. Zhu, R. Zhao, and H. Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. In *ICCV*, 2023.
- [47] J. Xu, X. Liu, Y. Wu, Y. Tong, Q. Li, M. Ding, J. Tang, and Y. Dong. ImageReward: Learning and evaluating human preferences for text-to-image generation. In *Neurips*, pages 15903–15935, 2023.
- [48] Z. Xue, J. Wu, Y. Gao, F. Kong, L. Zhu, M. Chen, Z. Liu, W. Liu, Q. Guo, W. Huang, and P. Luo. DanceGRPO: Unleashing GRPO on visual generation. 2025. arXiv:2505.07818.
- [49] J. Yong and X. Y. Zhou. *Stochastic controls: Hamiltonian systems and HJB equations*, volume 43 of *Applications of Mathematics (New York)*. Springer-Verlag, New York, 1999.
- [50] D. Zhang, Y. Zhang, J. Gu, R. Zhang, J. Susskind, N. Jaitly, and S. Zhai. Improving GFlowNets for text-to-image diffusion alignment. 2024. arXiv:2406.00633.
- [51] Q. Zhang, M. Tao, and Y. Chen. gDDIM: generalized denoising diffusion implicit models. In *ICLR*, 2023.
- [52] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.
- [53] H. Zhao, H. Chen, J. Zhang, D. D. Yao, and W. Tang. Scores as Actions: a framework of fine-tuning diffusion models by continuous-time reinforcement learning. 2024. arXiv:2409.08400.
- [54] H. Zhao, H. Chen, J. Zhang, D. D. Yao, and W. Tang. Scores as Actions: fine tuning diffusion generative models by continuous-time reinforcement learning. In *ICML*, pages 77371 – 77389, 2025.

APPENDIX CONTENTS

Appendix Contents	14
Appendix A. Proof of Deterministic Optimal Control	15
A.1. HJB equation of Deterministic Optimal Control	15
A.2. Pontryagin Maximal Principle(PMP) on Deterministic Control	15
A.3. Proof of Theorem 3.1	16
A.4. Justification to General Time Horizon	16
Appendix B. Extension to Stochastic cases	16
B.1. Stochastic Maximum Principle (SMP)	17
B.2. Stochastic adjoint matching	18
B.3. General Stochastic Adjoint Matching	19
Appendix C. Control intensity throughout denoising timesteps	22
C.1. Variance Exploding (VE) Examples	22
C.2. Variance Preserving (VP) Example	23
C.3. Probability Flow Example	24
Appendix D. Experimental Details	27
D.1. Analysis on SiT Control Intensity	27
D.2. Analysis on SiT Schedules	28
D.3. Performance Evaluation on SiT	30
D.4. Detailed Performance Evaluation on FLUX	31
D.5. How our algorithm accelerates post-training pipeline	33
D.6. Hyper-parameters and details in training	34
D.7. Training curve	35
Appendix E. Additional Images	36

APPENDIX A. PROOF OF DETERMINISTIC OPTIMAL CONTROL

In this section, we derive all conclusions using a more general version of formulation (3.1)

$$\min_u \mathcal{L}(u; \mathbf{X}^u) = \int_0^1 \left(f(\|u(X_t, t)\|) + h(X_t, t) \right) dt + g(X_1), \quad X_0 = x. \quad (\text{A.1})$$

In this main part of this paper, we set $h = 0$ by default.

A.1. HJB equation of Deterministic Optimal Control. Consider the formulation (A.1), deterministic dynamic (3.2) and value function (3.3), we apply the dynamic programming principle on $[t, t + \Delta t]$, then obtain

$$V(x, t) = \min_u \left\{ \left(f(\|u(x, t)\|) + h(x, t) \right) \Delta t + V\left(x + (v^{base}(x, t) + u(x, t))\Delta t, t + \Delta t\right) \right\}.$$

Expanding V to first order in Δt yields

$$0 = \min_u \left[f(\|u\|) + h(x, t) + \partial_t V(x, t) + \nabla_x V(x, t)^\top (v^{base}(x, t) + u) \right].$$

The minimizing control satisfies

$$\nabla_u f(\|u\|) + \nabla_x V(x, t) = 0.$$

Therefore, when f is polynomial (or otherwise explicitly differentiable and invertible in a useful way), one can solve for $u^*(x, t)$ pointwisely from this equation. This establishes the deterministic Hamilton–Jacobi–Bellman equation

$$f(\|u^*(x, t)\|) + h(x, t) + \partial_t V(x, t) + \nabla_x V(x, t)^\top (v^{base}(x, t) + u^*(x, t)) = 0, \quad V(x, 1) = g(x).$$

A.2. Pontryagin Maximal Principle(PMP) on Deterministic Control. The deterministic Hamiltonian of Problem (A.1) under dynamic (3.2) is

$$H(t, x, u, a) = a^\top \left(v^{base}(x, t) + u(x, t) \right) + f(\|u\|) + h(x, t).$$

The Pontryagin maximum principle gives the forward-backward system

$$\begin{cases} \dot{X}_t = \nabla_a H(t, x, u, a) = v^{base}(x, t) + u(x, t), \\ \dot{a}_t = -\nabla_x H(t, x, u, a) = -\left[a^\top \nabla_x v^{base}(x, t) + \nabla_x h(x, t) \right], \\ a_1 = \nabla_{X_1} g(X_1). \end{cases} \quad (\text{A.2})$$

For u^* to be the optimal control, the necessary first order condition is that

$$0 = \nabla_u H(t, x, u, a) = a + \nabla_u f(\|u\|).$$

a_t is the costate and can be defined as gradient of value function, *i.e.* $\nabla_x V(x, t)$, which perfectly aligns with Appendix A.1. That is under deterministic optimal control perspective, HJB and PMP give the same relationship between optimal control and value function. Therefore, lean adjoint \tilde{a}_t acts as a reasonable surrogate in Algorithm 1.

A.3. Proof of Theorem 3.1. The result follows from PMP in a straightforward way. In our setup given by (A.2), since the optimal control minimizes H pointwise, we have

$$\nabla_u H = f'(\|u\|) \frac{u}{\|u\|} + a = 0.$$

Thus u is colinear with $-a$. Let $r = \|u\|$. Taking norms gives

$$f'(r) = \|a\|.$$

By invertibility of f' , we obtain

$$r = (f')^{-1}(\|a\|).$$

Substituting back yields

$$u^* = -\frac{(f')^{-1}(\|a\|)}{\|a\|} a.$$

Typically when $f(x) = \frac{1}{p\lambda}x^p$, then $f'(x) = x^{p-1}/\lambda$, we have $(f')^{-1}(y) = \lambda^{\frac{1}{p-1}}y^{\frac{1}{p-1}}$. Substituting $y = \|a_t\|$ into Theorem 3.1 yields

$$u_t^* = -\frac{(f')^{-1}(\|a_t\|)}{\|a_t\|} a_t = -\lambda^{\frac{1}{p-1}} \|a_t\|^{\frac{2-p}{p-1}} a_t.$$

A.4. Justification to General Time Horizon. By the deterministic PMP theorem, we apply (A.2) on the truncated interval $[t, 1]$, and the adjoint variable satisfies a same formulation

$$\dot{a}_s = -\nabla_x H(X_s, u_s^*, a_s, s), \quad a_1 = -\nabla g(X_1).$$

For all $t \leq s \leq 1$, we have

$$\nabla_u H(X_s, u_s^*, a_s, s) = 0.$$

With a similar argument to Appendix A.3, we yields

$$u_s^* = -\frac{(f')^{-1}(\|a_s\|)}{\|a_s\|} a_s.$$

APPENDIX B. EXTENSION TO STOCHASTIC CASES

In this section we consider the stochastic control problem

$$dX_t = (b(X_t, t) + \sigma(t)u(X_t, t))dt + \sigma(t)dB_t,$$

with objective

$$J(u) = \mathbb{E} \left[\int_0^1 \left(\frac{1}{2} \|u(X_t, t)\|^2 + h(X_t, t) \right) dt + g(X_1) \right], \quad X_0 \text{ given.}$$

Here B_t is a Brownian motion and control u is adapted to the filtration generated by the noise and observations. For clear illustration, in stochastic case we only apply quadratic penalty $\frac{1}{2}\|u_t\|^2$, general form can also be derived similarly as in deterministic optimal control.

B.1. Stochastic Maximum Principle (SMP). For the controlled SDE, the stochastic Hamiltonian is

$$H(t, x, u, p, q) = p^\top (b(x, t) + \sigma(t)u) + \text{Tr}(q^\top \sigma(t)) + \frac{1}{2}\|u\|^2 + h(x, t).$$

The classical stochastic maximum principle introduces a pair of adjoint processes (p_t, q_t) solving the BSDE

$$dp_t = -\nabla_{X_t} H(t, X_t, u_t, p_t, q_t)dt + q_t dB_t, \quad p_1 = \nabla_{X_1} g(X_1).$$

Since $\sigma = \sigma(t)$ is independent of x , we have

$$dp_t = -\left[p_t^\top \nabla_x b(x, t) + \nabla_x h(x, t)\right]dt + q_t dB_t, \quad p_1 = \nabla_{X_1} g(X_1).$$

The optimality condition is

$$0 = \nabla_u H(t, X_t, u_t, p_t, q_t) = \sigma(t)^\top p_t + u_t,$$

so that

$$u_t^* = -\sigma(t)^\top p_t.$$

On a historical note, adjoint variables (or costates) are Lagrangian multipliers in control theory, representing the sensitivity of the optimal cost to changes in state. In the stochastic control setting, the adjoint variables are a pair: first-order value gradient p_t and second-order correction q_t from noise, which are specified by a system of backward stochastic differential equations derived from the maximum principle [49]. The ‘‘adjoint’’ defined in [7] is an estimate of the first-order value gradient by averaging out the noise, and can thus be viewed as a degenerate form in standard control theory. On the other hand, such-defined adjoint in deterministic control is exactly the one derived from the maximum principle. Next we will explain the intuition for both p_t and q_t .

