ROOT: RESOLVING COEFFICIENT AMBIGUITY IN ENDPOINT REWARD TUNING FOR FLOW MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Forward-process reward tuning uses rewards on behavior-generated endpoints to define a reward-improved endpoint distribution, then re-noises these endpoints to learn its induced conditional flow. We identify ***coefficient ambiguity*** in stochastic velocity training, where positive weights defining the same endpoint distribution and induced flow can produce different updates. Condition-wise reward centering exposes this ambiguity by rescaling exponential weights without changing the target at a fixed tilt. We show that the target and behavior distributions uniquely determine a mean-one target-to-behavior density ratio. `ROOT` uses this probability-defined ratio as the coefficient for stochastic velocity learning, making its construction invariant to condition-wise rescaling of endpoint-equivalent weights. Its single-pass component, `RootSolve`, yields a behavior-centered finite-tilt objective whose conditional population optimum matches the target flow under behavior-flow consistency. Its multi-pass component, `RootTrace`, reuses scored rollouts through moving-reference updates that retain reward-dependent corrections on later passes while removing the pull toward the original behavior prediction. On Stable Diffusion 3.5 Medium, `ROOT` reaches matched Diffusion-NFT training-reward thresholds with 2.2–$3.4\times$ and 10–$20\times$ fewer outer iterations across single- and multi-pass settings, respectively, and achieves comparable final performance under sequential multi-reward training with about 41% fewer iterations.

1 INTRODUCTION

Reward post-training adapts diffusion and flow models to preferences, compositional constraints, and other objectives defined on generated samples (Black et al., 2024; Fan et al., 2023; Wallace et al., 2024; Liu et al., 2025). Recent forward-process approaches use a behavior model to generate multiple clean endpoints x_0 under condition c, a black-box evaluator to assign rewards $R(x_0, c)$, and fresh noising to construct velocity-training samples (Zheng et al., 2026; Potaptchik et al., 2026; Fan et al., 2025). By resampling noise and time, they use endpoint feedback across multiple stochastic velocity-training instances.

However, the reward-improved endpoint distribution determines only normalized endpoint probabilities, leaving freedom in the coefficients used by stochastic velocity training. We refer to this as the ***coefficient ambiguity problem***. For example, condition-wise advantage centering gives $A(x_0, c) = R(x_0, c) - \zeta(c)$, where $\zeta(c)$ is shared by endpoints under the same condition. At fixed tilt τ, Kullback–Leibler (KL)-regularized reward improvement gives exponential weights $e^{\tau A} = e^{-\tau\zeta(c)}e^{\tau R} = a(c)e^{\tau R}$ (Peters et al., 2010; Abdolmaleki et al., 2018; Peng et al., 2019). The condition-wise factor cancels after normalization, so R and A define the same endpoint target and induced flow. Their stochastic velocity updates can nevertheless differ, even after aligning their conditional mean-field scales. Training therefore depends on a coefficient representation that carries no additional information about the target.

Recent forward-process methods connect reward coefficients to velocity learning in different ways. DiffusionNFT maps group-centered rewards to $[0, 1]$ optimality probabilities used in reward-dependent velocity branches (Zheng et al., 2026). Tilt Matching derives velocity updates from exponential reward tilting (Potaptchik et al., 2026). These constructions specify how coefficients

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.

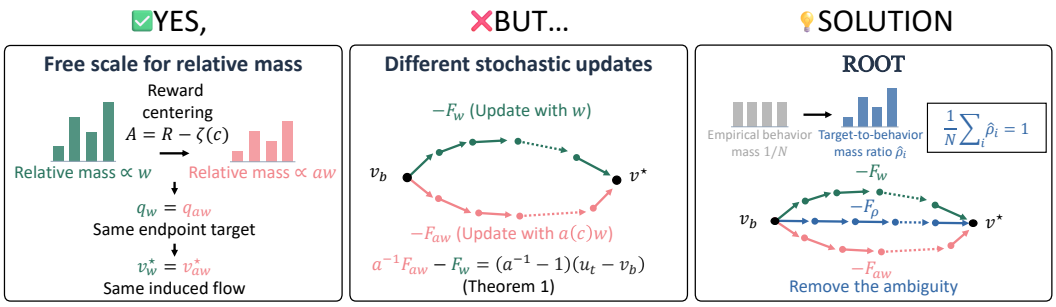


Figure 1: `ROOT` resolves coefficient ambiguity induced by endpoint-equivalent weights. A condition-wise scale preserves the endpoint target and induced flow but changes the calibrated stochastic field. `ROOT` removes this freedom using the mean-one target-to-behavior ratio.

enter velocity learning. We therefore ask: *How is the normalized endpoint target used to determine the training coefficient without an arbitrary condition-wise scale?*

Our key insight is to measure the target distribution relative to the behavior distribution that generated the rollout. For a realized group of N sampled endpoints, each endpoint is assigned empirical behavior mass $1/N$. Dividing its normalized target mass by this behavior mass gives the training coefficient directly. The population counterpart is the unique mean-one target-to-behavior density ratio. `ROOT` uses this ratio so that endpoint-equivalent weights produce the same stochastic field. Figure 1 illustrates how this probability reference resolves coefficient ambiguity.

