FASTER: FAST ADJOINT STOCHASTIC TRANSPORT FOR ENDPOINT REFINEMENT IN REWARD-GUIDED IMAGE EDITING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reward-guided image editing at test time seeks to improve a specified reward while preserving source content and visual plausibility. Many existing approaches optimize candidates through pretrained generation processes, making repeated adjustment depend on costly large-model execution and, in some cases, backbone backpropagation. We develop a theoretical framework that jointly accounts for reward, source preservation, and pretrained-prior preferences, allowing the desired output distribution to be specified separately from the dynamics used to realize it. Based on this framework, we introduce FASTER, which trains a small network for each source and objective to perform inexpensive editing, while pretrained and reward models provide feedback on candidate outputs. By reusing each candidate and its feedback across multiple small-network updates, FASTER reduces repeated sampling and supervision queries without placing the pretrained generative backbone inside the inner optimization loop. On SD3, FASTER leads all four target metrics and several validation metrics among the evaluated methods. Compared with the evaluated baseline that optimizes controls along pretrained generation trajectories, FASTER achieves editing-time speedups of up to $6.91\times$ on Stable Diffusion 3 and $24.14\times$ on Stable Diffusion 1.5.

1 INTRODUCTION

Pretrained diffusion and flow models provide rich visual priors for image generation (Ho et al., 2020; Song et al., 2021; Rombach et al., 2022; Lipman et al., 2023; Liu et al., 2023; Esser et al., 2024). However, their conditioning interfaces do not fully express every downstream objective: human preferences, reference styles, and task-specific semantic requirements may instead be specified through rewards available at test time. Reward-guided inference adapts generated outputs to such objectives without retraining the pretrained model (Bansal et al., 2023; Yu et al., 2023; Ye et al., 2024; Tang et al., 2025; Eyring et al., 2024; Susladkar et al., 2026). We focus on *reward-guided image editing at test time*: given a source image and a reward objective, improve the objective while retaining content that need not change and maintaining visual plausibility (Chang et al., 2026). Unlike unconstrained reward maximization, this task requires meaningful, source-preserving changes; a higher reward alone does not establish a successful edit.

Existing approaches typically inject reward guidance into pretrained generation or optimize candidates through a generation mapping or trajectory. Guidance methods use external gradients to adjust intermediate states, with refinements to the guidance geometry and update rules (Chung et al., 2023; Bansal et al., 2023; Yu et al., 2023; He et al., 2024; Yang et al., 2024; Ye et al., 2024; Susladkar et al., 2026). Optimization-based approaches instead adjust input noise, editable variables, or trajectory controls to improve the generated output (Wallace et al., 2023; Ben-Hamu et al., 2024; Tang et al., 2025; Eyring et al., 2024; Chang et al., 2026). Other advances improve efficiency through reward optimization of one-step models, flow-based editing, or efficient reward guidance (Eyring et al., 2024; Patel et al., 2025; Huang et al., 2026). Despite their different computational costs, many of these approaches couple the pretrained model's role as a source of visual knowledge with its role in executing candidate updates. Repeated adjustment invokes the generator again, and some approaches additionally backpropagate through its backbone; candidate evaluation introduces further

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

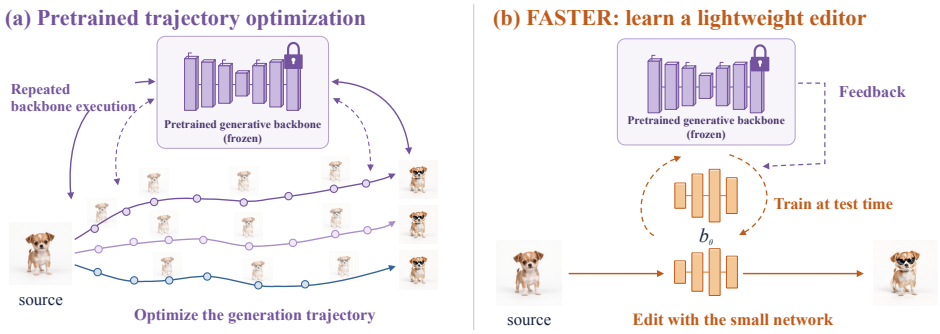

Figure 1: **Decoupling pretrained knowledge from editing dynamics.** (a) Trajectory optimization repeatedly executes a frozen generative backbone. (b) FASTER learns a small per-image editor at test time, reusing frozen-model feedback across updates; the learned network then performs the edit.

decoding and reward-computation costs. Can pretrained knowledge continue to constrain desirable edits while a separate, lightweight process performs the actual adjustment?

Figure 1 contrasts optimization along pretrained generation trajectories with FASTER's lightweight editing process: the former repeatedly executes the large generative backbone, whereas the latter uses a small network to perform editing with feedback from frozen models. We establish a theoretical basis for this separation by jointly analyzing reward improvement, source preservation, and pretrained-prior plausibility, specifying a target output distribution and constructing an independent editing process that realizes it under ideal conditions. Each task fixes a source and an objective, so the learned editor need only adapt the current instance rather than acquire general-purpose generation capabilities.

Based on this framework, we introduce **FASTER**: a small network trained at test time performs inexpensive editing updates, while pretrained and reward models provide feedback at candidate outputs. Reusing each candidate and its feedback across multiple network updates reduces repeated sampling and costly supervision queries. To support explicit semantic changes that increasing an image–text score may fail to produce, we further introduce an editing term that obtains semantic directions from the pretrained model's conditional predictions, without backpropagation through its backbone. This term enters the same feedback interface, allowing the lightweight editor to use pretrained semantic knowledge while retaining its inexpensive update process. On Stable Diffusion 3 (Esser et al., 2024), FASTER leads the target metric in all four tasks and several validation metrics among the evaluated methods. Relative to the evaluated baseline that optimizes controls along pretrained generation trajectories, it achieves editing-time speedups of up to $6.91\times$ on Stable Diffusion 3 (Esser et al., 2024) and $24.14\times$ on Stable Diffusion 1.5 (Rombach et al., 2022).

Our contributions are threefold:

- **Decoupling the editing target from its dynamics.** We combine reward, source preservation, and pretrained-prior preferences into a target distribution, and construct separate lightweight dynamics that realize it under proper conditions theoretically.
- **FASTER: inexpensive editing with a small network.** A per-image small network executes inexpensive edits and reuses candidate feedback to reduce repeated computation. A semantic editing term supplies directions from conditional model predictions through the same interface, without generative-backbone backpropagation.
- **Editing quality and efficiency across tasks.** FASTER leads all four target metrics and several validation metrics in the SD3 comparison, with up to $\mathbf{6.91}\times$ and $\mathbf{24.14}\times$ speedups on SD3 and SD1.5, respectively. A 768×768 SD3 ImageReward edit takes approximately 7.7 seconds on a single consumer GPU.

2 Related Work

Inference-time guidance and alignment. Training-free guidance steers pretrained diffusion and flow models toward external objectives without updating model weights. DPS, Universal Guidance,

and FreeDoM construct guidance from denoised predictions or external energies (Chung et al., 2023; Bansal et al., 2023; Yu et al., 2023), while MPGD, DSG, and TFG further improve the geometry and design of guidance (He et al., 2024; Yang et al., 2024; Ye et al., 2024). Alternatively, DOODL and D-Flow optimize latent variables through the pretrained generative mapping (Wallace et al., 2023; Ben-Hamu et al., 2024), with DNO and ReNO improving robustness or efficiency of direct optimization (Tang et al., 2025; Eyring et al., 2024). Distributional methods such as DAS and Feynman–Kac steering instead reshape inference through reweighting and resampling (Kim et al., 2025; Singhal et al., 2025). These approaches provide flexible test-time alignment, but generally remain coupled to pretrained generation during optimization.

Training-free and reward-guided image editing. Training-free image editing must also preserve source correspondence. Existing methods use inversion and attention control (Hertz et al., 2023b; Mokady et al., 2023), controlled corruption (Meng et al., 2022), and feature reuse (Tumanyan et al., 2023; Cao et al., 2023). Other approaches rely on score differences (Hertz et al., 2023a) or directly model source-to-target editing dynamics (Xu et al., 2024; Kulikov et al., 2025). For reward-guided editing, ITOC optimizes controls along the pretrained trajectory (Chang et al., 2026). FlowChef exploits flow geometry (Patel et al., 2025), whereas RewardFlow combines multiple differentiable rewards (Susladkar et al., 2026). FMRG further improves guidance efficiency through flow-map lookahead (Huang et al., 2026). FASTER instead specifies a source-conditioned endpoint target and learns a separate lightweight transport to realize it.

Generative priors and stochastic-control optimization. Frozen models provide priors for inverse imaging (Feng et al., 2023; Feng & Bouman, 2024; Mardani et al., 2024) and score-distillation-based 3D optimization (Poole et al., 2023; Wang et al., 2023). UniDB formulates diffusion bridges as SOC with a tunable terminal penalty (Zhu et al., 2025), while UniDB++ accelerates sampling within this framework through a training-free solver (Pan et al., 2026). For SOC solvers, Adjoint Matching learns controls by regression (Domingo-Enrich et al., 2025), Adjoint Sampling reuses terminal information through reference bridges (Havens et al., 2025), and Efficient Adjoint Matching simplifies the reference dynamics and terminal cost (Shin et al., 2026). FASTER uses frozen-model endpoint supervision to learn a lightweight per-image editor.

3 PRELIMINARIES

We edit in latent space: $\boldsymbol{s}$ and $\boldsymbol{y}$ denote source and output latents, $\mathcal{D}$ is a frozen decoder, and $r(\boldsymbol{y}) = R(\mathcal{D}(\boldsymbol{y}))$ is a task reward. We write p_{pre} for the pretrained latent prior. The control time $\lambda \in [0, 1]$ below is distinct from the pretrained model's noise time t used for semantic queries in Sec. 4.

3.1 REFERENCE DYNAMICS AND STOCHASTIC CONTROL

A reference process describes state evolution before applying a task-dependent control. From a fixed initial state $\boldsymbol{x}_{\mathrm{init}}$, its path law Q and a controlled path law $P^{\boldsymbol{b}}$ are defined by

$$Q: \qquad d\boldsymbol{x}_\lambda = \boldsymbol{a}_0(\boldsymbol{x}_\lambda, \lambda)\, d\lambda + \sigma_\lambda\, d\boldsymbol{w}_\lambda, \tag{1a}$$

$$P^{\boldsymbol{b}}: \qquad d\boldsymbol{x}_\lambda = [\boldsymbol{a}_0(\boldsymbol{x}_\lambda, \lambda) + \boldsymbol{b}(\boldsymbol{x}_\lambda, \lambda)]\, d\lambda + \sigma_\lambda\, d\boldsymbol{w}_\lambda, \qquad \boldsymbol{x}_0 = \boldsymbol{x}_{\mathrm{init}}. \tag{1b}$$

Here $\boldsymbol{a}_0$ is the reference drift, $\boldsymbol{b}$ the control drift, and $\sigma_\lambda > 0$ the shared diffusion scale. In pretrained-trajectory control, $\boldsymbol{a}_0$ contains the frozen generator's prediction, so each state update still invokes the large model even when the control branch is small. ITOC illustrates this computational dependence (Chang et al., 2026); its fixed-noise trajectory optimization is distinct from the distributional formulation below.

For a terminal cost g, stochastic optimal control minimizes

$$J(\boldsymbol{b}) = \mathbb{E}_{P^{\boldsymbol{b}}}\left[\int_0^1 \frac{\|\boldsymbol{b}(\boldsymbol{x}_\lambda, \lambda)\|^2}{2\sigma_\lambda^2}\, d\lambda + g(\boldsymbol{x}_1)\right] = D_{\mathrm{KL}}(P^{\boldsymbol{b}} \| Q) + \mathbb{E}_{P^{\boldsymbol{b}}}[g(\boldsymbol{x}_1)]. \tag{2}$$

The equality requires the usual Girsanov and integrability conditions (Domingo-Enrich et al., 2025; Havens et al., 2025). The reference process regularizes paths, while g specifies the preference over endpoints. These are separate design choices: the pretrained prior p_{pre} can inform the endpoint preference without dictating the reference dynamics Q.

3.2 Adjoint Matching

Adjoint Matching (AM) trains a control by regressing it toward an adjoint-derived target (Domingo-Enrich et al., 2025). For a sampled trajectory held fixed during regression, its lean adjoint in the terminal-cost setting obeys

$$\frac{d\widetilde{\boldsymbol{a}}_\lambda}{d\lambda} = -\big[\nabla_{\boldsymbol{x}} \boldsymbol{a}_0(\boldsymbol{x}_\lambda, \lambda)\big]^\top \widetilde{\boldsymbol{a}}_\lambda, \qquad \widetilde{\boldsymbol{a}}_1 = \nabla g(\boldsymbol{x}_1), \tag{3}$$

and supplies the physical-drift regression target $-\sigma_\lambda^2 \widetilde{\boldsymbol{a}}_\lambda$. AM is a solver for the control problem, not another definition of its objective. Although its lean adjoint avoids differentiating the learned control, Eq. (3) retains derivatives of the reference drift. A pretrained reference can therefore incur large-model work both when sampling paths and when propagating supervision. This motivates the choice of reference dynamics in Sec. 4.

4 FASTER

Pretrained-trajectory control repeatedly executes the large generator as the edit changes (Sec. 3). FASTER instead specifies a per-image endpoint target and learns a small controller to reach it, reusing endpoint feedback across controller updates (Fig. 2).