Intuition for p_t . In the stochastic maximum principle, the adjoint process p_t can be interpreted as the sensitivity of the future **minimal** cost with respect to the current state X_t . The value function $V(t, x)$ in SOC setting is defined as the minimum **expected** future cost starting from state x at time t :

$$V(t, x) := \min_u \mathbb{E} \left[\int_t^1 \left(\frac{1}{2}\|u_s\|^2 + h(X_s, s) \right) ds + g(X_1) \mid X_t = x \right],$$

More precisely, if the value function $V(x, t)$ is sufficiently smooth, then along the optimal trajectory one has

$$p_t = \nabla_x V(x, t).$$

This is exactly the same as adjoint a_t in deterministic optimal control PMP setting. If p_t is large in some direction, then moving the current state in that direction will significantly increase the future cost. In the control-affine case

$$dX_t = (b(X_t, t) + \sigma(t)u_t)dt + \sigma(t)dB_t,$$

with quadratic control cost, the optimality condition gives

$$u_t^* = -\sigma(t)^\top p_t.$$

Thus, the optimal control pushes the state in the direction that decreases the future cost.

Intuition for q_t . The process p_t is itself random, because it is evaluated along the stochastic trajectory X_t driven by Brownian Motion B_t . To understand the role of q_t , suppose again that the value function is smooth and

$$p_t = \nabla_x V(x, t).$$

Applying Itô's formula to p_t , gradient of value function, we obtain

$$dp_t = \left[\partial_t \nabla_x V(x, t) + \nabla_x^2 V(x, t) (b(x, t) + \sigma(t)u_t) + \frac{1}{2} \text{Tr} \left(\sigma(t)\sigma(t)^\top \nabla_x^3 V(x, t) \right) \right] dt + \nabla_x^2 V(x, t) \sigma(t) dB_t.$$

On the other hand, the stochastic maximum principle writes the adjoint equation as

$$dp_t = -H_x(t, X_t, u_t, p_t, q_t) dt + q_t dB_t.$$

Comparing the martingale parts of the two expressions, we get

$$q_t = \nabla_x^2 V(x, t) \sigma(t).$$

Hence q_t is the Brownian noise coefficient of the adjoint process p_t . Since p_t represents the gradient of the future cost, q_t describes how this gradient changes along the stochastic noise directions.

B.2. Stochastic adjoint matching. Adjoint Matching reformulates the stochastic optimal control problem as a regression-style fixed-point problem. For the controlled process

$$dX_t = \left(b(X_t, t) + \sigma(t)u(X_t, t) \right) dt + \sigma(t)dB_t,$$

define the cost-to-go under a fixed control u by

$$J(u; x, t) := \mathbb{E}_{X \sim p^u} \left[\int_t^1 \left(\frac{1}{2} \|u(X_s, s)\|^2 + h(X_s, s) \right) ds + g(X_1) \mid X_t = x \right].$$

The value function is then

$$V(x, t) = \min_u J(u; x, t) = J(u^*; x, t),$$

and the optimal control satisfies

$$u^*(x, t) = -\sigma(t)^\top \nabla_x V(x, t) = -\sigma(t)^\top \nabla_x J(u^*; x, t).$$

Instead of constructing an importance-weighted estimator of u^* , Adjoint Matching directly regresses the current control toward

$$-\sigma(t)^\top \nabla_x J(u; x, t),$$

so that the optimal control is characterized as the fixed point

$$u(x, t) = -\sigma(t)^\top \nabla_x J(u; x, t).$$

Pathwise adjoint versus the SMP adjoint. The adjoint used in Adjoint Matching should not be confused with the classical stochastic maximum principle adjoint (p_t, q_t) . For a sampled trajectory $X \sim p^u$, this work[7] defines the pathwise adjoint

$$a(t; \mathbf{X}^u, u) := \nabla_{X_t} \left[\int_t^1 \left(\frac{1}{2} \|u(X_s, s)\|^2 + h(X_s, s) \right) ds + g(X_1) \right].$$

This quantity is the sensitivity of the realized remaining cost along one trajectory with respect to the intermediate state X_t . It is therefore a pathwise sensitivity, not the BSDE adjoint in

the stochastic maximum principle. Its conditional expectation recovers the gradient of the fixed-control cost-to-go:

$$\mathbb{E}_{X \sim p^u} [a(t; \mathbf{X}^u, u) \mid X_t = x] = \nabla_x J(u; x, t).$$

Consequently, only at the optimal control $u = u^*$ do we obtain

$$\mathbb{E}_{X \sim p^{u^*}} [a(t; \mathbf{X}^u, u^*) \mid X_t = x] = \nabla_x J(u^*; x, t) = \nabla_x V(x, t).$$

Thus the pathwise adjoint used by Adjoint Matching becomes an estimator of the value-gradient adjoint only at the optimal fixed point.

Basic Adjoint Matching objective. Using the pathwise adjoint as a stochastic target, the basic Adjoint Matching loss is

$$\mathcal{L}_{\text{Basic-AM}}(u; X) = \frac{1}{2} \int_0^1 \left\| u(X_t, t) + \sigma(t)^\top a(t; X, \bar{u}) \right\|^2 dt, \quad X \sim p^{\bar{u}},$$

where $\bar{u} = \text{stopgrad}(u)$. This objective can be viewed as a consistency loss for the fixed-point equation

$$u(x, t) = -\sigma(t)^\top \nabla_x J(u; x, t).$$

Adjoint ODE in Stochastic AM can be derived as

$$\begin{aligned} \frac{d}{dt} a(t; \mathbf{X}^u, u) &= - \left[a^\top \nabla_x v^{\text{base}}(x, t) + a^\top \sigma(t) \nabla_x u(x, t) + \nabla_x h(x, t) + u(x, t)^\top \nabla_x u(x, t) \right] \\ a(1; X) &= \nabla_{X_1} g(X_1) \end{aligned}$$

At the optimal control, Adjoint Matching uses the fixed-point relation

$$u^*(x, t) = \mathbb{E} \left[-\sigma(t)^\top a(t; \mathbf{X}^u, u^*) \mid X_t = x \right].$$

Multiplying by $\nabla_x u^*(x, t)$ gives

$$\mathbb{E} \left[u^*(x, t)^\top \nabla_x u^*(x, t) + a(t; \mathbf{X}^u, u^*)^\top \sigma(t) \nabla_x u^*(x, t) \mid X_t = x \right] = 0.$$

These are exactly the two u -dependent Jacobian terms in the full pathwise adjoint ODE. Thus, at the optimum, their conditional expectation vanishes. The lean adjoint drops these terms and uses

$$\frac{d}{dt} \tilde{a}(t; \mathbf{X}) = - \left[\tilde{a}(t; \mathbf{X})^\top \nabla_x v^{\text{base}}(X_t, t) + \nabla_x h(X_t, t) \right], \quad \tilde{a}(1; X) = \nabla_{X_1} g(X_1).$$

Hence \tilde{a} is not generally the true pathwise gradient; it is a simplified surrogate adjoint that preserves the optimal fixed point while avoiding the expensive $\nabla_x u$ terms.

B.3. General Stochastic Adjoint Matching. The memoryless schedule $\sigma(t)^2 = 2\eta_t$ (see Theorem 1, [7]) is a sufficient theoretical condition for robust fine-tuning results, as it cancels the initial-value bias and yields the tilted terminal law $p^*(X_1) \propto p_{\text{base}}(X_1) \exp(r(X_1))$.

However, this condition is not necessarily the best choice to implement in finite-step training: since the memoryless schedule injects very large noise near early denoising times, it may over-explore highly noisy states and weaken useful reward propagation in practice. To implement this memoryless noise schedule, we have to clip near the initial noise and decrease step size for higher accuracy.

Theorem B.1. Denote $\kappa_t := \frac{\alpha'_t}{\alpha_t}$ and $\eta_t := \beta_t(\frac{\alpha'_t}{\alpha_t}\beta_t - \beta'_t)$, where α_t and β_t are schedules define in refence flow (2.1). Suppose we already have a flow model v^{base} or score estimator s^{base} , equivalent backward sampling SDEs with noise schedule $\sigma(t)$ are

$$\begin{aligned} dX_t &= \left[\kappa_t x + \left(\frac{\sigma^2(t)}{2} + \eta_t \right) s^{\text{base}}(X_t, t) \right] + \sigma(t) dB_t \\ &= \underbrace{\left[\left(1 + \frac{\sigma^2(t)}{2\eta_t} \right) v^{\text{base}}(X_t, t) - \frac{\sigma^2(t)\kappa_t}{2\eta_t} X_t \right]}_{b(X_t, t)} dt + \sigma(t) dB_t \end{aligned}$$

where the base model satisfies $v^{\text{base}}(x, t) = \kappa_t x + \eta_t s^{\text{base}}(x, t)$. Assume $\sigma(t) > 0$, $\eta_t > 0$, and the controlled model u_θ is obtained by the difference between finetuned model and base model. Then the induced stochastic optimal control for

$$dX_t = [b(X_t, t) + \sigma(t)u_\theta(X_t, t)]dt + \sigma(t)dB_t, \quad (\text{B.1})$$

is

$$u_\theta(x, t) = \frac{\sigma(t)^2 + 2\eta_t}{2\sigma(t)} (s_\theta(x, t) - s^{\text{base}}(x, t)) \quad (\text{B.2})$$

$$= \frac{\sigma(t)^2 + 2\eta_t}{2\sigma(t)\eta_t} (v_\theta(x, t) - v^{\text{base}}(x, t)). \quad (\text{B.3})$$