`RootSolve` turns this probability-defined ratio into a behavior-centered velocity-training objective using re-noised behavior endpoints. Under behavior-flow consistency, we show that its conditional population optimum matches the flow induced by the reward-improved endpoint distribution. When the target equals the behavior distribution, the mean-one ratio further cancels reward-driven stochastic forcing at the behavior solution samplewise. To reuse each scored rollout across multiple optimizer updates, we further design `RootTrace`. Repeated `RootSolve` retains a term that pulls predictions toward the original behavior model, even when an endpoint's probability mass is unchanged. `RootTrace` uses `RootSolve` on the first pass and moves the velocity reference to the current prediction on later passes, retaining the reward-dependent correction without additional reward evaluations.

Our contributions are summarized as follows:

- **Coefficient ambiguity and canonical endpoint ratios.** We show that the endpoint-equivalent positive weights can induce different finite-tilt stochastic fields even after mean-field scale alignment. The target-to-behavior density ratio selects the unique mean-one representative of the normalized endpoint target, making the resulting construction invariant to condition-wise rescaling of endpoint-equivalent weights.
- **`RootSolve` for single-pass training.** We realize the identified density ratio in a behavior-sampled finite-tilt objective and prove that its unique conditional population optimum is the flow induced by the target endpoint distribution under behavior-flow consistency.
- **`RootTrace` for multi-pass training.** We develop moving-reference updates that reuse scored endpoints while retaining reward-dependent corrections at the current prediction. The construction removes the behavior-restoring contribution on reuse passes and recovers `RootSolve` exactly in the single-pass case.

2 Preliminaries

2.1 Conditional Flow Matching

For a condition c, we denote the endpoint density of a frozen behavior model by $p_b(\cdot \mid c)$. Conditional flow matching learns a velocity through regression on endpoint–noise pairs (Lipman et al., 2023; Liu et al., 2023; Albergo & Vanden-Eijnden, 2023; Albergo et al., 2025). With $x_0 \sim p_b(\cdot \mid c)$,

we use

$$x_t = (1 - \sigma_t)x_0 + \sigma_t \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I_d), \qquad u_t = \epsilon - x_0. \tag{1}$$

Here $\sigma_t \in [0, 1]$ is the noise level at model time t. Velocities are parameterized with respect to σ_t. We write $s = (x_t, t, c)$. Together, the behavior endpoint law and the forward kernel induce the behavior joint law of (x_0, s). Here $\mathbb{E}_b[\cdot \mid s]$ denotes conditional expectation under this joint law.

Our conditional analysis uses the following assumption.

Assumption 1 (Behavior-flow consistency). For almost every s, the frozen behavior prediction satisfies

$$v_b(s) = \mathbb{E}_b[u_t \mid s]. \tag{2}$$

This is the population regression identity satisfied by an exact flow-matching minimizer for the behavior-induced joint law. A finite learned model and finite-step sampler can introduce approximation error, so we use it as an idealized consistency assumption.

2.2 Endpoint Rewards and Forward-Training Interface

Reward tuning converts raw reward $R(x_0, c)$ or advantage $A(x_0, c)$ into positive endpoint weights $w(x_0, c)$. Generally, the weights define a reward-improved endpoint distribution by reweighting the behavior distribution,

$$q_w(x_0 \mid c) = \frac{w(x_0, c)}{\mathbb{E}_b[w \mid c]} p_b(x_0 \mid c). \tag{3}$$

KL-regularized reward improvement is a common formulation for reward tuning (Peters et al., 2010; Abdolmaleki et al., 2018; Peng et al., 2019), which gives exponential endpoint weights $w_R = e^{\tau R}$ or $w_A = e^{\tau A}$ at fixed tilt τ. Appendix A.1 gives the objective and proves this solution. Thus, reward information determines how probability mass is redistributed among behavior endpoints, and only the relative weights within each condition determine this target distribution. In particular, for any $a(c) > 0$, $w'(x_0, c) = a(c)w(x_0, c)$ gives $q_{w'} = q_w$.