4.1 From a Per-Image Editing Objective to Lightweight Control

For a fixed source latent $\boldsymbol{s}$, a successful edit should improve the task reward $r(\boldsymbol{y})$, avoid unnecessary changes to the source, and favor plausible outputs under the pretrained prior p_{pre}. Let $D(\boldsymbol{s}, \boldsymbol{y})$ measure source change and ρ be an output density for this editing instance. We minimize the following functional over output densities:

$$F_{\boldsymbol{s}}(\rho) = D_{\mathrm{KL}}(\rho \| p_{\mathrm{pre}}) + \alpha \mathbb{E}_\rho[D(\boldsymbol{s}, Y)] - \beta \mathbb{E}_\rho[r(Y)], \qquad \alpha \geq 0, \quad \beta > 0. \tag{4}$$

The KL term favors plausible outputs, while the preservation cost anchors them to $\boldsymbol{s}$. The minimizer is the target below with $\gamma = 1$; an independent weight $\gamma \geq 0$ controls the prior contribution (Appendix A):

$$\pi_{\boldsymbol{s}}(\boldsymbol{y}) = Z_{\boldsymbol{s}}^{-1} p_{\mathrm{pre}}(\boldsymbol{y})^\gamma \exp[-\alpha D(\boldsymbol{s}, \boldsymbol{y}) + \beta r(\boldsymbol{y})]. \tag{5}$$

We assume $0 < Z_{\boldsymbol{s}} < \infty$ and use $D(\boldsymbol{s}, \boldsymbol{y}) = \|\boldsymbol{y} - \boldsymbol{s}\|^2/2$. This target specifies preferred outputs without prescribing their generation paths.

For a fixed source and request, the source supplies content and frozen models supply feedback. A small per-image controller therefore need only learn the current edit, rather than the full image distribution.

Let $\boldsymbol{z}_\lambda$ be the residual from $\boldsymbol{s}$ and $\tau > 0$ a noise scale. We choose a zero-drift Brownian reference Q_τ. Its endpoint density and endpoint-conditioned bridge are analytic:

$$d\boldsymbol{z}_\lambda = \tau \, d\boldsymbol{w}_\lambda, \quad \boldsymbol{z}_0 = 0, \quad q_\tau(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; 0, \tau^2 \boldsymbol{I}), \tag{6a}$$

$$\boldsymbol{z}_\lambda \mid \boldsymbol{z}_1 \sim \mathcal{N}(\lambda \boldsymbol{z}_1, \tau^2 \lambda(1-\lambda)\boldsymbol{I}), \quad 0 < \lambda < 1. \tag{6b}$$

Adding the per-image controller gives the editing dynamics

$$d\boldsymbol{z}_\lambda = \boldsymbol{b}_\theta(\boldsymbol{z}_\lambda, \lambda; \boldsymbol{s})\, d\lambda + \tau \, d\boldsymbol{w}_\lambda, \qquad \boldsymbol{y}_\lambda = \boldsymbol{s} + \boldsymbol{z}_\lambda. \tag{7}$$

We implement the controller with a small network fitted separately for each source and objective:

$$\boldsymbol{b}_\theta(\boldsymbol{z}, \lambda; \boldsymbol{s}) = \boldsymbol{m} + f_\psi(\boldsymbol{z}, \lambda, \boldsymbol{s}), \qquad \theta = (\boldsymbol{m}, \psi). \tag{8}$$

The latent-shaped offset $\boldsymbol{m}$ is independent of state and time. The three-layer convolutional network f_ψ takes the residual, source latent, and broadcast time as input and predicts a correction. Each editing step evaluates only this small network; only θ is trained, with frozen pretrained models providing endpoint feedback. Architecture and initialization details are in Appendix E.

The reference endpoint q_τ is Gaussian, however, so it need not equal the desired residual density $\bar{\pi}_{\boldsymbol{s}}(\boldsymbol{z}) = \pi_{\boldsymbol{s}}(\boldsymbol{s} + \boldsymbol{z})$. We compensate with the terminal cost

$$g_{\boldsymbol{s}}(\boldsymbol{z}) = \log q_\tau(\boldsymbol{z}) - \log \bar{\pi}_{\boldsymbol{s}}(\boldsymbol{z}). \tag{9}$$

Since $q_\tau e^{-g_{\boldsymbol{s}}} = \bar{\pi}_{\boldsymbol{s}}$, the unrestricted optimum of Eq. (2) has $\boldsymbol{y}_1 \sim \pi_{\boldsymbol{s}}$ under the conditions in Appendix B. The terminal cost cancels the reference endpoint preference and imposes the editing target; a finite controller with estimated feedback approximates this ideal law.

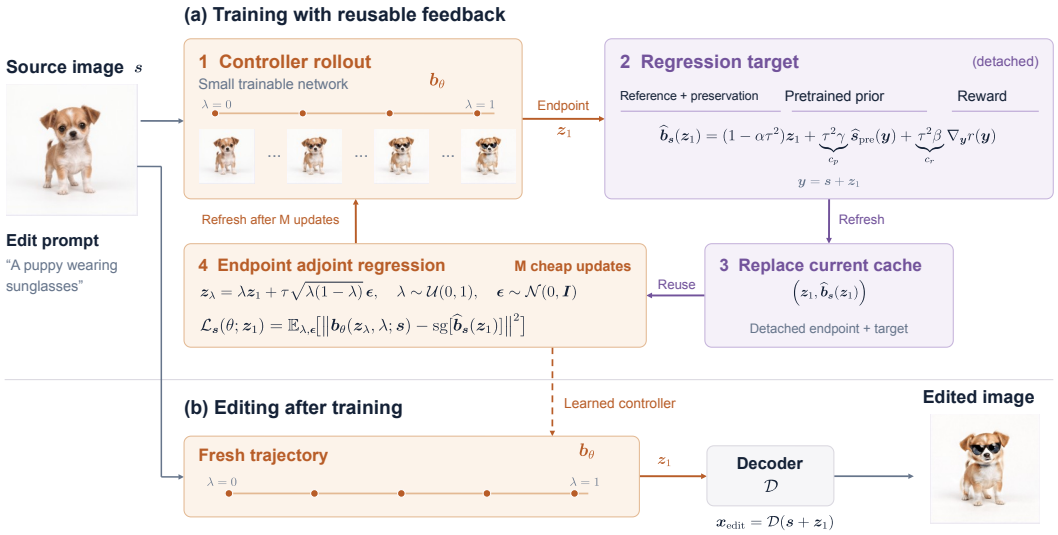

Figure 2: **FASTER training and editing pipeline.** (a) Each of K refreshes caches an endpoint and its detached preservation, prior, and reward target for M controller updates. (b) After training, a fresh controller rollout is decoded into the edited image.

4.2 Learning from Reusable Endpoint Supervision

Endpoint feedback requires decoding, reward gradients, and pretrained-model queries. We reuse each evaluated endpoint for multiple small-network updates, reducing repeated rollouts and feedback computation.

The optimal path law $P_{\boldsymbol{s}}^\star$ for Eq. (9) has density ratio $dP_{\boldsymbol{s}}^\star/dQ_\tau = \bar{\pi}_{\boldsymbol{s}}(\boldsymbol{z}_1)/q_\tau(\boldsymbol{z}_1)$, which depends only on the endpoint. Thus its conditional paths remain the reference bridges in Eq. (6) (Appendix B). This structure is useful before convergence as well. For a current path law P with endpoint density p_1, the path-space objective decomposes as

$$J_{\boldsymbol{s}}(P) = D_{\text{KL}}(p_1 \| \bar{\pi}_{\boldsymbol{s}}) + \mathbb{E}_{\boldsymbol{z}_1 \sim p_1}[D_{\text{KL}}(P(\cdot \mid \boldsymbol{z}_1) \| Q_\tau(\cdot \mid \boldsymbol{z}_1))]\,. \tag{10}$$

Replacing the conditional paths by reference bridges preserves the endpoint distribution and removes the second term. This motivates pairing current endpoints with resampled bridge states. We adopt Reciprocal Adjoint Matching from Adjoint Sampling (Havens et al., 2025), whose reciprocal-projection analysis supports this construction.

The Brownian reference in Eq. (6) has $\boldsymbol{a}_0 = 0$, hence $\nabla_{\boldsymbol{x}}\boldsymbol{a}_0 = 0$; the lean-adjoint equation (3) therefore gives $d\widetilde{\boldsymbol{a}}_\lambda/d\lambda = 0$ and $\widetilde{\boldsymbol{a}}_\lambda = \nabla g_{\boldsymbol{s}}(\boldsymbol{z}_1)$. Its physical-drift label is therefore $-\tau^2\nabla g_{\boldsymbol{s}}(\boldsymbol{z}_1)$, eliminating backward adjoint propagation. For quadratic preservation, the practical label at $\boldsymbol{y} = \boldsymbol{s} + \boldsymbol{z}_1$ is

$$\widehat{\boldsymbol{b}}_{\boldsymbol{s}}(\boldsymbol{z}_1) = (1 - \alpha\tau^2)\boldsymbol{z}_1 + \underbrace{\tau^2\gamma}_{c_p}\,\widehat{\boldsymbol{s}}_{\text{pre}}(\boldsymbol{y}) + \underbrace{\tau^2\beta}_{c_r}\,\nabla_{\boldsymbol{y}} r(\boldsymbol{y}). \tag{11}$$

The first term combines reference correction and preservation; the remaining terms provide estimated prior and reward gradients. Here $\widehat{\boldsymbol{s}}_{\text{pre}}$ is a prior-guidance estimate obtained from noisy model queries; Appendix D describes its surrogate-prior interpretation. Reward gradients pass through the frozen decoder and reward model at this endpoint. We evaluate the label once, then store it with the detached endpoint.

For the stored pair, draw $\lambda \sim \mathcal{U}(0,1)$ and independent $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$. Sample the reference bridge as $\boldsymbol{z}_\lambda = \lambda\boldsymbol{z}_1 + \tau\sqrt{\lambda(1-\lambda)}\,\boldsymbol{\epsilon}$ and minimize

$$\mathcal{L}_{\boldsymbol{s}}(\theta; \boldsymbol{z}_1) = \mathbb{E}_{\lambda,\boldsymbol{\epsilon}}\left[\left\| \boldsymbol{b}_\theta(\boldsymbol{z}_\lambda, \lambda; \boldsymbol{s}) - \text{sg}[\widehat{\boldsymbol{b}}_{\boldsymbol{s}}(\boldsymbol{z}_1)] \right\|^2\right]. \tag{12}$$

Algorithm 1 FASTER: per-image endpoint refinement

Require: Source latent $\boldsymbol{s}$; reward r; α, τ, c_p, c_r; refreshes K; updates M.
1: Initialize $\boldsymbol{m}$ and the final layer of f_ψ to zero in Eq. (8).
2: **for** $k = 1, \dots, K$ **do**
3: Roll out an endpoint $\boldsymbol{z}_1$ using Eq. (7).
4: Evaluate Eq. (11); replace the cache with detached $(\boldsymbol{z}_1, \widehat{\boldsymbol{b}}_{\boldsymbol{s}}(\boldsymbol{z}_1))$.
5: **for** $j = 1, \dots, M$ **do**
6: Draw a minibatch of independent $(\lambda, \boldsymbol{\epsilon})$ pairs.
7: Form detached $\boldsymbol{z}_\lambda = \lambda \boldsymbol{z}_1 + \tau\sqrt{\lambda(1-\lambda)}\boldsymbol{\epsilon}$.
8: Update $\theta = (\boldsymbol{m}, \psi)$ using the averaged loss in Eq. (12).
9: **end for**
10: **end for**
11: Roll out a fresh endpoint $\boldsymbol{z}_1$ with the learned $\boldsymbol{b}_\theta$; return $\mathcal{D}(\boldsymbol{s} + \boldsymbol{z}_1)$.

The label and sampled states are detached. Resampling bridge states supplies new training inputs without new rollouts or feedback queries. Appendix C establishes population consistency under exact target endpoints and labels, and discusses finite-update limitations.

Each of K refreshes replaces the cached endpoint–label pair and performs M controller updates. After fitting, a fresh controller sample is decoded as the output (Algorithm 1). Appendix H examines the effect of varying K and M.

4.3 Semantic Editing through a Model-Derived Reward

An image–text score can increase without fulfilling a specific editing request. We therefore use the pretrained model's conditional information to define a semantic reward for the same endpoint solver.

Prior work observes that text conditioning shapes semantic content and spatial structure during early denoising (Balaji et al., 2022; Hertz et al., 2023b). This motivates querying a higher-noise state $\boldsymbol{x}_t = (1-t)\boldsymbol{y} + t\boldsymbol{\epsilon}$ rather than only the clean candidate $\boldsymbol{y}$. We draw $t \sim \nu$ from a fixed high-noise interval inside $(0, 1)$ and independent $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$. This direct construction avoids a pretrained sampling rollout.