Proof sketch. According to Equation (10), (11) in [7], the unified base SDE can be written as

$$b^{\text{base}}(x, t) = \kappa_t x + \left(\frac{\sigma(t)^2}{2} + \eta_t \right) s^{\text{base}}(x, t).$$

After fine-tuning, the drift becomes

$$b_\theta(x, t) = \kappa_t x + \left(\frac{\sigma(t)^2}{2} + \eta_t \right) s_\theta(x, t).$$

Thus the drift difference is

$$b_\theta - b^{\text{base}} = \left(\frac{\sigma(t)^2}{2} + \eta_t \right) (s_\theta - s^{\text{base}}).$$

In the control-affine SOC form, this difference must equal $\sigma(t)u_\theta$:

$$\sigma(t)u_\theta = \left(\frac{\sigma(t)^2}{2} + \eta_t \right) (s_\theta - s^{\text{base}}).$$

Hence

$$u_\theta = \frac{\sigma(t)^2 + 2\eta_t}{2\sigma(t)} (s_\theta - s^{\text{base}}) = \frac{\sigma(t)^2 + 2\eta_t}{2\sigma(t)\eta_t} (v_\theta - v^{\text{base}}).$$

Adjoint Matching regresses the current control onto the lean adjoint target $-\sigma(t)^\top \tilde{a}_t$, so substituting this expression for u_θ gives the stated loss. \square

Corollary B.2. *the Adjoint Matching loss*

$$\mathcal{L}_{\text{AM}}(\theta) = \mathbb{E} \int_0^1 \left\| \frac{\sigma(t)^2 + 2\eta_t}{2\sigma(t)\eta_t} (v_\theta(X_t, t) - v^{\text{base}}(X_t, t)) + \sigma(t)^\top \tilde{a}_t \right\|^2 dt$$

is the least-squares matching objective for the control u_θ under an arbitrary non-memoryless noise schedule.

Algorithm 2 is the designed for KL-regularization stochastic adjoint matching with general noise schedule.

Algorithm 2 Stochastic Adjoint Matching with General Noise Schedule

Require: Pre-trained velocity field $v^{\text{base}}(x, t)$, noise schedule $\sigma(t)$, step size h , number of iterations N , batch size m .

1: Initialize $v_\theta \leftarrow v^{\text{base}}$

2: **for** $n = 0, \dots, N - 1$ **do**

3: Sample m trajectories $\{X_t\}_{t \in \{0, h, \dots, 1\}}$ using current model via:

$$X_{t+h} = X_t + h \left(v_\theta(X_t, t) + \frac{\sigma(t)^2}{2\eta_t} (v_\theta(X_t, t) - \kappa_t X_t) \right) + \sqrt{h} \sigma(t) \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, I)$$

4: Solve lean adjoint ODE backward:

$$\tilde{a}_{t-h} = \tilde{a}_t + h \tilde{a}_t^\top \nabla_{X_t} \left(v^{\text{base}}(X_t, t) + \frac{\sigma(t)^2}{2\eta_t} (v^{\text{base}}(X_t, t) - \kappa_t X_t) \right), \quad \tilde{a}_1 = -\nabla_{X_1} r(X_1)$$

Note that both X_t and \tilde{a}_t should be computed without gradients, *i.e.* Stop gradients:

$X_t = \text{stopgrad}(X_t)$, $\tilde{a}_t = \text{stopgrad}(\tilde{a}_t)$.

5: Compute loss:

$$\mathcal{L}(\theta) = \frac{h}{m} \sum_t \left\| \frac{\sigma(t)^2 + 2\eta_t}{2\sigma(t)\eta_t} (v_\theta(X_t, t) - v_{\text{base}}(X_t, t)) + \sigma(t)\tilde{a}_t \right\|^2$$

6: Compute the gradient $\nabla_\theta \mathcal{L}$ and update θ

7: **end for**

APPENDIX C. CONTROL INTENSITY THROUGHOUT DENOISING TIMESTEPS

In this section, we are motivated to train stochastic and deterministic controls with adaptive importance weights on denoising time steps. To explain the efficiency of our adaption, we analyze the tractable inconsistent monotonicity trend of control norms. For simplicity, we illustrate with 1D Gaussian priors, in which the noise schedules and the log-likelihoods of process distributions (the “score” functions) jointly contribute to unique control intensity maxima. Then we move on to the ODE analysis, in which the multi-modal mix-Gaussian priors become the principal factor that steers control intensity.

Proposition C.1. *Consider a one-dimensional backward process with Brownian noise schedule $\beta(\cdot)$: $dX_t = b(X_t, T - t) dt + \eta\sqrt{2\beta(T - t)} dB_t$, in which $\eta \geq 0$ is a free parameter on injected noise, differentiable reward function $r(\cdot)$ is evaluated on X_T at the terminal backward denoising time. If the associated adjoint [7] evolves according to*

$$\frac{d}{dt}a(X_t, t) = -\nabla_X b(X_t, T - t) a(X_t, t), \quad a(X_T, T) = -\nabla r(X_T).$$

Then the optimal control admits the form

$$u^*(X_t, t) = c_\beta^*(t, T, \eta) \nabla r(X_T),$$

in which we define the time-aware component

$$c_\beta^*(t, T, \eta) = \eta\sqrt{2\beta(T - t)} \exp\left(\int_t^T \nabla_X b(X_s, T - s) ds\right).$$

Remark. Consequently, the monotonicity of $|u^*(X_t, t)|$ is determined jointly by the noise schedule β and the accumulated drift Jacobian $\nabla_X b$. In particular, if $c_\beta^*(t, T, \eta)$ has a unique critical point $t^* \in (0, T)$, with

$$\frac{d}{dt}c_\beta^*(t, T, \eta) > 0 \quad \text{for } t < t^*, \quad \frac{d}{dt}c_\beta^*(t, T, \eta) < 0 \quad \text{for } t > t^*,$$

then both $|u^*(X_t, t)|$ and $\mathbb{E}|u^*(X_t, t)|$ exhibits a unique maximum in $t^* \in [0, T]$.

C.1. Variance Exploding (VE) Examples. Assume we start from a simplest Gaussian model $Y_0 \sim \mathcal{N}(0, 1)$ and consider its corresponding 1D Variance Exploding (VE) forward process $dY_t = \sqrt{2t} dB_t$. The VE trajectory $Y_t = Y_0 + \int_0^t \sqrt{2s} dB_s \sim \mathcal{N}(0, 1 + t^2)$ follows immediately from our strong assumption on $\text{Law}(Y_0)$, and its score function exhibits a closed form expression: $\nabla \log p(Y_t) = -\frac{Y_t}{1+t^2}$. If we fix a finite time horizon T , the backward process $X_t := Y_{T-t}$ of our VE model is given in the form of gDDIM[51]:

$$dX_t = -\frac{(1 + \eta^2)(T - t)}{1 + (T - t)^2} X_t dt + \eta\sqrt{2(T - t)} dB_t$$

where $X_0 \sim \mathcal{N}(0, 1 + T^2)$. By solving the t -linear ODE from Proposition C.1, we obtain

$$a(X_t, t) = a(X_T, T)[1 + (T - t)^2]^{-(1+\eta^2)/2} = -\nabla r(X_T)[1 + (T - t)^2]^{-(1+\eta^2)/2}.$$

Therefore, the time-aware component of optimal control follows

$$c_{VE}^*(X_t, t) = \eta\sqrt{2(T - t)} \left(\frac{(T - t)}{[1 + (T - t)^2]^{(1+\eta^2)}}\right)^{1/2}, \quad \arg \max_t c_{VE}^* = T - \frac{1}{\sqrt{1 + 2\eta^2}}.$$

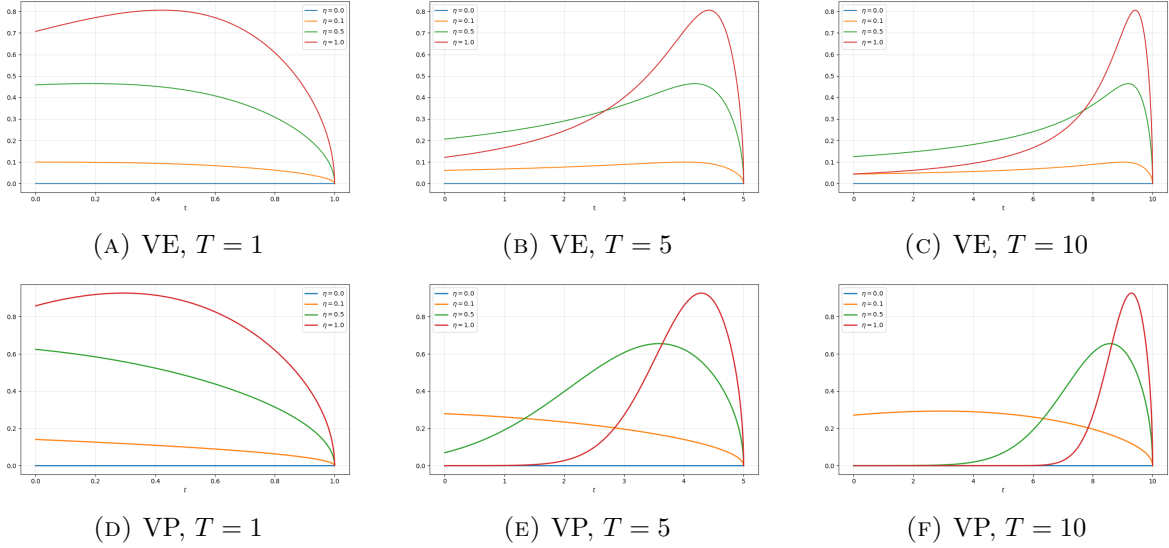


FIGURE 6. Curves of $c^*(t, T, \eta)$ versus denoising time t under different terminal horizons T and stochasticity levels η in the 1D VE (top row) and VP (bottom row) cases. Data variance is set to 1.