The endpoint distribution in Eq. (3) induces a corresponding flow through the same forward noising process used in flow matching. With Eq. (1) and noisy state $s = (x_t, t, c)$, the conditional population velocity induced by q_w is

$$v_w^\star(s) = \frac{\mathbb{E}_b[w u_t \mid s]}{Z_w(s)}, \qquad Z_w(s) = \mathbb{E}_b[w \mid s] > 0. \tag{4}$$

Appendix A.2 derives this conditional change of measure and states the integrability conditions. The same condition-wise scaling $w \mapsto a(c)w$ also cancels in Eq. (4). Hence, the target identifies the class $[w] = \{a(c)w : 0 < a(c) < \infty\}$ up to behavior-null sets, leaving the common scale free.

Condition-wise advantage centering provides a simple instance of this representation freedom. For $A(x_0, c) = R(x_0, c) - \zeta(c)$, $w_A = e^{-\tau\zeta(c)} w_R$, so the coefficient representation changes while the normalized endpoint target and induced flow remain unchanged. [1]

3 ROOT: Endpoint-Ratio Reward Tuning

ROOT separates the stochastic realization of a supplied coefficient from the identification of that coefficient. We first fix a calibrated realization, then resolve the remaining coefficient ambiguity using the target-to-behavior endpoint ratio, and finally extend the construction to multi-pass training.

3.1 From Target Flow to a Calibrated Stochastic Field

From population target to stochastic realization. Section 2.2 specifies the conditional population velocity $v_w^\star(s)$ induced by a positive endpoint weight w. Write $v = v_\theta(s)$. We denote the sample loss gradient in model-output space by $F(v) = \nabla_v \ell(v)$ and call it the stochastic field. By the chain rule, its contribution to the parameter gradient is $(\partial v_\theta(s)/\partial\theta)^\top F(v)$. To construct this field

[1] More generally, affine preprocessing $A(x_0, c) = \frac{R(x_0,c) - \zeta(c)}{d(c)}$ changes the effective tilt when $d(c) \neq 1$; Appendix A.3 gives the corresponding distinction between reward translation and scaling.

from behavior samples, Eq. (4) provides the weighted residual that fixes the desired conditional root $\mathbb{E}_b[w(v-u_t) \mid s] = Z_w(s)\big(v - v_w^\star(s)\big)$, while Assumption 1 provides the zero-mean behavior residual $\mathbb{E}_b[u_t - v_b \mid s] = 0$. Adding a multiple of this residual can change the samplewise field without changing its conditional root. We therefore consider the linear behavior-controlled family

$$\widetilde{F}_{w;\alpha,\beta}(v) = \alpha(s)w(v-u_t) + \beta(s)(u_t - v_b), \qquad \alpha(s) > 0,$$

where $\alpha(s)$ and $\beta(s)$ are fixed after conditioning on s. Its conditional mean is

$$\mathbb{E}_b[\widetilde{F}_{w;\alpha,\beta}(v) \mid s] = \alpha(s)Z_w(s)\big(v - v_w^\star(s)\big).$$

This family preserves the same conditional root across $\beta(s)$ and positive $\alpha(s)$, exposing an overall field scale and a zero-mean residual coefficient. Appendix B.1 gives the full moment derivation.

Identity calibration. For unit endpoint weights $w \equiv 1$, the target equals behavior and $v_w^\star = v_b$. We require the sample gradient to vanish at $v = v_b$ for every forward-noising draw. This gives $\beta(s) = \alpha(s)$ whenever $u_t - v_b$ is conditionally nondegenerate. Treating the remaining common state-wise scale as external weighting and setting $\alpha(s) = 1$ gives

$$F_w(v) = w(v-u_t) + (u_t - v_b) = w(v - v_b) - (w-1)(u_t - v_b). \tag{5}$$

Its conditional mean is

$$\mathbb{E}_b[F_w(v) \mid s] = Z_w(s)\big(v - v_w^\star(s)\big), \tag{6}$$

so $v_w^\star(s)$ is the unique conditional population root whenever $Z_w(s) > 0$.

This calibration fixes how a supplied coefficient enters the stochastic field. Section 3.2 then identifies the coefficient associated with a prescribed normalized endpoint target. Appendix B.2 gives the calibration details, Appendix B.3 gives the control-variate interpretation, and Appendix B.4 relates the calibrated field to local tilt dynamics. A unified comparison with other forward-training realizations is provided in Appendix E.