Inspired by classifier-free guidance (Ho & Salimans, 2022), we compare target and source descriptions c_{tgt} and c_{src} at the same noisy candidate. Let $p_t(\cdot \mid c)$ be the noisy conditional density. Define

$$E_{\mathrm{edit}}(\boldsymbol{y}) = \mathbb{E}_{t,\boldsymbol{\epsilon}}\left[\log \frac{p_t(\boldsymbol{x}_t \mid c_{\mathrm{tgt}})}{p_t(\boldsymbol{x}_t \mid c_{\mathrm{src}})}\right]. \tag{13}$$

Let $\boldsymbol{s}_t(\boldsymbol{x}, c) = \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x} \mid c)$ be the conditional score and $\boldsymbol{v}_c$ the conditional velocity. Under SD3's data-to-noise convention, their relation is

$$\boldsymbol{s}_t(\boldsymbol{x}, c) = -\frac{\boldsymbol{x} + (1-t)\boldsymbol{v}_c(\boldsymbol{x}, t)}{t}. \tag{14}$$

Taking the target-minus-source score difference cancels $\boldsymbol{x}_t$. The linear Jacobian $\partial \boldsymbol{x}_t / \partial \boldsymbol{y} = (1-t)\boldsymbol{I}$ then gives

$$\boldsymbol{g}_{\mathrm{edit}}(\boldsymbol{y}) := \nabla_{\boldsymbol{y}} E_{\mathrm{edit}}(\boldsymbol{y}) = \mathbb{E}_{t,\boldsymbol{\epsilon}}\left[\frac{(1-t)^2}{t}\big(\boldsymbol{v}_{c_{\mathrm{src}}} - \boldsymbol{v}_{c_{\mathrm{tgt}}}\big)(\boldsymbol{x}_t, t)\right]. \tag{15}$$

Thus paired conditional forward predictions provide the gradient without backpropagation through the pretrained backbone; Appendix G gives the derivation.

Adding the reward with weight $\eta \geq 0$ subtracts $\eta E_{\mathrm{edit}}(\boldsymbol{s} + \boldsymbol{z})$ from the terminal cost, giving the practical label

$$\widehat{\boldsymbol{b}}^{\mathrm{edit}}_{\boldsymbol{s}}(\boldsymbol{z}_1) = \widehat{\boldsymbol{b}}_{\boldsymbol{s}}(\boldsymbol{z}_1) + \underbrace{\tau^2 \eta}_{c_e}\, \widehat{\boldsymbol{g}}_{\mathrm{edit}}(\boldsymbol{s} + \boldsymbol{z}_1). \tag{16}$$

Here $\widehat{\boldsymbol{g}}_{\mathrm{edit}}$ estimates Eq. (15) using paired source/target model predictions. We cache and reuse the augmented label in the same bridge regression and refresh loop of Sec. 4.2.

Table 1: Results on Stable Diffusion 3. Each task includes a target metric, which is directly optimized, and validation metrics used to assess potential reward hacking. Arrows indicate whether higher (↑) or lower (↓) values are better. Bold values indicate the best result for each metric.

Metric	ITOC	GA	DPS	FreeDoM	TFG	MPGD	**FASTER**
Human preference							
TARGET **ImageReward**↑	1.8326	1.8645	1.4687	1.6535	1.5128	0.9413	**1.8958**
VALIDATION HPSv2↑	0.2619	0.2608	0.2579	0.2576	0.2580	0.2563	**0.2624**
VALIDATION CLIPScore↑	0.2734	0.2776	0.2566	0.2388	0.2539	0.2655	**0.2783**
VALIDATION Aesthetic↑	**5.7933**	5.6329	5.5978	5.6719	5.4951	5.4667	5.6991
Style transfer							
TARGET **GramDiff**↓	8.3341	9.6810	7.3644	6.9180	9.9923	10.2211	**4.9746**
VALIDATION CLIP-I_{sty}↑	0.5022	0.4951	0.5079	0.5068	0.5104	0.4945	**0.5168**
VALIDATION $DINO_{sty}$↑	0.1727	0.1233	0.2361	**0.2460**	0.2013	0.1478	0.1442
Counterfactual generation							
TARGET **Logit**$_{tgt}$↑	18.8789	11.4157	20.5937	21.6252	20.6566	16.0873	**29.8948**
VALIDATION CLIPScore↑	0.1884	0.1620	0.1882	0.1950	0.1882	0.1823	**0.2253**
Text-guided facial editing							
TARGET **CLIPScore**↑	0.2654	0.2121	0.2544	0.2495	0.2509	0.2093	**0.3538**
VALIDATION ImageReward↑	-0.8159	-0.8309	-0.7610	-0.7436	-0.7539	-0.9225	**-0.6027**
VALIDATION HPSv2↑	**0.2573**	0.2564	0.2531	0.2549	0.2535	0.2572	0.2533

5 EXPERIMENTS

Following Chang et al. (2026), we evaluate four reward-guided editing tasks with 300 source–condition pairs each. Human preference optimizes ImageReward (Xu et al., 2023). Style transfer minimizes the Gram-matrix difference between output and reference-image features from CLIP ViT-B/16 (Radford et al., 2021). Counterfactual generation maximizes the target-class logit of a robust ResNet-50 classifier (Engstrom et al., 2019), using a class label as the reward condition. Text-guided facial editing maximizes CLIP ViT-L/14 image–text similarity for the requested attribute. Each task also reports validation metrics not directly optimized: HPSv2 (Wu et al., 2023), CLIPScore, and Aesthetic for human preference; reference-image CLIP-I and DINO (Caron et al., 2021) similarity for style; target-class text CLIPScore for counterfactual generation; and ImageReward and HPSv2 for facial editing. These metrics assess whether target improvements generalize beyond the optimized reward.

We use Stable Diffusion 3 Medium (SD3) (Esser et al., 2024) for the main quality comparison. The baselines are ITOC (Chang et al., 2026), gradient ascent (GA), DPS (Chung et al., 2023), FreeDoM (Yu et al., 2023), TFG (Ye et al., 2024), and MPGD (He et al., 2024); GA performs gradient ascent in latent space. Appendix F details the datasets, reward models, evaluation protocol, and baseline implementations, and provides the SD1.5 results.

5.1 MAIN RESULTS

FASTER achieves the best target metric in all four SD3 tasks and leads several validation metrics (Table 1). For human preference, the ImageReward improvement is accompanied by the highest HPSv2 and CLIPScore; for counterfactual generation, both the target-class logit and validation CLIPScore lead the comparison. Facial editing similarly combines the highest target CLIPScore with the best ImageReward. In style transfer, FASTER achieves the lowest GramDiff and highest reference CLIP-I, although its DINO similarity remains below the best result. These gains extend to multiple metrics that are not directly optimized, although FASTER does not lead every validation metric.

Alongside these quality results, FASTER offers a substantial editing-time advantage over the baseline that optimizes controls along pretrained generation trajectories (Fig. 3). FASTER achieves

speedups of up to $6.91\times$ on SD3 and $24.14\times$ on SD1.5. This advantage is consistent with assigning repeated updates to a small network and reusing costly endpoint feedback. A 768×768 SD3 ImageReward edit takes approximately 7.7 s of editing time on a single consumer GPU. Appendix F gives the timing protocol, memory measurements, and comparisons with all baselines.

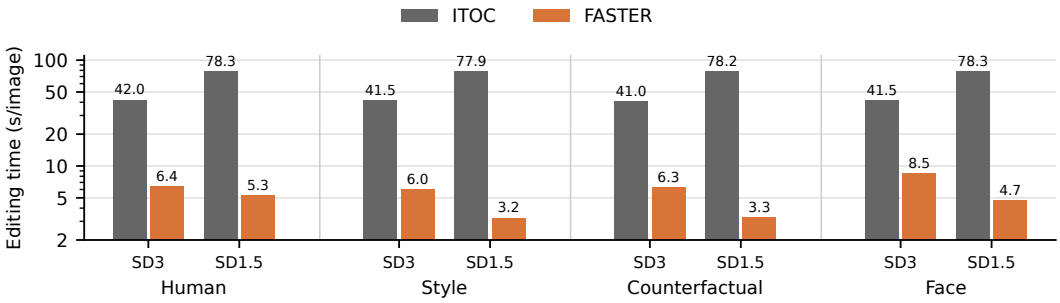

Figure 3: Editing time of ITOC and FASTER across four tasks on SD3 and SD1.5.

The qualitative examples in Fig. 4 show how these edits retain source content while changing the requested attributes. In the text-guided example, it produces a smile while largely preserving the man's facial appearance, hairstyle, and composition. The counterfactual example changes the lorikeet toward an African grey appearance while retaining the source bird's head pose and outline. In the human-preference example, the frog and surrounding scene remain recognizable. For style transfer, the rabbit retains its overall silhouette and pose while acquiring the silver-grey palette and metallic appearance of the reference train. Together, these examples illustrate targeted changes without wholesale replacement of the source composition.

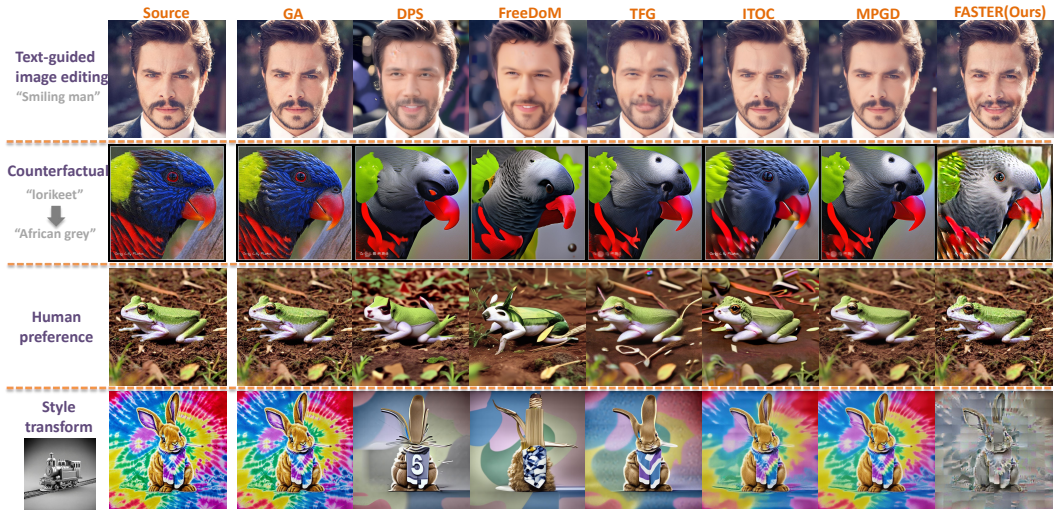

Figure 4: Qualitative comparisons on four SD3 editing tasks. Rows show text-guided image editing, counterfactual generation, human preference enhancement, and style transfer. For text-guided image editing, the target prompt is supplied to the pretrained generative model of each applicable method. The style reference is shown on the left.

5.2 Editing beyond CLIPScore

CLIPScore can assess image–text alignment, but maximizing it does not guarantee the requested semantic edit.

In Fig. 5, optimization raises CLIPScore from 0.223 to 0.356 without adding sunglasses. The Editing Term (Sec. 4.3) produces visible sunglasses despite a lower score of 0.271, supplying semantic

guidance beyond score maximization. Further Editing Term examples and parameter sweeps appear in Appendix G.



Figure 5: FASTER with CLIPScore optimization or the Editing Term for the same sunglasses request. Values are CLIPScore.

5.3 Reward strength and outer-loop budget

Figure 6 varies the outer-loop budget K and reward coefficient c_r on the ImageReward task. For the tested source–prompt pairs, a suitable c_r allows more outer-loop steps to improve ImageReward while validation metrics recover or remain strong at later checkpoints. In contrast, the largest c_r brings little additional ImageReward over the moderate settings, but HPSv2, CLIPScore, and Aesthetic all deteriorate as K grows. The gap between a saturated target reward and worsening validation metrics indicates reward hacking in these settings. These observations show that the benefit of increasing K depends on the reward strength. Appendix G also shows the visual progression with K when the Editing Term drives a larger semantic change.

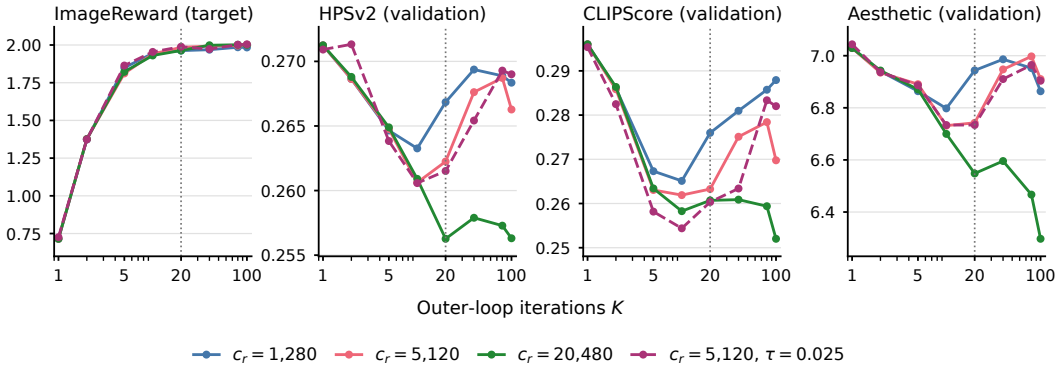

Figure 6: Effect of K and four parameter settings on the SD3 ImageReward task. Each curve averages the same three source–prompt pairs at saved checkpoints from one run per setting. All settings use $\alpha\tau^2 = 1$ and $c_p = 0$. Solid curves use $\tau = 0.0125$; the dashed curve uses $\tau = 0.025$. The dotted vertical line marks $K = 20$.