To illustrate the inconsistent monotonicity of u^* , we plot the norm of c^* against denoising time $t \in [0, T]$. We see from Figure 6 that the maximum value of $c^*(t, T, \eta)$ is more heavily tilted towards terminal time-steps for larger T , an observation consistent with Equation (C.1).

C.2. Variance Preserving (VP) Example. Similarly, for VP model $dY_t = -tY_t dt + \sqrt{2t}dB_t$, $Y_0 \sim \mathcal{N}(0, 1)$, the marginal distribution is standard Gaussian $\mathcal{N}(0, 1)$. Therefore, its score function $\nabla \log p(Y_t) = -Y_t$, and the backward process of VP model is given in the form of gDDIM[51]:

$$dX_t = -\eta^2(T-t)X_t dt + \eta\sqrt{2(T-t)}dB_t.$$

With Proposition C.1,

$$a(X_t, t) = a(X_T, T)e^{-\frac{\eta^2}{2}(T-t)^2} = -\nabla r(X_T)e^{-\frac{\eta^2}{2}(T-t)^2},$$

and the time-aware component of optimal control follows

$$c_{VP}^*(X_t, t) = \eta\sqrt{2(T-t)} \cdot e^{-\frac{\eta^2}{2}(T-t)^2}, \quad \arg \max_t c_{VP}^* = T - \frac{1}{\sqrt{2\eta}} \quad (\text{C.1})$$

Similar to 1D VE case, we plot the curve of c^* and t in Figure 6. A similar argument can be applied to a bi-modal data prior as well:

Proposition C.2. *Let $(X_t)_{0 \leq t \leq T}$ follow the ODE flow $dX_t = b(X_t, t) dt$, in which $p_{noise} = X_0 \sim \mathcal{N}(0, T^2)$ and $p_{data} = \bar{X}_1 \sim \mathcal{N}(-\mu, 1) + \mathcal{N}(\mu, 1)$. ($1 \ll \mu \ll T$). Then the control norm has a unique extreme value at $t \in [0, T]$.*

Proof. Suppose $X_t = X_0 + \int_0^t \sqrt{2t} dt$, $X_t \sim \mathcal{N}(-\mu, 1 + t^2) + \mathcal{N}(\mu, 1 + t^2)$, which gives

$$p(t, x) = \frac{1}{2\sqrt{2\pi(1+t^2)}} \left(\exp\left(-\frac{(x-\mu)^2}{2(1+t^2)}\right) + \exp\left(-\frac{(x+\mu)^2}{2(1+t^2)}\right) \right),$$

and

$$s(t, x) := \nabla_x \log p(t, x) = \frac{-x + \mu \tanh\left(\frac{\mu x}{1+t^2}\right)}{1+t^2}.$$

When $t = 0$, for $0 < x < O(\mu)$, $s(0, x) = -x + \mu \tanh(\mu x) > 0$. But $s(x, T) \approx -\frac{x}{1+T^2} < 0$. Therefore, for multimodal data, the control norm increases at $t = T$ and decreases when $t \rightarrow 0$. \square

C.3. Probability Flow Example. To motivate the empirical effectiveness of adaptive AM algorithms on flow-based models, we study the following probability flow example. In this elementary setting, we explicitly calculate the control norm intensity, which attains a unique maximum at some later denoising time-step.

Moreover, we discover that higher-order polynomial regularization $f(r) = \lambda r^p/p$ preserves the same peak location but flattens the normalized control curves, so a broader later denoising time region remains close to the maximal control strength. This improves second-order truncated adjoint matching by keeping the neighboring control signals effective as well.

Proposition C.3. *Let $X_t = (1-t)X_0 + tX_1$, where $X_0 \sim \mathcal{N}(\mu, \sigma^2)$ and $X_1 \sim \mathcal{N}(0, 1)$, then the control norm takes a unique extreme value at $t^* = \frac{\sigma^2}{1+\sigma^2}$.*

Proof. Observe that X_t is the linear combination of 2 i.i.d Gaussian, so

$$X_t \sim \mathcal{N}((1-t)\mu, (1-t)^2\sigma^2 + t^2)$$

and we denote $(1-t)\mu := m(t)$ and $(1-t)^2\sigma^2 + t^2 := D(t)$. Since $(X_1 - X_0, X_t)$ is joint Gaussian,

$$\begin{aligned} v(t) &= \mathbb{E}[X_1 - X_0 | X_t = x] = \mathbb{E}[X_1 - X_0] + \frac{\text{Cov}(X_1 - X_0, X_t)}{\text{Var}(X_t)}(x - m(t)) \\ &= -\mu + \frac{t - (1-t)\sigma^2}{D(t)}(x - m(t)) \\ &:= A(t)x + B(t). \end{aligned}$$

Since $\nabla_x v_t(x) = A(t)$ and with lean adjoint formulation,

$$\begin{aligned} a(t) &= a(1) \exp\left(-\int_t^1 A(s) ds\right) \\ &= a(1) \exp\left(-\frac{1}{2} \int_t^1 \frac{D'(s)}{D(s)} ds\right) \\ &= a(1) \exp\left(-\frac{1}{2} \log D(t)\right) \\ &= \frac{a(1)}{\sqrt{(1-t)^2\sigma^2 + t^2}}. \end{aligned}$$

\square

Proposition C.4 (Normalized temporal control strength). *With a same setup as Proposition C.3, for generalized polynomial regularization $f(r) = \frac{1}{p}r^p$ ($p > 1$), the optimal lean-adjoint control satisfies $\|u_t^*\| \propto D(t)^{-\frac{1}{2(p-1)}}$. Let*

$$t^* = \frac{\sigma^2}{1 + \sigma^2}$$

be the unique maximizer of $\|u_t^*\|$. Then the normalized relative control strength

$$R_p(t) := \frac{\|u_t^*\|}{\max_{s \in [0,1]} \|u_s^*\|}$$

is given by

$$R_p(t) = \left(\frac{D(t^*)}{D(t)} \right)^{\frac{1}{2(p-1)}} = \left(\frac{\frac{\sigma^2}{1+\sigma^2}}{(1-t)^2\sigma^2 + t^2} \right)^{\frac{1}{2(p-1)}}.$$

Remark C.1. In particular, $R_p(t^*) = 1$ and $R_p(t) \leq 1$ for all $t \in [0, 1]$. Moreover, for every fixed $t \neq t^*$, $R_p(t)$ is increasing in p . Therefore larger p yields a flatter normalized control profile and keeps a broader time region close to the maximal control strength.

Proof. From the Gaussian interpolation calculation,

$$\|a(t)\| = \frac{\|a(1)\|}{\sqrt{D(t)}}.$$

For $f(r) = r^p/(p\lambda)$, the optimality condition gives

$$\|u_t^*\| = \lambda^{\frac{1}{p-1}} \|a(t)\|^{\frac{1}{p-1}} = C_p D(t)^{-\frac{1}{2(p-1)}},$$

where $C_p = \lambda^{1/(p-1)} \|a(1)\|^{1/(p-1)}$ is independent of t . Thus maximizing $\|u_t^*\|$ is equivalent to minimizing $D(t)$. Since

$$D'(t) = -2(1-t)\sigma^2 + 2t,$$

the unique critical point is

$$t^* = \frac{\sigma^2}{1 + \sigma^2},$$

and

$$D(t^*) = \frac{\sigma^2}{1 + \sigma^2}.$$

Dividing $\|u_t^*\|$ by its maximum gives

$$R_p(t) = \left(\frac{D(t^*)}{D(t)} \right)^{\frac{1}{2(p-1)}}.$$

Since $D(t) \geq D(t^*)$, we have $R_p(t) \leq 1$. Finally, for $t \neq t^*$,

$$0 < \frac{D(t^*)}{D(t)} < 1,$$

so decreasing the exponent $1/(2(p-1))$ increases $R_p(t)$. Hence $R_p(t)$ is increasing in p . \square

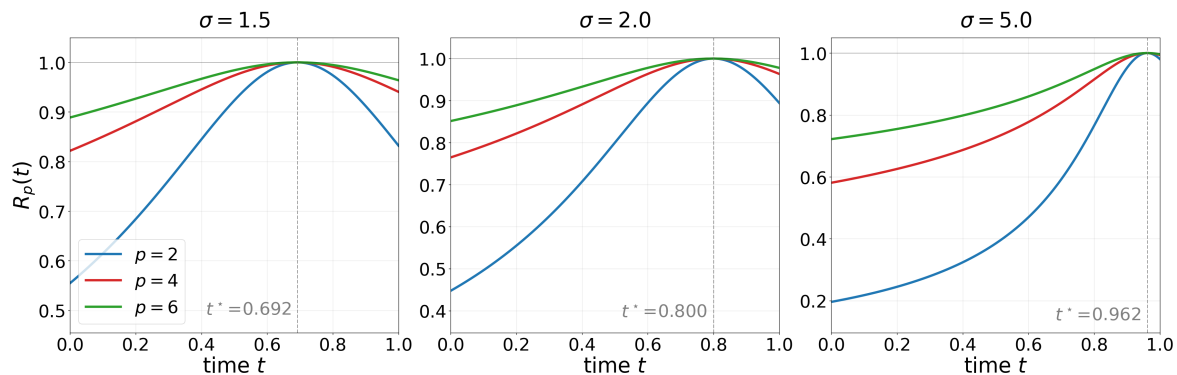


FIGURE 7. Normalized relative control strength $R_p(t)$ for the polynomial regularizer with order $p \in \{2, 4, 6\}$ and three noise levels $\sigma \in \{1.5, 2.0, 5.0\}$.