6 Conclusion

We introduced a decoupled view of reward-guided image editing at test time, in which pretrained models provide visual knowledge while a separate lightweight process executes the edit. Our theoretical framework jointly accounts for reward improvement, source preservation, and prior preferences, establishing a basis for specifying desirable outputs independently of pretrained generation trajectories. Building on this framework, FASTER trains a small per-image network and reuses candidate outputs and their costly feedback across multiple inexpensive updates. A complementary editing term obtains semantic directions from conditional model predictions without backpropagating through the pretrained backbone, supporting changes that image–text score maximization alone may fail to produce. Across four tasks, FASTER achieves competitive target and independent validation metrics, with editing-time speedups of up to $6.91\times$ on SD3 and $24.14\times$ on SD1.5 relative to the evaluated baseline that optimizes controls along pretrained generation trajectories.

AI USE STATEMENT

We used generative AI tools to assist with drafting parts of the manuscript and improving its wording and readability. The authors reviewed and revised the AI-assisted text and take responsibility for the final content of this paper, including its scientific claims and results.

REFERENCES

Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Qinsheng Zhang, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, Tero Karras, and Ming-Yu Liu. eDiff-I: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022. URL https://arxiv.org/abs/2211.01324.

Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 843–852, 2023. URL https://openaccess.thecvf.com/content/CVPR2023W/GCV/html/Bansal_Universal_Guidance_for_Diffusion_Models_CVPRW_2023_paper.html.

Heli Ben-Hamu, Omri Puny, Itai Gat, Brian Karrer, Uriel Singer, and Yaron Lipman. D-Flow: Differentiating through flows for controlled generation. In *Forty-first International Conference on Machine Learning*, volume 235, pp. 3462–3483. PMLR, 2024. URL https://proceedings.mlr.press/v235/ben-hamu24a.html.

Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. MasaCtrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22503–22513. IEEE, 2023.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. URL https://arxiv.org/abs/2104.14294.

Jinho Chang, Jaemin Kim, and Jong Chul Ye. Training-free reward-guided image editing via trajectory optimal control. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=YL2NgqN3Vh.

Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=OnD9zGAGT0k.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Carles Domingo-Enrich, Michal Drozdzal, Brian Karrer, and Ricky T. Q. Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=xQBRrtQM8u.

Logan Engstrom, Andrew Ilyas, Hadi Salman, Shibani Santurkar, and Dimitris Tsipras. Robustness (Python library), 2019. URL https://github.com/MadryLab/robustness.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 12606–12633. PMLR, 2024. URL https://proceedings.mlr.press/v235/esser24a.html.

Luca Eyring, Shyamgopal Karthik, Karsten Roth, Alexey Dosovitskiy, and Zeynep Akata. ReNO: Enhancing one-step text-to-image models through reward-based noise optimization. In *Advances in Neural Information Processing Systems*, volume 37, pp. 125487–125519. Curran Associates, Inc., 2024.

Berthy Feng and Katherine Bouman. Variational bayesian imaging with an efficient surrogate score-based prior. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=db2pFKVcm1.

Berthy T Feng, Jamie Smith, Michael Rubinstein, Huiwen Chang, Katherine L Bouman, and William T Freeman. Score-based diffusion models as principled priors for inverse imaging. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10486–10497. IEEE, 2023.

Aaron J Havens, Benjamin Kurt Miller, Bing Yan, Carles Domingo-Enrich, Anuroop Sriram, Daniel S. Levine, Brandon M Wood, Bin Hu, Brandon Amos, Brian Karrer, Xiang Fu, Guan-Horng Liu, and Ricky T. Q. Chen. Adjoint sampling: Highly scalable diffusion samplers via adjoint matching. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=6Eg1OrHmg2.

Yutong He, Naoki Murata, Chieh-Hsin Lai, Yuhta Takida, Toshimitsu Uesaka, Dongjun Kim, Wei-Hsiang Liao, Yuki Mitsufuji, J. Zico Kolter, Ruslan Salakhutdinov, and Stefano Ermon. Manifold preserving guided diffusion. In *International Conference on Learning Representations*, 2024.

Amir Hertz, Kfir Aberman, and Daniel Cohen-Or. Delta denoising score. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2328–2337. IEEE, 2023a.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross-attention control. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=_CDixzkzeyb.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7514–7528, 2021. URL https://aclanthology.org/2021.emnlp-main.595/.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. URL https://arxiv.org/abs/2207.12598.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020.

Jerry Y Huang, Justin Lin, Sheel Shah, Kartik Nair, and Nicholas M Boffi. How to guide your flow: Few-step alignment via flow map reward guidance. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://github.com/jrrhuang/fmrg.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1710.10196.

Sunwoo Kim, Minkyu Kim, and Dongmin Park. Test-time alignment of diffusion models without reward over-optimization. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=vi3DjUhFVm.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-Pic: An open dataset of user preferences for text-to-image generation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 36652–36663. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/73aacd8b3b05b4b503d58310b523553c-Abstract-Conference.html.

Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. FlowEdit: Inversion-free text-based editing using pre-trained flow models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19721–19730. IEEE, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

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

Morteza Mardani, Jiaming Song, Jan Kautz, and Arash Vahdat. A variational perspective on solving inverse problems with diffusion models. In *International Conference on Learning Representations*, 2024.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022.

George A. Miller. WordNet: A lexical database for English. *Communications of the ACM*, 38(11): 39–41, 1995. doi: 10.1145/219717.219748.

Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6038–6047. IEEE, 2023.

Mokai Pan, Kaizhen Zhu, Yuexin Ma, Yanwei Fu, Jingyi Yu, Jingya Wang, and Ye Shi. A unified and fast-sampling diffusion bridge framework via stochastic optimal control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026. doi: 10.1109/TPAMI.2026.3710696. URL https://doi.org/10.1109/TPAMI.2026.3710696.

Maitreya Patel, Song Wen, Dimitris N Metaxas, and Yezhou Yang. FlowChef: Steering of rectified flow models for controlled generations. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15308–15318. IEEE, 2025.

Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D diffusion. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=FjNys5c7VyY.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, volume 139, pp. 8748–8763. PMLR, 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10674–10685. IEEE, 2022.

Jeongwoo Shin, Dongsoo Shin, Yuchen Zhu, Wei Guo, Yongxin Chen, Joonseok Lee, Jaewoong Choi, and Jaemoo Choi. Efficient adjoint matching for fine-tuning diffusion models. *arXiv preprint arXiv:2605.11480*, 2026.

Raghav Singhal, Zachary Horvitz, Ryan Teehan, Mengye Ren, Zhou Yu, Kathleen McKeown, and Rajesh Ranganath. A general framework for inference-time scaling and steering of diffusion models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=Jp988ELppQ.

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

Onkar Susladkar, Dong-Hwan Jang, Tushar Prakash, Adheesh Juvekar, Vedant Shah, Ayush Barik, Nabeel Bashir, Muntasir Wahed, Ritish Shrirao, and Ismini Lourentzou. RewardFlow: Generate images by optimizing what you reward. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20066–20076, June 2026.

Zhiwei Tang, Jiangweizhi Peng, Jiasheng Tang, Mingyi Hong, Fan Wang, and Tsung-Hui Chang. Inference-time alignment of diffusion models with direct noise optimization. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=JpbqiD7n9r.

Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1921–1930. IEEE, 2023.

Bram Wallace, Akash Gokul, Stefano Ermon, and Nikhil Naik. End-to-end diffusion latent optimization improves classifier guidance. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7246–7256. IEEE, 2023.

Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. ProlificDreamer: High-fidelity and diverse text-to-3D generation with variational score distillation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 8406–8441. Curran Associates, Inc., 2023.

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. URL https://arxiv.org/abs/2306.09341.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. ImageReward: Learning and evaluating human preferences for text-to-image generation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 15903–15935. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/33646ef0ed554145eab65f6250fab0c9-Abstract-Conference.html.

Sihan Xu, Yidong Huang, Jiayi Pan, Ziqiao Ma, and Joyce Chai. Inversion-free image editing with language-guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9452–9461, June 2024.

Lingxiao Yang, Shutong Ding, Yifan Cai, Jingyi Yu, Jingya Wang, and Ye Shi. Guidance with spherical gaussian constraint for conditional diffusion. In *Forty-first International Conference on Machine Learning*, volume 235, pp. 56071–56095. PMLR, 2024. URL https://proceedings.mlr.press/v235/yang24h.html.

Haotian Ye, Haowei Lin, Jiaqi Han, Minkai Xu, Sheng Liu, Yitao Liang, Jianzhu Ma, James Zou, and Stefano Ermon. TFG: Unified training-free guidance for diffusion models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 22370–22417. Curran Associates, Inc., 2024.

Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. FreeDoM: Training-free energy-guided conditional diffusion model. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 23117–23127. IEEE, 2023.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018. URL https://arxiv.org/abs/1801.03924.

Kaizhen Zhu, Mokai Pan, Yuexin Ma, Yanwei Fu, Jingyi Yu, Jingya Wang, and Ye Shi. UniDB: A unified diffusion bridge framework via stochastic optimal control. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 79985–80012. PMLR, 2025. URL https://proceedings.mlr.press/v267/zhu25o.html.

A Source-Conditioned Target: Derivation and Scope

A.1 Per-image variational objective

For a fixed source $\boldsymbol{s}$, consider $F_{\boldsymbol{s}}$ in Eq. (4). Assume p_{pre} is a density and the following normalizer is finite and positive:

$$Z_{\boldsymbol{s}}^{(1)} = \int p_{\mathrm{pre}}(\boldsymbol{y}) \exp[-\alpha D(\boldsymbol{s}, \boldsymbol{y}) + \beta r(\boldsymbol{y})] d\boldsymbol{y}, \tag{17a}$$

$$\pi_{\boldsymbol{s}}^{(1)}(\boldsymbol{y}) = (Z_{\boldsymbol{s}}^{(1)})^{-1} p_{\mathrm{pre}}(\boldsymbol{y}) \exp[-\alpha D(\boldsymbol{s}, \boldsymbol{y}) + \beta r(\boldsymbol{y})]. \tag{17b}$$

For densities with finite objective, substitution gives

$$D_{\mathrm{KL}}(\rho \| \pi_{\boldsymbol{s}}^{(1)}) = D_{\mathrm{KL}}(\rho \| p_{\mathrm{pre}}) + \alpha \mathbb{E}_\rho D(\boldsymbol{s}, Y) - \beta \mathbb{E}_\rho r(Y) + \log Z_{\boldsymbol{s}}^{(1)}, \tag{18a}$$

$$F_{\boldsymbol{s}}(\rho) = D_{\mathrm{KL}}(\rho \| \pi_{\boldsymbol{s}}^{(1)}) - \log Z_{\boldsymbol{s}}^{(1)}. \tag{18b}$$

Nonnegativity of KL proves the unique minimizer, up to equality almost everywhere. The tempered target in Eq. (5) permits a separate prior weight. Its corresponding functional is

$$F_{\boldsymbol{s}}^{(\gamma)}(\rho) = \int \rho(\boldsymbol{y}) \log \rho(\boldsymbol{y}) d\boldsymbol{y} - \gamma \mathbb{E}_\rho \log p_{\mathrm{pre}}(Y) + \alpha \mathbb{E}_\rho D(\boldsymbol{s}, Y) - \beta \mathbb{E}_\rho r(Y), \tag{19a}$$

$$= D_{\mathrm{KL}}(\rho \| \pi_{\boldsymbol{s}}) - \log Z_{\boldsymbol{s}}. \tag{19b}$$

This identity requires finite terms and a finite positive $Z_{\boldsymbol{s}}$; it does not require the prior power to be separately normalized.

A.2 Conditional regularization versus marginal matching

For sources $S \sim \mu$, let $T(\boldsymbol{y} \mid \boldsymbol{s})$ be the output density, $j_T(d\boldsymbol{s}, d\boldsymbol{y}) = \mu(d\boldsymbol{s}) T(\boldsymbol{y} \mid \boldsymbol{s}) d\boldsymbol{y}$, and p_T its output marginal. Under integrability, source-wise minimizers also minimize $\mathbb{E}_S F_S(T(\cdot \mid S))$. The identity

$$\mathbb{E}_S D_{\mathrm{KL}}(T(\cdot \mid S) \| p_{\mathrm{pre}}) = I_{j_T}(S; Y) + D_{\mathrm{KL}}(p_T \| p_{\mathrm{pre}}) \tag{20}$$

shows that conditional prior regularization includes both mutual information and marginal matching. The source-distance term supplies the explicit preservation preference, without imposing a hard per-sample distance bound.

B Endpoint Realization and Its Scope

Proposition 1 (Endpoint realization)**.** *Let Q_τ be the zero-drift Brownian path law from Eq.* (6)*, q_τ its endpoint density, and $P_{\boldsymbol{s}}^{\boldsymbol{b}}$ the path law with drift $\mathbf{b}$. Suppose the residual endpoint density $\bar{\pi}_{\boldsymbol{s}}$ is positive, has finite $D_{\mathrm{KL}}(\bar{\pi}_{\boldsymbol{s}} \| q_\tau)$, and its tilted path law*

$$\frac{dP_{\boldsymbol{s}}^\star}{dQ_\tau} = \frac{\bar{\pi}_{\boldsymbol{s}}(\boldsymbol{z}_1)}{q_\tau(\boldsymbol{z}_1)} \tag{21}$$

is realizable by an admissible control. Under the Girsanov and integrability conditions in Appendix B.1,

$$J_{\boldsymbol{s}}(\boldsymbol{b}) = D_{\mathrm{KL}}(P_{\boldsymbol{s}}^{\boldsymbol{b}} \| P_{\boldsymbol{s}}^\star). \tag{22}$$

The unrestricted optimum therefore satisfies $\boldsymbol{s} + \boldsymbol{z}_1 \sim \pi_{\boldsymbol{s}}$.

B.1 Assumptions and proof of Proposition 1

Work on continuous residual paths starting at zero. Assume that $\bar{\pi}_{\boldsymbol{s}}$ is positive and normalized, with finite $D_{\mathrm{KL}}(\bar{\pi}_{\boldsymbol{s}} \| q_\tau)$. For each candidate control, assume the Girsanov stochastic exponential is a true martingale, and the relevant energies and logarithmic expectations are finite. Attaining the optimum within a controlled SDE also requires realizability of the tilted law. Smoothness of a target alone does not establish these conditions.

Set $w_{\boldsymbol{s}}(\boldsymbol{z}) = \bar{\pi}_{\boldsymbol{s}}(\boldsymbol{z})/q_\tau(\boldsymbol{z})$. Its expectation is

$$\mathbb{E}_{Q_\tau}[w_{\boldsymbol{s}}(\boldsymbol{z}_1)] = \int q_\tau(\boldsymbol{z}) \frac{\bar{\pi}_{\boldsymbol{s}}(\boldsymbol{z})}{q_\tau(\boldsymbol{z})} d\boldsymbol{z} = 1. \tag{23}$$

Consequently $dP^\star_{\boldsymbol{s}}/dQ_\tau = w_{\boldsymbol{s}}(\boldsymbol{z}_1) = e^{-g_{\boldsymbol{s}}(\boldsymbol{z}_1)}$ defines a probability measure. For any measurable endpoint set A,

$$P^\star_{\boldsymbol{s}}(\boldsymbol{z}_1 \in A) = \int_A q_\tau(\boldsymbol{z}) w_{\boldsymbol{s}}(\boldsymbol{z}) d\boldsymbol{z} = \int_A \bar{\pi}_{\boldsymbol{s}}(\boldsymbol{z}) d\boldsymbol{z}. \tag{24}$$

The tilt depends only on the endpoint, so the conditional path remains the reference bridge. Expanding the path KL gives

$$\begin{aligned} D_{\mathrm{KL}}(P^{\boldsymbol{b}}_{\boldsymbol{s}} \| P^\star_{\boldsymbol{s}}) &= \mathbb{E}_{P^{\boldsymbol{b}}_{\boldsymbol{s}}}\left[\log \frac{dP^{\boldsymbol{b}}_{\boldsymbol{s}}}{dQ_\tau} - \log \frac{dP^\star_{\boldsymbol{s}}}{dQ_\tau}\right] && \text{(25a)} \\ &= D_{\mathrm{KL}}(P^{\boldsymbol{b}}_{\boldsymbol{s}} \| Q_\tau) + \mathbb{E}_{P^{\boldsymbol{b}}_{\boldsymbol{s}}} g_{\boldsymbol{s}}(\boldsymbol{z}_1) && \text{(25b)} \\ &= \mathbb{E}_{P^{\boldsymbol{b}}_{\boldsymbol{s}}}\left[\int_0^1 \frac{\|\boldsymbol{b}(\boldsymbol{z}_\lambda, \lambda; \boldsymbol{s})\|^2}{2\tau^2} d\lambda + g_{\boldsymbol{s}}(\boldsymbol{z}_1)\right]. && \text{(25c)} \end{aligned}$$

The last equality uses Girsanov's path-energy identity. KL is minimized at the target path law whenever that law is realizable. Translation by $\boldsymbol{s}$ has unit Jacobian, giving output density $\pi_{\boldsymbol{s}}$.

We used normalized densities in $g_{\boldsymbol{s}}$. With the unnormalized energy $F_{\boldsymbol{s}} = \alpha D(\boldsymbol{s}, \cdot) - \gamma \log p_{\mathrm{pre}} - \beta r$, implementation may use $g^0_{\boldsymbol{s}} = \log q_\tau + F_{\boldsymbol{s}}(\boldsymbol{s} + \cdot)$. Then $g^0_{\boldsymbol{s}} = g_{\boldsymbol{s}} - \log \mathcal{Z}_{\boldsymbol{s}}$ and $J^0 = D_{\mathrm{KL}}(P^{\boldsymbol{b}}_{\boldsymbol{s}} \| P^\star_{\boldsymbol{s}}) - \log \mathcal{Z}_{\boldsymbol{s}}$; gradients and minimizers are unchanged.

B.2 The realizing drift

Assume

$$h_{\boldsymbol{s}}(\boldsymbol{z}, \lambda) = \mathbb{E}_{Q_\tau}[e^{-g_{\boldsymbol{s}}(\boldsymbol{z}_1)} \mid \boldsymbol{z}_\lambda = \boldsymbol{z}] \tag{26}$$

is finite, positive and $C^{1,2}$ at interior times, and its logarithmic gradient defines an admissible non-explosive diffusion. It solves the backward heat equation. Itô's formula under Q_τ gives

$$dh_{\boldsymbol{s}}(\boldsymbol{z}_\lambda, \lambda) = \tau \nabla h_{\boldsymbol{s}}(\boldsymbol{z}_\lambda, \lambda)^\top d\boldsymbol{w}_\lambda. \tag{27}$$

The normalized process is the conditional likelihood-ratio martingale for the endpoint tilt. Girsanov therefore adds drift

$$\boldsymbol{b}^\star(\boldsymbol{z}, \lambda; \boldsymbol{s}) = \tau^2 \nabla \log h_{\boldsymbol{s}}(\boldsymbol{z}, \lambda). \tag{28}$$

This realizes the desired path measure under the assumptions. A finite neural family need not contain this drift.

B.3 Reference scale and source dependence

For a fixed target, $q_\tau e^{-g_{\boldsymbol{s}}} = \bar{\pi}_{\boldsymbol{s}}$, so changing τ does not change the ideal endpoint. It does change the optimal drift, bridge geometry and numerical behavior. The source-distance penalty and conditional prior regularization have distinct roles, as shown by the mutual-information decomposition in Sec. A.2.

C Adjoint Sampling and Endpoint Regression

Proposition 2 (Consistency of endpoint regression)**.** *Consider the population counterpart of Eq.* (12)*: draw $\boldsymbol{z}_1$ from $\bar{\pi}_{\boldsymbol{s}}$, draw $\boldsymbol{z}_\lambda$ from its Brownian bridge, and replace the cached approximate label by the exact clean-score label $\boldsymbol{b}^{\mathrm{clean}}_{\boldsymbol{s}}(\boldsymbol{z}_1)$ in Eq.* (33)*. Assume the labels have finite second moment and the target satisfies the regularity and tail conditions used below. With $h_{\boldsymbol{s}}(\boldsymbol{z}, \lambda) = \mathbb{E}_{Q_\tau}[e^{-g_{\boldsymbol{s}}(\boldsymbol{z}_1)} \mid \boldsymbol{z}_\lambda = \boldsymbol{z}]$, the unrestricted population minimizer of this squared-error objective is*

$$\boldsymbol{b}^\star(\boldsymbol{z}, \lambda; \boldsymbol{s}) = \tau^2 \nabla \log h_{\boldsymbol{s}}(\boldsymbol{z}, \lambda) = \mathbb{E}_{P^\star_{\boldsymbol{s}}}[\boldsymbol{b}^{\mathrm{clean}}_{\boldsymbol{s}}(\boldsymbol{z}_1) \mid \boldsymbol{z}_\lambda = \boldsymbol{z}]. \tag{29}$$

C.1 Reciprocal projection and path KL

For a path law P with endpoint marginal P_1, define

$$\mathcal{R}_{Q_\tau}(P)(d\omega) = \int Q_\tau(d\omega \mid Z_1 = \boldsymbol{z})\, P_1(d\boldsymbol{z}), \tag{30}$$

where ω is a residual path starting at zero.

Let P and Q_τ share the deterministic initial state, with $P \ll Q_\tau$ and finite path KL. Disintegrating over the terminal state gives

$$D_{\mathrm{KL}}(P\|Q_\tau) = D_{\mathrm{KL}}(P_1\|q_\tau) + \mathbb{E}_{z\sim P_1} D_{\mathrm{KL}}(P(\cdot \mid Z_1 = z)\|Q_\tau(\cdot \mid Z_1 = z)). \tag{31}$$

For $\mathcal{R}_{Q_\tau}(P)$ the second term vanishes and the first is unchanged. Terminal expectations are also unchanged, so the path-space objective $D_{\mathrm{KL}}(P\|Q_\tau) + \mathbb{E}_P g_s(Z_1)$ cannot increase. Interpreting the projected law as an admissible drift process additionally requires realizability and the usual Girsanov conditions. This is a property of the exact projection, not a monotonicity guarantee for finite neural regression steps.

C.2 Control scaling and reference bridges

Adjoint Matching formulates control learning as squared-error regression onto a lean-adjoint target (Domingo-Enrich et al., 2025). Using the normalized-control convention of Adjoint Sampling (Havens et al., 2025), let $\boldsymbol{u} = \boldsymbol{b}/\tau$. The physical drift is $\tau\boldsymbol{u}$ and the energy is $\|\boldsymbol{u}\|^2/2$. Since the reference drift is zero and there is no running state cost, the lean reference adjoint has $\widetilde{\boldsymbol{a}}_\lambda = \nabla g_{\boldsymbol{s}}(\boldsymbol{z}_1)$ by Eq. (3). The corresponding control-matching residual is

$$\frac{1}{2}\left\|\boldsymbol{u}_\theta + \tau\widetilde{\boldsymbol{a}}_\lambda\right\|^2 = \frac{1}{2\tau^2}\left\|\boldsymbol{b}_\theta + \tau^2\nabla g_{\boldsymbol{s}}(\boldsymbol{z}_1)\right\|^2. \tag{32}$$

Reciprocal Adjoint Matching replaces controlled intermediate states by reference-bridge samples conditional on endpoints from the current controller (Havens et al., 2025). Together with the physical-drift residual above, this gives Eq. (12) when endpoint gradients are cached and the ideal score is replaced by its estimator. The factor $1/(2\tau^2)$ is fixed and positive, so we omit it in the main-text MSE; the unrestricted minimizer is unchanged. Dividing by the latent dimension d would likewise change only the overall loss scale. The adjoint refers to the zero-drift reference dynamics; it does not assert that the nonlinear learned controller has a constant exact adjoint. Expanding $g_{\boldsymbol{s}} = \log q_\tau - \log \bar{\pi}_{\boldsymbol{s}}$ gives the exact clean-score label before introducing a practical score estimator:

$$\boldsymbol{b}_{\boldsymbol{s}}^{\mathrm{clean}}(\boldsymbol{z}_1) := -\tau^2\nabla_{\boldsymbol{z}_1} g_{\boldsymbol{s}}(\boldsymbol{z}_1) \tag{33a}$$

$$= \boldsymbol{z}_1 + \tau^2[\gamma\nabla_{\boldsymbol{y}}\log p_{\mathrm{pre}}(\boldsymbol{y}) - \alpha\nabla_{\boldsymbol{y}} D(\boldsymbol{s},\boldsymbol{y}) + \beta\nabla_{\boldsymbol{y}} r(\boldsymbol{y})]_{\boldsymbol{y}=\boldsymbol{s}+\boldsymbol{z}_1}. \tag{33b}$$

The $\boldsymbol{z}_1$ term comes from $-\tau^2\nabla\log q_\tau(\boldsymbol{z}_1)$; inserting the quadratic preservation cost gives the clean analogue of Eq. (11). Replacing the clean score by $\widehat{\boldsymbol{s}}_{\mathrm{pre}}$ yields the implemented label in that equation.

Under Q_τ, $\mathrm{Cov}(\boldsymbol{z}_\lambda, \boldsymbol{z}_1) = \tau^2\lambda\boldsymbol{I}$, $\mathrm{Var}(\boldsymbol{z}_\lambda) = \tau^2\lambda\boldsymbol{I}$ and $\mathrm{Var}(\boldsymbol{z}_1) = \tau^2\boldsymbol{I}$. Gaussian conditioning gives mean $\lambda\boldsymbol{z}_1$ and covariance $\tau^2\lambda\boldsymbol{I} - (\tau^2\lambda\boldsymbol{I})(\tau^2\boldsymbol{I})^{-1}(\tau^2\lambda\boldsymbol{I}) = \tau^2\lambda(1-\lambda)\boldsymbol{I}$, proving Eq. (6).