According to Figure 7, The peak location $t^* = \sigma^2/(1 + \sigma^2)$ migrates toward $t = 1$ as σ grows (vertical dashed line in each panel), reflecting that the informative control mass concentrates near the deterministic endpoint X_1 when X_0 is highly diffuse. Across all σ , raising the exponent p flattens the normalized profile and widens the time region in which $\|u_t^*\|$ stays close to its maximum; the effect is modest at $\sigma = 1.5$ but dramatic at $\sigma = 5.0$, where $R_2(0) \approx 0.20$ while $R_6(0) \approx 0.72$.

APPENDIX D. EXPERIMENTAL DETAILS

D.1. **Analysis on SiT Control Intensity.** At each denoising step along a sampling trajectory of **SiT-XL/2**, we compute the per-step control $u_{t_k} = v_\theta(x_{t_k}, t_k) - v_{\text{base}}(x_{t_k}, t_k)$ and record its scaled L_2 magnitude $\|u_{t_k}\|_2 \Delta t$ for each sample in a training batch.

The i -th iteration panel reports per-step controls measured *immediately after* the i -th gradient update, e.g. “Iter 0” column shows the controls produced by the model after one optimization step, which explains the small fluctuation around zero in that row. As fine-tuning progresses, the controls grow into a coherent profile, but early steps ($t \lesssim 0.3$) consistently accumulate less mass than later ones. This suggests that the early portion of the trajectory contributes little reward-relevant signal even after 250 iterations and is the direct empirical motivation for late-horizon truncation.

The same observation also explains why a higher-order ODE AM helps. **6th ODE-AM-Full** sustains non-trivial control through $t = 1$ where **2nd ODE-AM-Full** collapses, achieving *implicitly* the late-horizon emphasis that **Truncate-12** enforces *explicitly*.

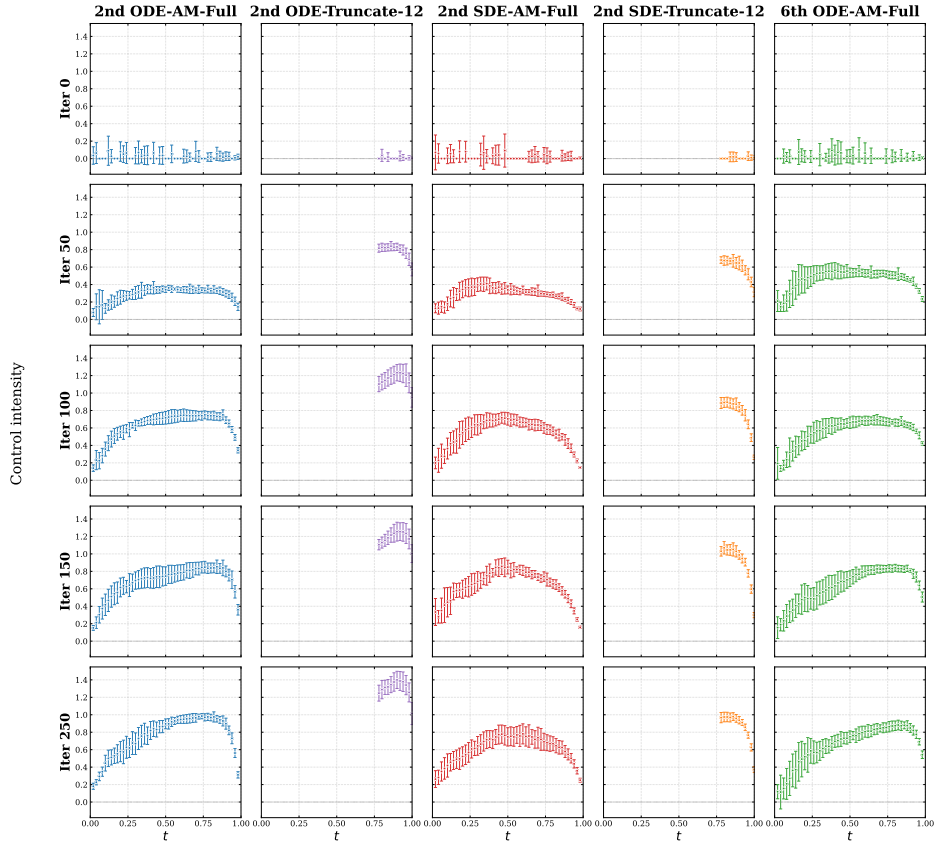


FIGURE 8. Control intensity along the diffusion trajectory across fine-tuning iterations (rows) for five adjoint matching variants (columns) on **SiT-XL/2**.

D.2. Analysis on SiT Schedules. In SiT, every sample is constructed as a stochastic interpolant $x_t = \alpha(t)x_1 + \sigma(t)x_0$ between data x_1 and Gaussian noise $x_0 \sim \mathcal{N}(0, I)$, with α and σ jointly fixing the marginal of x_t at every denoising time $t \in [0, 1]$ (here $t = 0$ is noise and $t = 1$ is data). Because this marginal is preserved by a whole family of reverse-time SDEs, inference is free to inject extra Gaussian noise on top of the deterministic flow,

$$dx_t = \left[v_\theta(x_t, t) + \frac{1}{2} w_t s_\theta(x_t, t) \right] dt + \sqrt{w_t} d\bar{W}_t,$$

where $w_t \geq 0$ controls how much fresh noise is mixed in.

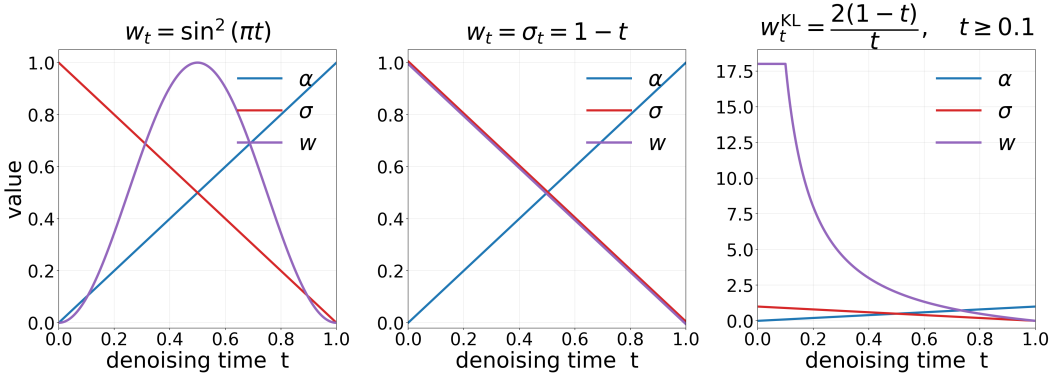


FIGURE 9. Interpolant coefficients $\alpha(t)$, $\sigma(t)$, and three noise schedules w_t on the Linear path $\{(\alpha_t, \sigma_t)\}_{t=0}^1 = \{(t, 1-t)\}_{t=0}^1$

Although the KL family wins for **pretraining** (Table 6, [26]), in AM finetune, w_t^{KL} near $t=0$ blows up gradients in the most chaotic part of the trajectory, while $w_t = 1-t$ keeps injecting noise right at the early stages where exploration is not very useful (see discussion in Appendix D.1). $\sin^2(\pi t)$ avoids both pathologies and gives the best empirical reward fine-tuning performance (best reward value with best sample diversity) among the three.

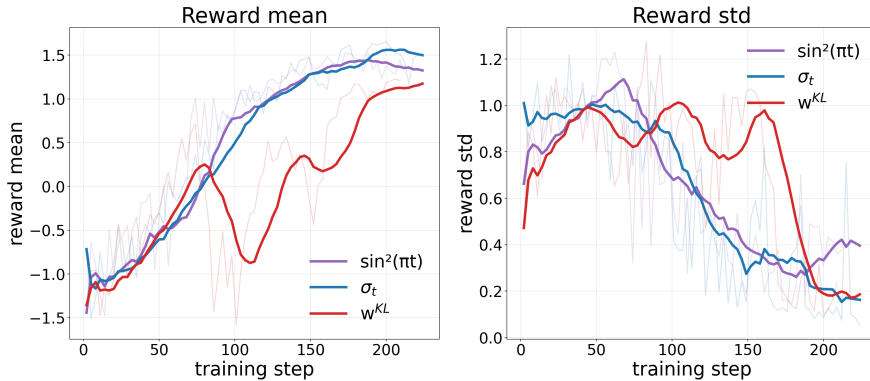


FIGURE 10. ImageReward mean (left) reward std (right) of finetuned **SiT-XL/2 + SDE-AM-Full** under the three SDE schedules; bold lines are 10-step rolling averages.

D.2.1. *Discussion on Memoryless Schedule.* The schedule w_t^{KL} coincides exactly with the memoryless noise schedule $\sigma_{\text{AM}}(t)^2 = 2\eta_t$ formalized in [7], where $\eta_t = \beta_t(\alpha'_t\beta_t/\alpha_t - \beta'_t)$. According to [7, Theorem 1], this schedule is *necessary* for the fine-tuned model to converge to the tilted distribution $p^*(X_1) \propto p_{\text{base}}(X_1) \exp(r(X_1))$ in continuous time, since it is the unique choice that $X_0 \perp X_1$ and thereby removes the value-function bias at $t = 0$.

By this criterion σ_t and $\sin^2(\pi t)$ are biased. However, empirically we find that the correct injection of noise outweighs this bias (Fig. 10). Both schedules deliver the right amount of stochasticity in the middle and later stages of the trajectory to encourage exploration, while keeping noise bounded at $t \leq 0.3$ so the optimization stays stable and avoids reward collapse. The truncated w_t^{KL} , by contrast, concentrates its stochasticity precisely where the adjoint signal is most fragile and is not ideal in practice.



FIGURE 11. From top to bottom: (1) $\sin^2(\pi t)$ schedule, (2) σ schedule, (3) w^{KL} schedule. ImageNet class is *lighthouse*.