C.3 Proof of Proposition 2

Let $k(\boldsymbol{v}-\boldsymbol{z})$ denote the Gaussian transition density with covariance $\tau^2(1-\lambda)\boldsymbol{I}$. Then

$$h_{\boldsymbol{s}}(\boldsymbol{z},\lambda) = \int k(\boldsymbol{v}-\boldsymbol{z})e^{-g_{\boldsymbol{s}}(\boldsymbol{v})}d\boldsymbol{v}. \tag{34}$$

Assume differentiation may pass under the integral and boundary terms vanish. Using $\nabla_{\boldsymbol{z}}k = -\nabla_{\boldsymbol{v}}k$ and integrating by parts,

$$\nabla_{\boldsymbol{z}} h_{\boldsymbol{s}} = -\int \nabla_{\boldsymbol{v}} k(\boldsymbol{v}-\boldsymbol{z})e^{-g_{\boldsymbol{s}}(\boldsymbol{v})}d\boldsymbol{v} \tag{35a}$$

$$= \int k(\boldsymbol{v}-\boldsymbol{z})\nabla_{\boldsymbol{v}} e^{-g_{\boldsymbol{s}}(\boldsymbol{v})}d\boldsymbol{v} \tag{35b}$$

$$= -\int k(\boldsymbol{v}-\boldsymbol{z})e^{-g_{\boldsymbol{s}}(\boldsymbol{v})}\nabla g_{\boldsymbol{s}}(\boldsymbol{v})d\boldsymbol{v}. \tag{35c}$$

The conditional endpoint density under $P_{\boldsymbol{s}}^{\star}$ is $k(\boldsymbol{v}-\boldsymbol{z})e^{-g_{\boldsymbol{s}}(\boldsymbol{v})}/h_{\boldsymbol{s}}(\boldsymbol{z},\lambda)$. Dividing by $h_{\boldsymbol{s}}$ and multiplying by τ^2 gives Eq. (29).

The endpoint tilt leaves reference conditional bridges unchanged; hence sampling an endpoint from $\bar{\pi}_{\boldsymbol{s}}$ and a reference bridge produces the joint marginal of $P_{\boldsymbol{s}}^{\star}$. For the square-integrable ideal label $\boldsymbol{b}_{\boldsymbol{s}}^{\mathrm{clean}}(\boldsymbol{z}_1)$, let $\boldsymbol{m}(\boldsymbol{z}_\lambda) = \mathbb{E}_{P_{\boldsymbol{s}}^{\star}}[\boldsymbol{b}_{\boldsymbol{s}}^{\mathrm{clean}}(\boldsymbol{z}_1) \mid \boldsymbol{z}_\lambda]$. Conditional-variance decomposition writes the expected squared error as $\|\boldsymbol{b}(\boldsymbol{z}_\lambda,\lambda;\boldsymbol{s}) - \boldsymbol{m}(\boldsymbol{z}_\lambda)\|^2$ plus a term independent of the predictor; the cross term vanishes by the definition of $\boldsymbol{m}$. The optimal predictor is therefore the conditional mean derived above.

C.4 Finite updates and replay

For a general frozen endpoint law ρ_k, the regression mean is taken under $\rho_k(\boldsymbol{z}_1)Q_\tau(d\boldsymbol{z}_\lambda \mid \boldsymbol{z}_1)$. The proof establishes the target fixed point, not a one-step exact solution or global convergence. In practice, endpoints follow the current controller law and Eq. (11) uses an approximate prior score; either change breaks the exact fixed-point premise. Equation (12) conditions on one detached endpoint and label per refresh. This empirical objective need not have the population minimizer even if that endpoint came from the target. Resampling bridges supplies stochastic gradients of this frozen empirical objective, not new independent terminal information. The next refresh replaces the stored pair; retaining older endpoints would instead define a replay variant.

D Endpoint Prior and Reward Oracles

The ideal theory uses the clean prior. A conservative surrogate replaces its energy $-\log p_{\mathrm{pre}}$ in Eq. (5). An exact clean score, a noisy diffusion score, and an arbitrary guidance vector are not interchangeable. One possible explicit surrogate is as follows. Let ℓ be a noise level sampled from a fixed distribution, a_ℓ and $\nu_\ell > 0$ fixed corruption coefficients, p_ℓ^{pre} the corresponding pretrained noisy density, and $\boldsymbol{\epsilon}$ standard Gaussian noise. Define

$$\mathcal{P}_{\mathrm{noise}}(\boldsymbol{y}) = \mathbb{E}_{\ell,\boldsymbol{\epsilon}}[-\log p_\ell^{\mathrm{pre}}(a_\ell \boldsymbol{y} + \nu_\ell \boldsymbol{\epsilon})]. \tag{36}$$

If differentiation and expectation may be exchanged, its gradient is $-\mathbb{E}_{\ell,\boldsymbol{\epsilon}}[a_\ell \nabla \log p_\ell^{\mathrm{pre}}(a_\ell \boldsymbol{y} + \nu_\ell \boldsymbol{\epsilon})]$. A pretrained score predictor can supply a Monte Carlo estimate by forward evaluation. This avoids differentiating the score network, but targets $\mathcal{P}_{\mathrm{noise}}$, not necessarily $-\log p_{\mathrm{pre}}$. For example, when the score predictor returns $\nabla \log p_\ell^{\mathrm{pre}}$ at the corrupted input, a single draw gives

$$\widehat{\boldsymbol{s}}_{\mathrm{pre}}(\boldsymbol{y};\ell,\boldsymbol{\epsilon}) = a_\ell \nabla \log p_\ell^{\mathrm{pre}}(a_\ell \boldsymbol{y} + \nu_\ell \boldsymbol{\epsilon}), \qquad \mathbb{E}_{\ell,\boldsymbol{\epsilon}}[\widehat{\boldsymbol{s}}_{\mathrm{pre}}] = -\nabla \mathcal{P}_{\mathrm{noise}}(\boldsymbol{y}). \tag{37}$$

If this estimator is exact in expectation and the normalizer exists, it is associated with the surrogate endpoint density $\widetilde{\pi}_{\boldsymbol{s}}(\boldsymbol{y}) \propto \exp[-\gamma \mathcal{P}_{\mathrm{noise}}(\boldsymbol{y}) - \alpha D(\boldsymbol{s},\boldsymbol{y}) + \beta r(\boldsymbol{y})]$. The score estimator still varies between queries and the surrogate generally differs from the clean-prior target. For the quadratic cost $D(\boldsymbol{s},\boldsymbol{y}) = \|\boldsymbol{y}-\boldsymbol{s}\|^2/2$, we have $\nabla_{\boldsymbol{y}} D = \boldsymbol{z}_1$. Together with $\nabla \log q_\tau(\boldsymbol{z}_1) = -\boldsymbol{z}_1/\tau^2$, Eq. (33) reduces to Eq. (11) after the clean prior score is replaced by $\widehat{\boldsymbol{s}}_{\mathrm{pre}}$. If the estimate is the Monte Carlo gradient of Eq. (36), its expectation is a gradient field, but it corresponds to a noisy-prior surrogate rather than the clean density in Eq. (5). A single sampled estimate has variance; the other approximations in the score predictor may add bias.

The prior-query settings are given in Appendix F. Inexact score predictions need not recover the clean-prior target; Efficient Adjoint Matching also discusses the difficulty of obtaining endpoint scores (Shin et al., 2026).

For image-space rewards, gradients must pass from $R(\mathcal{D}(\boldsymbol{y}))$ back to $\boldsymbol{y}$ even though both networks' weights are frozen. Intermediate activation graphs are released after obtaining the detached endpoint gradient. Source latents and fixed text features can be cached. Learned controller features can only be cached across updates if their producing parameters do not change.

E Controller and Numerical Implementation

The main-text parameterization in Eq. (8) combines a trainable latent offset with a state- and time-dependent convolutional correction. We give its architecture and numerical implementation here.

For a latent tensor of shape $C \times H \times W$, the controller concatenates $\boldsymbol{z}$, $\boldsymbol{s}$ and a broadcast scalar time into $2C + 1$ channels. The nonlinear branch consists of three 3×3 convolutions with padding one, channel widths $(2C + 1, h, h, C)$, and SiLU after the first two layers. The translation contains CHW trainable scalars. Including all biases, the total number of parameters is

$$N_\theta = CHW + 9h^2 + 27Ch + 11h + C. \tag{38}$$

For example, $C = 4$, $H = W = 64$ and $h = 16$ give 20,596 parameters including the translation. This is an architecture example, not a claim that every experimental setting uses those dimensions. The translation and final convolution start at zero; other layers use their usual initialization.

Euler–Maruyama simulates Eq. (7) with $\Delta\lambda = 1/N$ and $N = 16$ and independent $\boldsymbol{\xi}_n \sim \mathcal{N}(0, \boldsymbol{I})$:

$$\boldsymbol{z}_{n+1} = \boldsymbol{z}_n + \Delta\lambda\, \boldsymbol{b}_\theta(\boldsymbol{z}_n, n\Delta\lambda; \boldsymbol{s}) + \tau\sqrt{\Delta\lambda}\, \boldsymbol{\xi}_n. \tag{39}$$

Endpoint simulation is detached from controller optimization. The prior model remains frozen and is queried outside the SDE steps. Reward-input gradients still pass through the frozen decoder and reward network; freezing weights does not eliminate those derivatives. After fitting, a fresh independent trajectory produces the output, rather than selecting the best training endpoint.

Experimental settings are listed in Appendix F and Table 2. Each refresh replaces the cached endpoint and label; older endpoints are not retained.

E.1 Endpoint-query reuse

With K endpoint refreshes and one candidate per refresh, the basic loop computes K endpoint targets, up to the chosen number of noisy-score samples per candidate. Its KM controller updates reuse those targets and draw new bridge times and noises. Each refresh also requires N evaluations of the small controller for Euler–Maruyama simulation. These counts describe the algorithmic placement of expensive queries; actual runtime depends on decoder, score-model, controller, and optimizer costs. Differentiation through the decoder is confined to the endpoint-query step. Since the targets are detached, the inner regression loop does not build or retain computation graphs for the pretrained model.

F Evaluation Protocol and Baseline Details

Frozen inputs. We instantiate the four tasks of Chang et al. (2026) with 300 frozen source/condition pairs per task. Human-preference sources use prompts from the Reward Feedback Learning (ReFL) data released with ImageReward (Xu et al., 2023); style sources are from the Pick-a-Pic test split (Kirstain et al., 2023) and are paired with five fixed style-reference images; counterfactual sources are ImageNet-1k images (Deng et al., 2009); facial-edit sources are CelebA-HQ images (Karras et al., 2018) with single-attribute text requests. All methods for a given model/task use the same source-image hashes, target text or class, and style reference when applicable. For counterfactual targets, we select a nearby ImageNet class using a WordNet IS-A graph (Miller, 1995).

Baseline implementations.

- **ITOC** (Chang et al., 2026) iteratively optimizes an image-conditioned sampling trajectory using reward adjoints.
- **Gradient Ascent (GA)** directly updates the editable latent using the task-reward gradient.
- **DPS** (Chung et al., 2023) applies posterior-style reward guidance during reverse sampling after source inversion. This is the source-editing adaptation used in the ITOC comparison suite.
- **FreeDoM** (Yu et al., 2023) uses its training-free energy guidance after source inversion, with the same frozen source set.
- **TFG** (Ye et al., 2024) uses training-free reward guidance during the corresponding inversion-conditioned reverse trajectory.

- **MPGD** (He et al., 2024) updates a predicted clean endpoint with the reward gradient. On SD1.5 we use an MPGD-LDM source-edit variant; on SD3 we use a flow-edit adaptation.

The model, source set, task reward, and evaluator are paired within each comparison. Differences in inversion and guidance mechanisms are intrinsic to the baselines. We report mean metric values.

Target and validation definitions. Human preference directly optimizes ImageReward and additionally reports HPSv2, image–text CLIPScore (Hessel et al., 2021), computed as the cosine similarity of CLIP ViT-L/14 image and text embeddings, and Aesthetic. Style directly minimizes the released positive GramDiff and validates with CLIP-I similarity to the style reference and DINO ViT-S/16 similarity. Counterfactual editing optimizes the robust classifier's target-class logit and validates with CLIPScore for the target-class text. Facial editing optimizes CLIPScore for the attribute request and validates with ImageReward and HPSv2. Source preservation is additionally available as LPIPS (Zhang et al., 2018) and source–output CLIP-I in the per-image evaluator records; these auxiliary columns are kept separate from the target/validation tables.

The reward implementations use ImageReward-v1.0 (Xu et al., 2023), OpenAI CLIP ViT-L/14 for image–text cosine similarity (Radford et al., 2021), and the robust ResNet-50 checkpoint `imagenet_l2_3_0.pt` distributed with the robustness library (Engstrom et al., 2019). The counterfactual reward is the target logit, not a target-text score. Style GramDiff uses non-class tokens from `hidden_states[2]` of `openai/clip-vit-base-patch16`; the optimization reward is the negative Frobenius difference between the output and reference Gram matrices, scaled by $1/100$. The table reports the corresponding positive difference. DINO validation uses ViT-S/16 (Caron et al., 2021); HPSv2 follows Wu et al. (2023). The Aesthetic evaluator uses the released CLIP-feature predictor from `https://github.com/christophschuhmann/improved-aesthetic-predictor`.

Paired comparison and selection. Every model–task comparison uses the same 300 source–condition pairs for all seven methods. FASTER parameters were selected using earlier pilot evaluations, before the full-set evaluation. In the tables, bold values indicate the best result for each metric. All required edited-image metrics are finite.

FASTER configurations. The selected runs use $K = 20$ outer refreshes and $M = 2$ controller updates per refresh. The latent SDE uses 16 steps for both SD1.5 and SD3, at working resolutions of 512^2 and 768^2, respectively. Each refresh samples one endpoint; each controller update independently draws four control times λ and four Gaussian noises conditioned on that endpoint, then averages their regression losses for one optimizer step. The controller has 16 hidden channels, a trainable terminal translation, and a nonlinear branch. We fit it with mean-squared error and Adam, using learning rates 0.02 for the translation and 0.001 for the branch. Bridge times are uniform on $[0, 1]$; the seed is zero. Each enabled prior query uses one sample with antithetic sampling enabled. The prior-score query window is diffusion steps $[100, 400]$ for SD1.5 and flow noise times $[0.1, 0.4]$ for SD3. The SD3 style objective uses symmetric bicubic preprocessing; the SD1.5 style objective uses the released preprocessing. The quantitative benchmark runs do not use the optional Editing Term; its separate noise-time windows are reported in Appendix G. For every row below, $\alpha = (\alpha\tau^2)/\tau^2$, $\beta = c_r/\tau^2$, and $\gamma = c_p/\tau^2$.

Table 2: Selected FASTER settings for the quantitative benchmarks.

Backbone	Task	$\alpha\tau^2$	τ	c_r	c_p
SD1.5	Human	1	0.0125	320	0.1
SD1.5	Style	0.01	0.002	10240	0
SD1.5	Counterfactual	1	0.002	2560	0
SD1.5	Facial edit	1	0.00625	1280	0.01
SD3	Human	1	0.0125	320	0
SD3	Style	0.01	0.0125	5120	0
SD3	Counterfactual	1	0.0125	20480	0
SD3	Facial edit	1	0.0125	20480	0.1

Baseline configurations. Table 3 lists the task-dependent settings. For ITOC, (n, w, ℓ) denotes trajectory-optimization iterations, reward weight, and learning rate; inversion depth is 0.5 on SD1.5 and 0.7 on SD3. GA makes 100 gradient-ascent steps in latent space with the listed learning rate. DPS, FreeDoM, and TFG use inversion depth 0.7 and the listed reward-guidance base coefficient ρ. DPS makes one guided reverse pass; FreeDoM uses two recurrence steps; TFG uses one recurrence with four inner gradient steps, the listed base coefficient μ, and a constant noise parameter $\sigma = 0.1$. DPS, FreeDoM, and TFG all use the increasing ρ schedule; DPS and FreeDoM set $\mu = 0$ and disable the optional guidance-noise schedule. TFG uses an increasing μ schedule, a constant-noise schedule with $\sigma = 0.1$, one noise draw per step, and unit gradient clipping scale. MPGD uses the listed clean-endpoint gradient step λ at every reverse step, with depth 0.7, constant guidance throughout the reverse interval, and no text classifier-free guidance. Its SD1.5 implementation uses the paper-style DDIM update; its SD3 implementation uses the flow adaptation described above. All baselines use 50 scheduler steps on SD1.5 or 28 on SD3, working resolutions of 512^2 or 768^2, and seed zero. All baselines use the same frozen source–condition pairs within each model and task.

Table 3: Task-specific baseline settings used for the reported results. The notation and shared settings are defined in the text.

Model	Task	ITOC (n, w, ℓ)	GA ℓ	DPS ρ	FreeDoM ρ	TFG (ρ, μ)	MPGD λ
SD3	Human	$(15, 500, 0.005)$	2	5	5	$(5, 1)$	1
SD3	Style	$(15, 1000, 0.005)$	3	50	20	$(40, 2.5)$	12
SD3	Counterfactual	$(15, 200, 0.005)$	1	7	3	$(7, 0.5)$	7
SD3	Face	$(15, 1000, 0.005)$	1.5	75	30	$(60, 2.5)$	10
SD1.5	Human	$(20, 500, 0.005)$	2	3	1	$(1, 0.5)$	1
SD1.5	Style	$(20, 200, 0.01)$	3	15	7.5	$(10, 1)$	12
SD1.5	Counterfactual	$(20, 50, 0.005)$	1	1	0.4	$(1, 0.1)$	7
SD1.5	Face	$(20, 1000, 0.005)$	1.5	40	20	$(30, 2.5)$	10

SD1.5 results. Table 4 reports the full SD1.5 comparison. FASTER leads the human-preference CLIPScore and Aesthetic validation columns, while its ImageReward is within 1.2% of the best GA result. It leads both the optimized CLIPScore and ImageReward in facial editing. Style transfer attains the lowest GramDiff but trails the best style validation scores, especially DINO. In counterfactual editing its CLIPScore is close to MPGD, yet its target logit remains lower.

Editing time and memory. Figure 7 extends Fig. 3 to all seven methods. All methods use the same timing scope: the GPU editing computation, including source inversion where required and all decoder and reward evaluations needed for optimization. We exclude model and image loading, input preprocessing and source encoding, final output decoding and saving, and offline metric evaluation. Timings are synchronized elapsed times for the editing routine, including its execution overhead rather than a sum of GPU-kernel durations. After excluding each worker's first image, we average the same three source images per task on one H20; speedups use unrounded means.

A separate 768×768 SD3 ImageReward edit with $K = 20$ on one RTX 4090 D takes 7.72 s and peaks at 18.75 GiB of live GPU tensor allocation. The text encoders are moved to CPU after prompt encoding, while the transformer, VAE, and ImageReward remain on the GPU. Peak memory is measured with PyTorch's peak allocated-memory counter, excluding allocator cache; time follows the editing-only scope above.

Table 4: Results on Stable Diffusion 1.5. Each task includes a target metric, which is directly optimized, and validation metrics used to assess potential reward hacking. Arrows indicate whether higher (↑) or lower (↓) values are better. Bold values indicate the best result for each metric.

Metric	ITOC	GA	DPS	FreeDoM	TFG	MPGD	**FASTER**
Human preference							
TARGET **ImageReward**↑	1.8981	**1.9532**	1.5374	1.3118	1.6908	1.1938	1.9316
VALIDATION HPSv2↑	**0.2666**	0.2615	0.2578	0.2555	0.2606	0.2535	0.2649
VALIDATION CLIPScore↑	0.2773	0.2743	0.2546	0.2366	0.2642	0.2531	**0.2798**
VALIDATION Aesthetic↑	5.8058	5.5443	5.4939	5.4869	5.5269	5.2346	**5.8075**
Style transfer							
TARGET **GramDiff**↓	9.0027	7.7940	5.8574	5.4821	6.0595	6.9080	**5.0592**
VALIDATION CLIP-I_{sty}↑	0.4944	0.5045	**0.5282**	0.5280	0.5252	0.5233	0.5190
VALIDATION $DINO_{sty}$↑	0.1585	0.1491	0.2471	**0.2645**	0.2460	0.1875	0.1895
Counterfactual generation							
TARGET **Logit**$_{tgt}$↑	19.7998	27.5803	17.0912	14.9241	18.9315	**31.7846**	28.8259
VALIDATION CLIPScore↑	0.1803	0.2017	0.1800	0.1702	0.1842	**0.2190**	0.2184
Text-guided facial editing							
TARGET **CLIPScore**↑	0.2886	0.2544	0.2763	0.2716	0.2831	0.2554	**0.3266**
VALIDATION ImageReward↑	-0.6209	-0.7547	-0.6610	-0.5575	-0.5089	-0.7707	**-0.4492**
VALIDATION HPSv2↑	0.2559	**0.2565**	0.2511	0.2536	0.2521	0.2509	0.2542

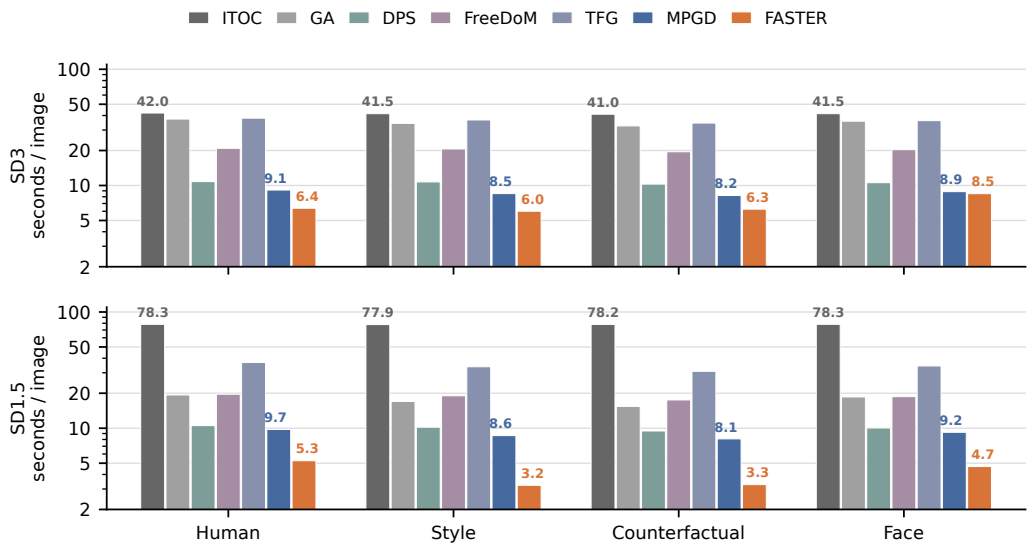

Figure 7: Editing time on one H20 across four tasks and two model families, averaged over three images per setting. The data and timing scope match Fig. 3; the vertical axis is logarithmic.

Additional visual results. Figures 8–11 show further counterfactual, facial, human-preference, and style-transfer edits, respectively.

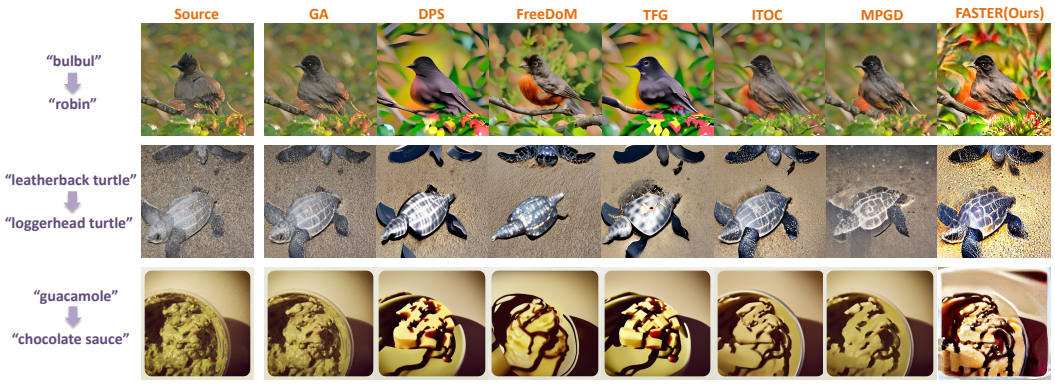

Figure 8: Additional matched counterfactual edits. Source and target classes are shown at left; method columns follow Fig. 4.

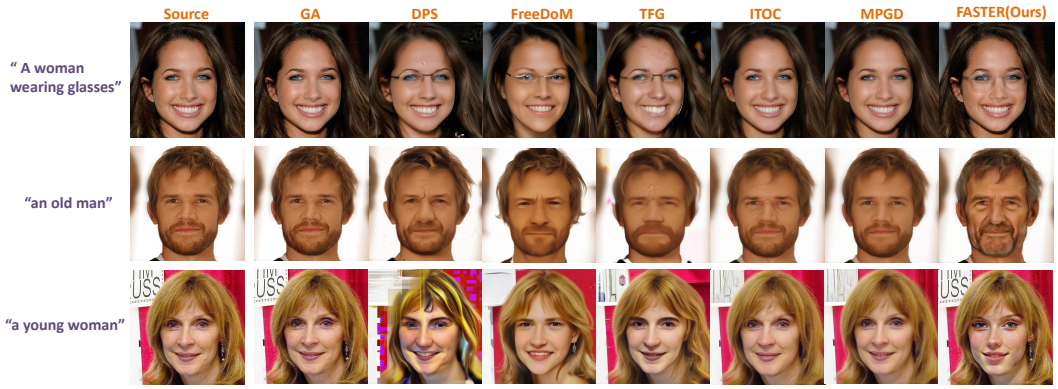

Figure 9: Additional matched facial edits. The requested text prompts are shown at left; method columns follow Fig. 4.

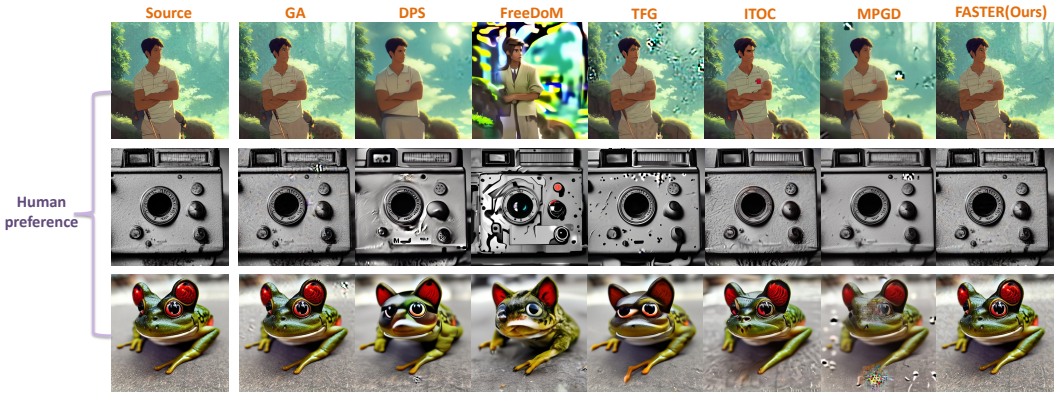

Figure 10: Additional matched human-preference examples, with the source image and outputs from each method.