D.3. Performance Evaluation on SiT. We use 1000 class-level prompts and generate two images for each class, yielding 2000 images in total. We conduct two ablation studies on SiT to examine the effect of the truncation horizon n_{truncate} and the reward scale. The base SiT model obtains HPSv2 0.183, ImageReward -0.658 , CLIPScore 0.239, and PickScore 0.190. Full-horizon AM already improves these metrics, with **2nd SDE-AM-Full** reaching HPSv2 0.279 and **2nd ODE-AM-Full** reaching HPSv2 0.290. However, both full AM variants require 83.8s per iteration, and their performance is weaker than the truncated variants.

Table 5 studies higher-order ODE-AM under different reward scales with $n_{\text{truncate}} = 10$. For **4th ODE-AM-10**, the best overall reward scale is 10^{13} , reaching ImageReward 0.574 and HPSv2 0.328. For **6th ODE-AM-10**, the best configuration is reward scale 10^{15} , which achieves HPSv2 0.329, ImageReward 0.535, CLIPScore 0.269, and PickScore 0.207. Compared with **DRaFT-1** and **ReFL-10**, which are faster per iteration but obtain HPSv2 around $0.296 \sim 0.297$, the higher-order AM variants provide stronger final reward alignment.

Method	Reward Scale	HPSv2 \uparrow	ImageReward \uparrow	CLIPScore \uparrow	PickScore \uparrow	Iter.Time (s) \downarrow
Base Model	–	0.183 \pm 0.038	-0.658 ± 0.915	0.239 \pm 0.049	0.190 \pm 0.011	–
DRaFT-1	–	0.296 \pm 0.041	0.266 \pm 0.909	0.260 \pm 0.041	0.200 \pm 0.011	26.7
ReFL-10	–	0.297 \pm 0.034	0.282 \pm 0.885	0.265 \pm 0.038	0.200 \pm 0.011	26.4
2nd SDE-AM-Full	10^5	0.279 \pm 0.038	-0.084 ± 0.989	0.244 \pm 0.052	0.197 \pm 0.012	83.8
2nd ODE-AM-Full	10^5	0.290 \pm 0.037	0.185 \pm 0.967	0.243 \pm 0.049	0.199 \pm 0.012	83.8
4th ODE-AM-10	10^{11}	0.309 \pm 0.038	0.361 \pm 0.902	0.266 \pm 0.041	0.203 \pm 0.011	47.3
4th ODE-AM-10	10^{13}	<u>0.328</u> \pm 0.038	0.574 \pm 0.834	<u>0.269</u> \pm 0.040	<u>0.206</u> \pm 0.011	47.3
4th ODE-AM-10	10^{15}	0.326 \pm 0.039	0.491 \pm 0.878	0.265 \pm 0.041	<u>0.206</u> \pm 0.011	47.3
6th ODE-AM-10	10^{11}	0.327 \pm 0.039	0.531 \pm 0.872	0.265 \pm 0.040	<u>0.206</u> \pm 0.011	47.7
6th ODE-AM-10	10^{15}	0.329 \pm 0.039	<u>0.535</u> \pm 0.862	0.269 \pm 0.039	0.207 \pm 0.011	47.7
6th ODE-AM-10	10^{19}	<u>0.328</u> \pm 0.039	0.497 \pm 0.871	0.261 \pm 0.041	0.205 \pm 0.011	47.7

TABLE 5. Higher order ODE ablations: baselines (top) vs. **4th ODE-AM-10** and **6th ODE-AM-10** on **SiT-XL/2**.

Table 6 shows the effect of varying n_{truncate} . For both SDE and ODE samplers, increasing the truncation horizon from very small values improves the reward metrics, but the gain saturates after roughly 10–15 terminal steps. For **2nd AM-SDE**, the best performance appears around $n_{\text{truncate}} = 12$, achieving ImageReward 0.522, CLIPScore 0.273, and PickScore 0.206. For **2nd AM-ODE**, the best performance appears around $n_{\text{truncate}} = 15$, achieving HPSv2 0.324, ImageReward 0.522, and PickScore 0.206. At the same time, truncated AM reduces the per-iteration time from 83.8s to roughly $39 \sim 57$ s, depending on the number of active terminal steps. Recall that we use 50 denoising steps for all SiT experiments. This confirms that most useful reward-alignment signal is concentrated near the terminal denoising stage.

Overall, the ablation studies further confirm two observations. First, full-horizon AM is not the best use of computation; truncating to the final part of the trajectory is both faster and more effective. Second, higher-order ODE-AM can further improve the reward metrics when combined with a properly chosen reward scale. These results justify our use of truncated and (or) higher-order adjoint matching in the main SiT experiments.

Method	Trunc. Steps	HPSv2 \uparrow	ImageReward \uparrow	CLIPScore \uparrow	PickScore \uparrow	Iter.Time (s) \downarrow
Base Model	–	0.183 \pm 0.038	–0.658 \pm 0.915	0.239 \pm 0.049	0.190 \pm 0.011	–
2nd SDE-AM	1	0.301 \pm 0.040	0.331 \pm 0.917	0.261 \pm 0.041	0.201 \pm 0.011	38.9
	2	0.282 \pm 0.042	0.227 \pm 0.916	0.265 \pm 0.041	0.202 \pm 0.011	40.2
	3	0.294 \pm 0.041	0.305 \pm 0.911	0.267 \pm 0.040	0.204 \pm 0.011	41.3
	5	0.304 \pm 0.040	0.410 \pm 0.891	0.268 \pm 0.039	<u>0.205</u> \pm 0.011	43.2
	10	<u>0.319</u> \pm 0.038	0.491 \pm 0.876	<u>0.271</u> \pm 0.038	0.206 \pm 0.011	47.6
	12	0.318 \pm 0.038	0.522 \pm 0.867	0.273 \pm 0.038	0.206 \pm 0.011	49.6
	15	0.319 \pm 0.038	0.489 \pm 0.864	0.269 \pm 0.039	<u>0.205</u> \pm 0.011	52.3
20	0.317 \pm 0.038	<u>0.496</u> \pm 0.874	0.270 \pm 0.041	<u>0.205</u> \pm 0.011	57.1	
2nd SDE-AM-Full	50	0.279 \pm 0.038	–0.084 \pm 0.989	0.244 \pm 0.052	0.197 \pm 0.012	83.8
2nd ODE-AM	1	0.266 \pm 0.043	0.116 \pm 0.913	0.265 \pm 0.041	0.200 \pm 0.011	<u>39.2</u>
	2	0.305 \pm 0.041	0.344 \pm 0.895	0.264 \pm 0.039	0.202 \pm 0.011	40.1
	3	0.307 \pm 0.040	0.367 \pm 0.884	0.265 \pm 0.040	0.203 \pm 0.011	40.8
	5	0.312 \pm 0.040	0.404 \pm 0.897	0.263 \pm 0.040	0.203 \pm 0.011	42.7
	10	0.316 \pm 0.040	0.450 \pm 0.903	0.266 \pm 0.040	0.204 \pm 0.011	47.3
	12	0.321 \pm 0.039	0.477 \pm 0.860	<u>0.266</u> \pm 0.039	<u>0.205</u> \pm 0.011	49.3
	15	0.324 \pm 0.039	0.522 \pm 0.877	0.267 \pm 0.040	0.206 \pm 0.011	52.0
20	<u>0.324</u> \pm 0.039	<u>0.478</u> \pm 0.870	0.265 \pm 0.041	<u>0.205</u> \pm 0.011	56.7	
2nd ODE-AM-Full	50	0.290 \pm 0.037	0.185 \pm 0.967	0.243 \pm 0.049	0.199 \pm 0.012	83.8

TABLE 6. Effect of truncation steps in **2nd SDE-AM**, **2nd ODE-AM** on **SiT-XL/2**.

D.4. Detailed Performance Evaluation on FLUX. Taking advantage of this distilled flow matching backbone, we can use fewer sampling steps; in our case, we use 20 steps for both forward sampling and adjoint ODE integration. Moreover, the conditional guidance is also distilled into this model. In contrast to the usual classifier-free guidance [15], where both conditional and unconditional velocity predictions must be evaluated, **FLUX.2-Klein-4B** only requires a single model evaluation at each step, which effectively helps alleviate OOM issues.

For post-training, we use our truncated ODE adjoint matching algorithms, **2nd ODE-AM-1** and **2nd ODE-AM-3**, where only the last **one** and **three** adjoint ODE steps are integrated and included in the loss, respectively. We find that keeping more terminal adjoint matching steps does not improve training, and even made optimization slower and harder to converge, especially for full-step adjoint matching. As baselines, we consider two reward-backpropagation methods that can be applied to flow models: **DRaFT-1**[6] and **ReFL-5**[47], where the reward signal is back-propagated through the last 1 step and through one randomly selected step among the last 5 steps, respectively.