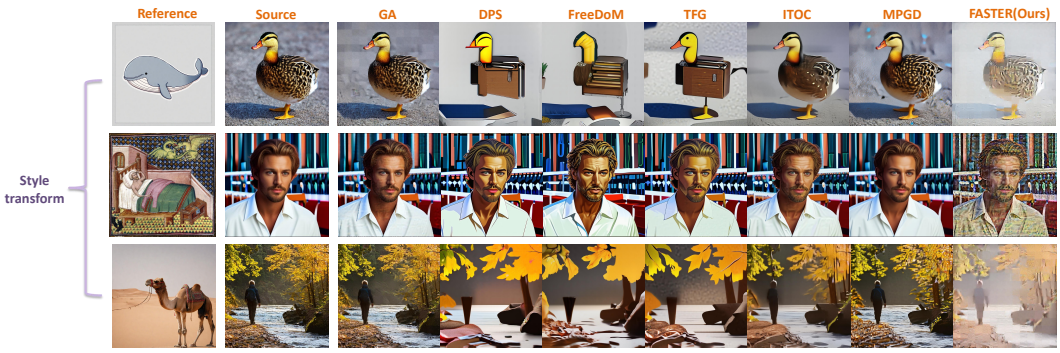

Figure 11: Additional style-transfer examples. The style reference and source image occupy the first two columns; method columns follow Fig. 4.

G Editing Term: Conditional-Score Difference

From velocity to an endpoint edit field. Use the source and target conditions c_{src} and c_{tgt} from Sec. 4.3. For the SD3 flow convention, a clean latent Y and Gaussian noise ϵ produce $x_t = (1-t)Y + t\epsilon$, and the exact conditional velocity is $v_c(x_t, t) = \mathbb{E}[\epsilon - Y \mid x_t, c]$. Rearranging the interpolation gives $\epsilon = x_t + (1-t)(\epsilon - Y)$. The Gaussian-mixture score identity therefore yields

$$s_c(x_t, t) := \nabla_{x_t} \log p_t(x_t \mid c) = -\frac{\mathbb{E}[\epsilon \mid x_t, c]}{t} = -\frac{x_t + (1-t)v_c(x_t, t)}{t}. \tag{40}$$

For $t \sim \nu$ in a fixed interval $0 < t_{\min} < t_{\max} < 1$ and independent Gaussian noise, we use the target-to-source conditional log-density ratio as a smoothed edit potential at endpoint y:

$$E_{\text{edit}}(y) = \mathbb{E}_{t,\epsilon}[\log p_t((1-t)y + t\epsilon \mid c_{\text{tgt}}) - \log p_t((1-t)y + t\epsilon \mid c_{\text{src}})]\,. \tag{41}$$

Assume the conditional densities are positive and differentiable, with an integrable gradient bound permitting differentiation under the expectation. For fixed t and ϵ, the Jacobian of x_t with respect to y is $(1-t)I$. Consequently, the pullback of the *target-minus-source score* is

$$\nabla_y E_{\text{edit}}(y) = \mathbb{E}_{t,\epsilon}\big[(1-t)(s_{c_{\text{tgt}}} - s_{c_{\text{src}}})(x_t, t)\big] \tag{42a}$$

$$= \mathbb{E}_{t,\epsilon}\left[\frac{(1-t)^2}{t}(v_{c_{\text{src}}} - v_{c_{\text{tgt}}})(x_t, t)\right]. \tag{42b}$$

This proves Eq. (15). The velocity order reverses because Eq. (40) has a minus sign. This is analogous to a conditional-guidance residual, with the source prompt rather than an unconditional prompt as reference. The implementation estimates Eq. (42) using paired source/target velocity evaluations at the same x_t and antithetic noise, then adds $c_e \nabla_y E_{\text{edit}}$ to the FASTER endpoint label. The pullback itself is linear and does not require backpropagation through the SD3 transformer. Equation (42) is exact for exact conditional scores; learned velocities yield an approximate edit field in practice.

Noise-window and strength sweep. Under this convention t increases with noise. Figure 12 uses one puppy source, one target prompt, a fixed checkpoint and two edit coefficients. Low-noise windows tend to produce local artifacts without the requested object; $[0.75, 0.90]$ gives a clear sunglasses shape. The trend is not monotone: $[0.85, 0.95]$ is weaker at $c_e = 100$.

For this visualization, the source prompt describes a small brown-and-white puppy on a plain white background, and the target adds realistic black sunglasses. Each SD3 cell runs to $K = 120$ with $M = 2$, 16 controller rollout steps, $\tau = 0.00125$, $\alpha\tau^2 = 1$, and $c_r = c_p = 0$. The figure uses the saved $K = 40$ checkpoint from each cell. The rows set $c_e \in \{100, 250\}$; the columns use the six noise-time windows printed in the figure.

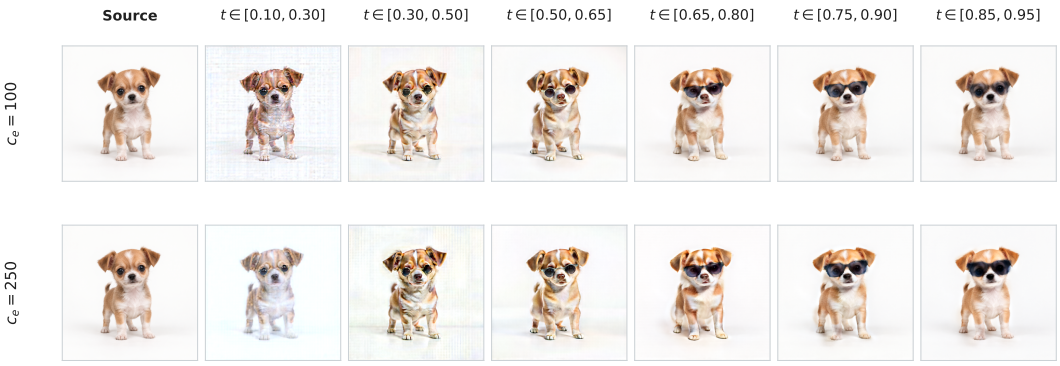

Figure 12: Adding sunglasses to the same puppy with the Editing Term. Columns vary the noise-time interval and rows vary c_e; the source is repeated for comparison.

Figure 13 follows one of the same runs rather than restarting at each budget. The sunglasses form early and become more distinct as the controller receives further endpoint updates.

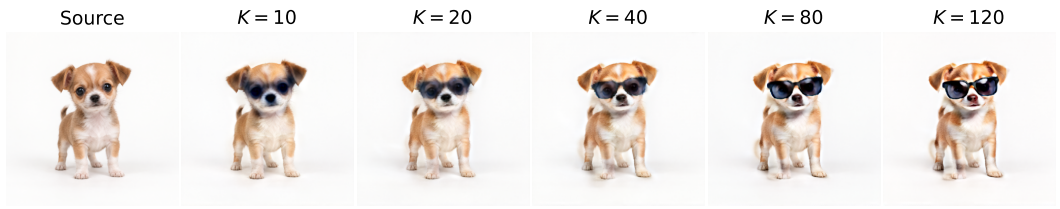

Figure 13: Saved outer-loop checkpoints for the puppy sunglasses edit at $c_e = 250$ and $t \in [0.75, 0.90]$.

Additional semantic edits. Figure 14 uses two visibly effective facial-attribute cases from the completed SD3 Editing Term sweep. They show saved $K = 20$ checkpoints from runs with $K = 40$, $M = 2$, 16 controller rollout steps, $\tau = 0.0125$, $\alpha\tau^2 = 1$, $c_r = c_p = 0$, and seed zero. The four output columns vary $c_e \in \{20, 50, 100, 200\}$; the two rows in each panel show noise-time intervals $[0.55, 0.70]$ and $[0.75, 0.90]$. The source and target prompts are "A man without lipstick" and "A man wearing lipstick" for the first case, and "A young man" and "An old man" for the second. These visual sweeps use the Editing Term without CLIPScore reward optimization and are separate from the four-task quantitative benchmark.

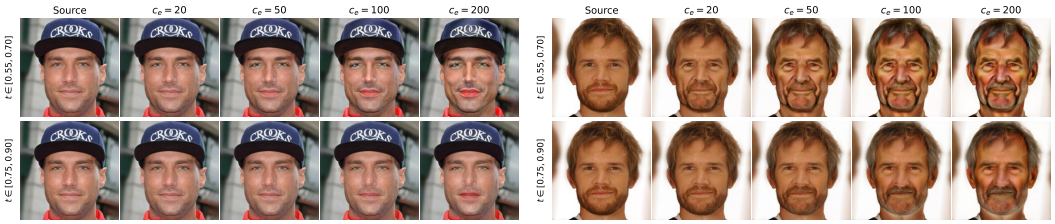

Figure 14: Editing Term strength and noise-time sweeps for lipstick (left) and apparent age (right). The source is repeated at left of each row.

H ENDPOINT FEEDBACK REUSE

We vary the number of controller updates per feedback refresh, $M \in \{1, 2, 4, 8, 16\}$, and inspect checkpoints at $K = 10, 20, \ldots, 100$ on four SD3 tasks. Each setting follows one continuous run on the same three source images per task, with the task-specific reward and preservation coefficients fixed. The cache holds only the latest endpoint–target pair. Thus, at fixed K, increasing M reuses the same number of expensive endpoint targets for more small-network updates (KM in total); this comparison varies supervision reuse rather than cache capacity or equal-time performance.

Figure 15 reports target and validation metrics. Moderate reuse improves the target at some query budgets, but increasing M is not uniformly beneficial: at later checkpoints, $M = 2$ gives stronger ImageReward and face-editing targets than the largest reuse settings, while style transfer favors fewer updates. Counterfactual target logits benefit from more reuse, illustrating that the preferred setting depends on the task. These curves characterize the selected examples rather than the full benchmark.

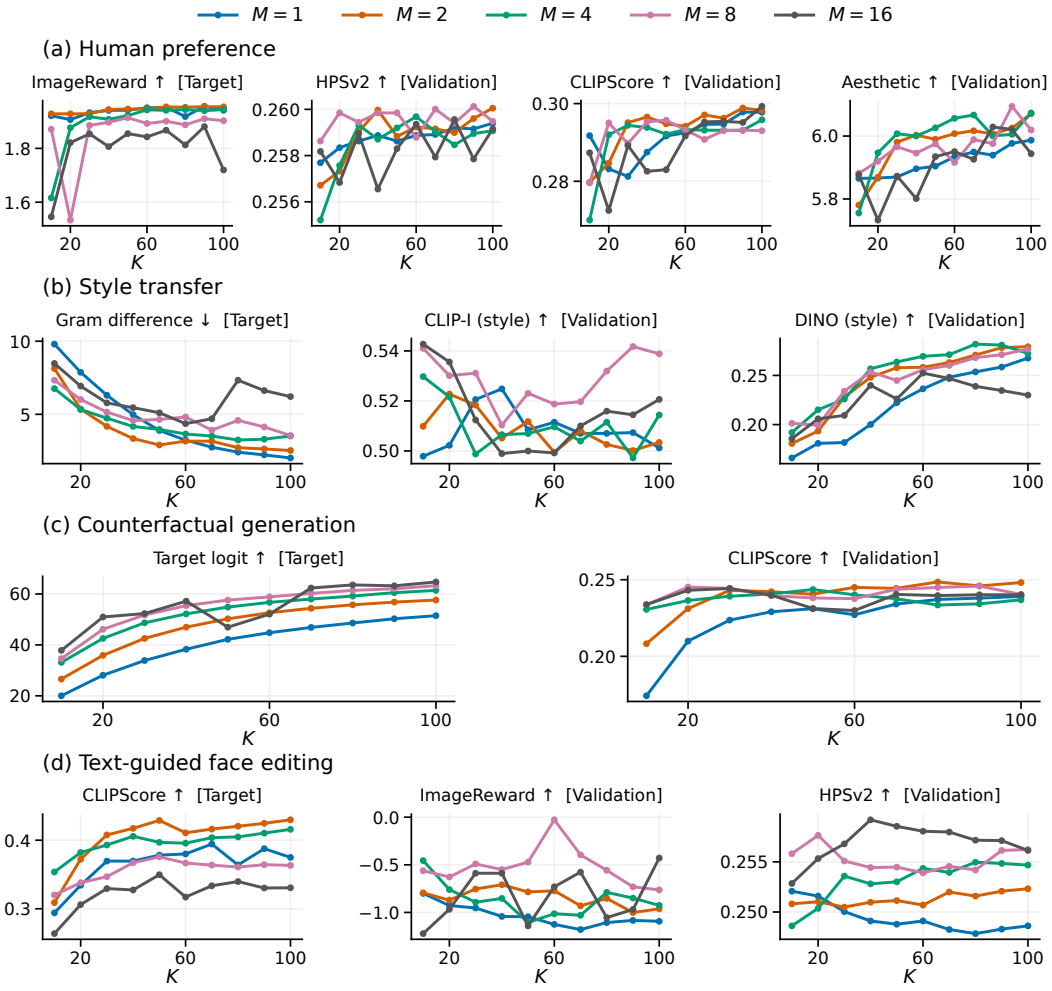

Figure 15: Endpoint feedback reuse across four SD3 tasks. Curves show three-image means as the number of outer refreshes K increases; colors denote the number of controller updates M per refresh. The first panel in each row is the optimized target, and the remaining panels are validation metrics.