For fairness, the main methods shown in Table 3 are all trained for 500 iterations. In addition to the HPSv2 reward curve during training, we also evaluate other types of feedback, not only image-quality-related metrics such as Aesthetic Score [33], ImageReward [47], and PickScore [16], but also within-prompt diversity metrics such as LPIPS [52] and MS-SSIM [44], as well as prompt-wise mode collapse metrics, namely Recall and Recovery. Note that we generate 6 images for each prompt in the test set. For Recall and Recovery, we first extract image

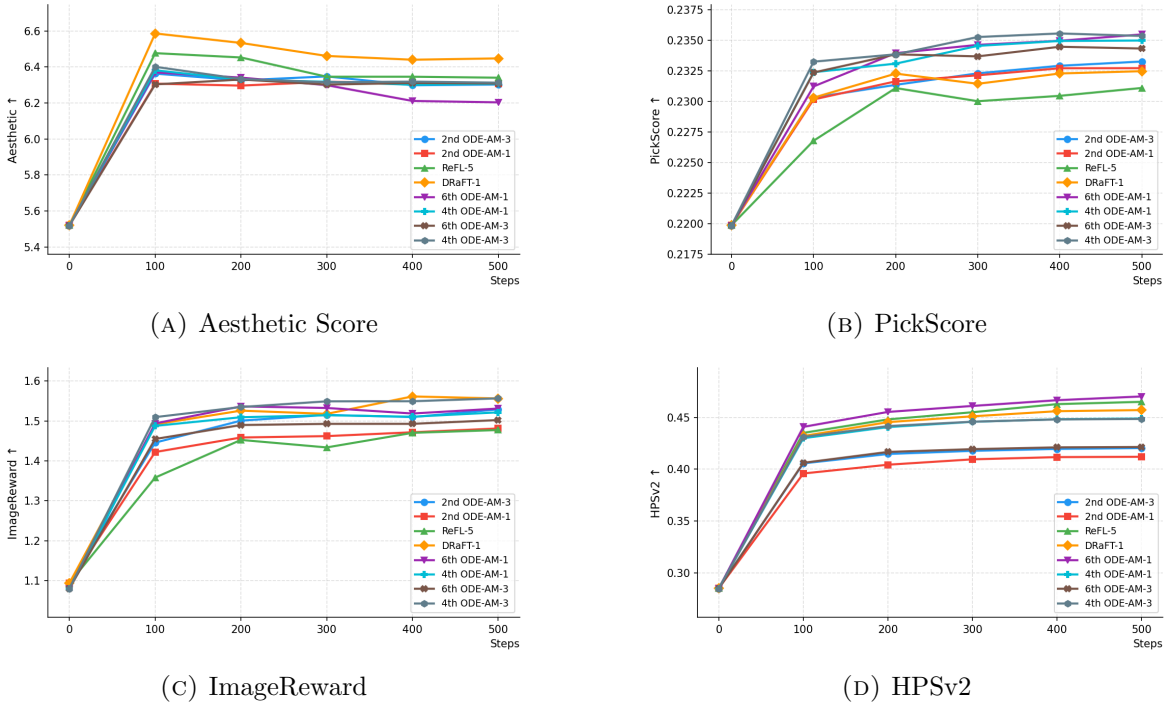


FIGURE 12. Image fidelity metrics evaluations on **FLUX.2-Klein-4B**.

features using **CLIP ViT-L/14**[30], and set $k = 5$ in the kNN-based evaluation; that is, we use the distance between each image and its 5th nearest neighbor as the radius.

We use only HPSv2 as the reward model during post-training, but other metrics, including Aesthetic Score, PickScore, and ImageReward, also improve substantially over the training; see Figure 13 for the evolution of image fidelity metrics through training using HPSv2. Some methods, **6th ODE-AM-1**, **DRaFT-1** and **ReFL-5**, appear to have slight advantages on training curve.

However, when it comes to image diversity, methods like **2nd ODE-AM-3**, **2nd ODE-AM-1** and **6th ODE-AM-3**, show strong potential for keep diversity; see Figure 13(A)(B) for pairwise within-prompt diversity. For simplicity, we report $1 - \text{MS-SSIM}$ instead of MS-SSIM itself. Overall, our methods are better at preserving diversity, which is consistent with the observation in prior work[25] that methods based on direct differentiation tend not to align with the target distribution, but instead favor only some modes of it. Moreover, to examine whether fine-tuning leads to mode collapse, we also compute Recall and Coverage to measure how much the fine-tuned distribution deviates from the original base distribution. As shown in Figure 13(C)(D), our methods like **2nd ODE-AM-1** and **6th ODE-AM-3** preserve the distribution significantly better than others. One possible reason is that those high-reward training methods like **DRaFT** and **ReFL** tend to produce overly bright or visually flashy images to obtain higher reward scores, often at the cost of semantic alignment. See Figure 5 for illustration. A detailed quantitative comparison between our method and the baselines is provided in Table 3 and Table 4.

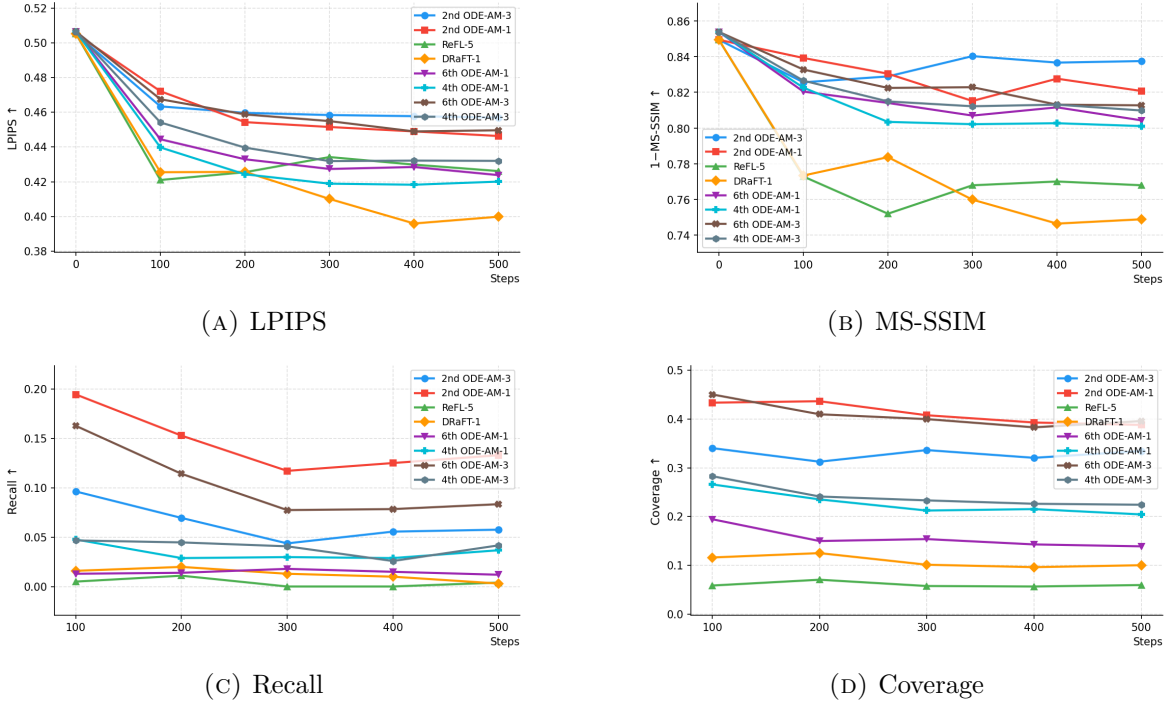


FIGURE 13. Within-prompt diversity metrics and prompt-wise mode preservation metrics on **FLUX.2-Klein-4B**.

D.5. How our algorithm accelerates post-training pipeline. In standard AM for diffusion models finetuning[7], each gradient update requires two sequential phases that both scale linearly with the number of ODE steps N . In the adjoint ODE integration phase, the adjoint signal is propagated backward through all N steps sequentially; each step requires one forward pass through the frozen base model to compute the vector-Jacobian product (VJP) with respect to the intermediate latent state. In the control loss phase, the trainable model is evaluated at each of the N trajectory states to compute the matching loss, with each evaluation requiring one forward pass through both the base and trainable models, followed by a backward pass to accumulate parameter gradients. Due to the sequential dependency between steps and memory limitations, neither phase can be trivially parallelized. For a large-scale model such as **FLUX.2-Klein-4B**, this results in prohibitively slow per-update wall time.

Our truncated AM algorithm restricts both phases to only the final $n_{\text{truncate}} \ll N$ steps of the trajectory: the adjoint ODE is integrated backward for n_{truncate} steps from the terminal state, and the control loss is evaluated only at the corresponding n_{truncate} trajectory states. Since both phases dominate total training time and each scales linearly with the number of active steps, the per-update cost reduces proportionally. In practice, a full 20-step update requires approximately 345 seconds, whereas the 1-step truncated variant reduces this to 32 seconds.

D.6. Hyper-parameters and details in training. For the experimental setup, training is performed on 7 Nvidia L40S GPUs. Note that although **SiT-XL/2** and **FLUX.2-Klein-4B** are training based on flow matching objective, their convention is a bit different. While **SiT-XL/2** generates images from $t = 0$ to $t = 1$, **FLUX.2-Klein-4B** does that from $t = 1$ to $t = 0$. In this paper, all analysis are based on $t = 0$ to $t = 1$ fashion.

Parameter setting for **FLUX.2-Klein-4B** is shown in Table 7. We post-trained it with $7 \times$ NVIDIA L40S GPUs with 48 GB VRAM each.

Category	FLUX.2-Klein-4B	SiT-XL/2-256
<i>Optimization</i>		
Optimizer	Adam	Adam
β_1	0.9	0.9
β_2	0.99	0.99
Learning Rate	1×10^{-5}	6×10^{-5}
LR Schedule	Cosine decay	Constant
Weight Decay	0.01	0.0
Gradient Clipping	1.0	10.0
<i>Training</i>		
Image Size	256	256
Number of Steps	20	50
Batch Size	16	8
Warmup Steps	10	25
Reward Scale	2×10^5	1×10^5
<i>Precision</i>		
Model Parameters	FP32	FP32
Adjoint Trajectory	FP32	FP32
Forward Pass	BF16	FP32

TABLE 7. Hyper-parameters for finetuning **FLUX.2-Klein-4B** and **SiT-XL/2-256** using Truncated Adjoint Matching.

D.7. **Training curve.** The **SiT-XL/2** training curves show that truncated and higher-order AM variants consistently outperform baselines; in particular **ODE-AM-6** gives the strongest trajectory. For **FLUX.2-Klein-4B**, all **Truncated AM** methods remain competitive with the two baselines **DRaFT-1** and **ReFL-5**.

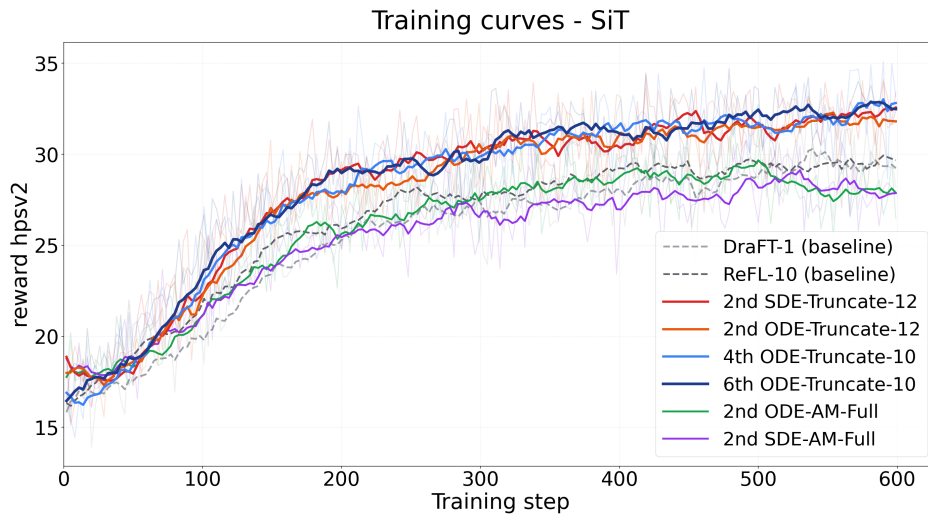


FIGURE 14. HPSv2 reward mean during training with different adaptive modes on **SiT-XL/2**.

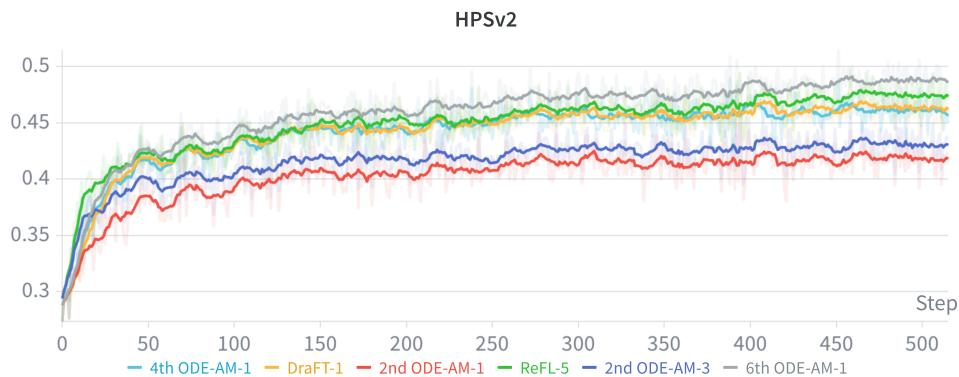


FIGURE 15. HPSv2 reward mean during training on **FLUX.2-Klein-4B**.

APPENDIX E. ADDITIONAL IMAGES

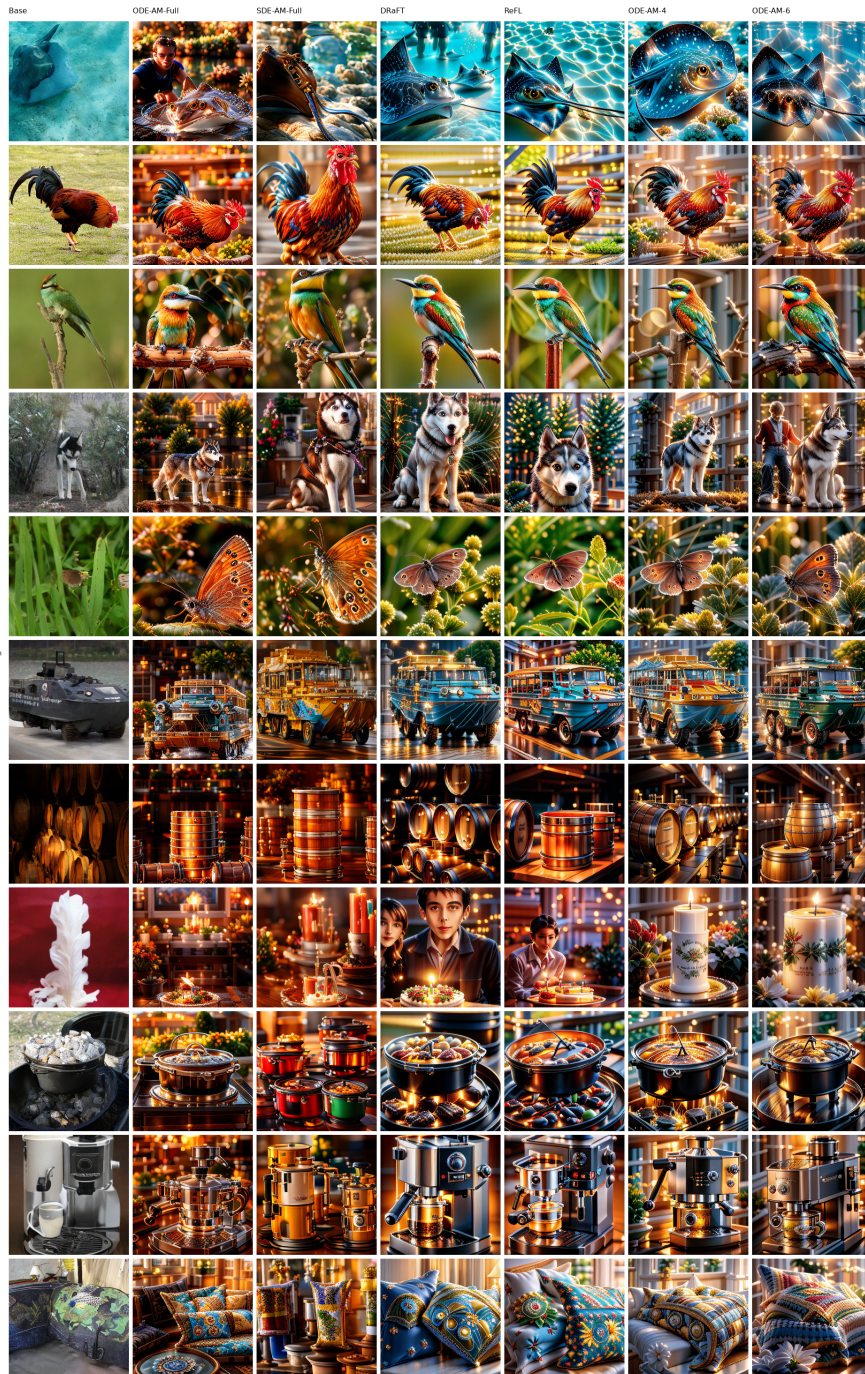


FIGURE 16. Methods from left to right: (1) Base, (2) 2nd ODE-AM-Full, (3) 2nd SDE-AM-Full, (4) DRaFT-1, (5) ReFL-10, (6) 2nd SDE-AM-12, (7) 4th ODE-AM-10, (8) 6th ODE-AM-10 on **SiT-XL/2**.



Prompt = "Photo of a baby chick dressed as an inmate."



Prompt = "Portrait of a beautiful cyberpunk Asian female."



Prompt = "A siamese cat curled up in a sink"



Prompt = "Sci-fi skill game icon in flat vector style, displayed in black and white."



Prompt = "A wizard casting a spell."



Prompt = "A Chinese dragon flying above a busy San Francisco street at night."



Prompt = "A volcano made of ivory vines and crimson rocks erupts with a smoke resembling a demonic eye."



Prompt = "Riding neon bicycles in the woods."

FIGURE 17. Methods from left to right: (1) Base, (2) 2nd ODE-AM-Full, (3) 6th ODE-AM-3, (4) 4th ODE-AM-3, (5) 2nd ODE-AM-3, (6) DRaFT-1, (7) ReFL-5 on FLUX.2-Klein-4B.



Prompt = "1979 magazine cover featuring a traditional Japanese village in the style of Moebius."



Prompt = "3D cat in shape of mango fruit."



Prompt = "5 roman soldiers burning down a small house in a snow filled valley."



Prompt = "A bathroom sink with a tooth shaped toothbrush holder."



Prompt = "A kitchen range with food cooking in the oven and on the stove top."



Prompt = "A motorcycle racer is going into a turn."



Prompt = "A park with marble statues."



Prompt = "A silver park bench sits in concrete with grass behind."

FIGURE 18. Methods from left to right: (1) Base, (2) 2nd ODE-AM-Full, (3) 6th ODE-AM-3, (4) 4th ODE-AM-3, (5) 2nd ODE-AM-3, (6) DRaFT-1, (7) ReFL-5 on FLUX.2-Klein-4B.

DEPARTMENT OF INDUSTRIAL ENGINEERING AND OPERATIONS RESEARCH, COLUMBIA UNIVERSITY.

Email address: zg2525@columbia.edu

DEPARTMENT OF INDUSTRIAL ENGINEERING AND OPERATIONS RESEARCH, COLUMBIA UNIVERSITY.

Email address: js6646@columbia.edu

DEPARTMENT OF INDUSTRIAL ENGINEERING AND OPERATIONS RESEARCH, COLUMBIA UNIVERSITY.

Email address: ddy1@columbia.edu

DEPARTMENT OF INDUSTRIAL ENGINEERING AND OPERATIONS RESEARCH, COLUMBIA UNIVERSITY.

Email address: wt2319@columbia.edu