3.2 RootSolve: Resolving Coefficient Ambiguity with Endpoint Ratios

Section 3.1 fixes a calibrated stochastic realization for a supplied positive endpoint coefficient w. Section 2.2 shows that a normalized endpoint target identifies the equivalence class $[w]$. We first show that this free scale changes the samplewise field, and then remove it by measuring target probability mass relative to behavior mass.

Theorem 1 (Coefficient ambiguity yields different stochastic fields). *Under Assumption 1 and finite conditional moments, w and $a(c)w$ define the same endpoint density and induced conditional flow, and their conditional mean fields share the same unique root $v_w^\star$. Writing $a = a(c)$, their scale-aligned sample fields satisfy*

$$a^{-1}F_{aw}(v) - F_w(v) = (a^{-1} - 1)(u_t - v_b), \tag{7}$$

which is nonzero when $a \neq 1$ and $u_t \neq v_b$.

Appendix C.1 gives the full conditions and proof. The factor a^{-1} aligns the conditional mean-field scale of F_{aw} with that of F_w, while Eq. (7) shows that a samplewise difference remains. The scale $a(c)$ carries no information about relative endpoint probabilities, yet remains in the samplewise field after this alignment. Thus, the normalized endpoint target fixes the desired conditional flow but still leaves the stochastic coefficient unidentified. To remove this freedom, we use the behavior distribution as the reference with respect to which target probability mass is measured.

Target-to-behavior ratio. For a realized behavior rollout, the empirical behavior measure provides this reference directly. For a realized condition c, let $\{x_0^{(i)}\}_{i=1}^N$ be the behavior-generated endpoints and define the empirical behavior measure $\widehat{P}_b(\cdot \mid c) = \frac{1}{N}\sum_{i=1}^N \delta_{x_0^{(i)}}(\cdot)$, where δ_x denotes the Dirac point mass at x. Conditional on the realized scores $A_i = R_i - \zeta(c)$, KL-regularized reward improvement assigns the empirical target masses $\widehat{Q}^\star(\{x_0^{(i)}\} \mid c) = \frac{e^{\tau A_i}}{\sum_{j=1}^N e^{\tau A_j}}$. The corresponding empirical target-to-behavior mass ratio is

$$\widehat{\rho}_i = \frac{\widehat{Q}^\star(\{x_0^{(i)}\} \mid c)}{\widehat{P}_b(\{x_0^{(i)}\} \mid c)} = \frac{e^{\tau R_i}}{N^{-1}\sum_j e^{\tau R_j}} = \frac{e^{\tau A_i}}{N^{-1}\sum_j e^{\tau A_j}}, \qquad \frac{1}{N}\sum_i \widehat{\rho}_i = 1. \tag{8}$$

Condition-shared centering cancels from the ratio exactly. Appendix C.2 gives the finite-simplex KL derivation and the nonuniform-mass extension.

The same construction has a direct population counterpart.

$$\rho(x_0, c) := \frac{q^\star(x_0 \mid c)}{p_b(x_0 \mid c)} = \frac{e^{\tau R(x_0,c)}}{\mathbb{E}_b[e^{\tau R} \mid c]} = \frac{e^{\tau A(x_0,c)}}{\mathbb{E}_b[e^{\tau A} \mid c]}, \qquad \mathbb{E}_b[\rho \mid c] = 1. \tag{9}$$

We use $\widehat{\rho}$ for the realized empirical mass ratio and ρ for its population density-ratio counterpart. The derivations below are written for ρ, while the implementation uses $\widehat{\rho}$ on each realized rollout group. For the empirical ratio, values above or below one indicate increased or decreased endpoint probability mass relative to behavior, while one leaves that mass unchanged. The mean-one property is more than a normalization convention because it uniquely selects one representative from the entire endpoint-equivalent class.

Proposition 1 (Canonical density-ratio representative). *For a fixed target density $q^\star$ supported on p_b with $\rho = q^\star/p_b > 0$ on the behavior support, ρ is the unique positive weight representing $q^\star$ whose behavior expectation is one, up to behavior-null sets.*

Proof is given in Appendix C.2. Proposition 1 selects $\rho = q^\star/p_b$ as the unique positive representative of the prescribed endpoint target with behavior mean one, removing the condition-wise scale freedom in Theorem 1. At the parameter-gradient level, Appendix C.2 further shows that any other endpoint-equivalent representative introduces a zero-mean perturbation whose conditional second moment scales with the squared deviation of its free scale from the canonical choice. Appendix C.3 distinguishes this population construction from its finite empirical realization.

Representation invariance. Proposition 1 also makes the stochastic construction independent of the unnormalized coefficient representation. If $w' = a(c)w$ induces the same endpoint target as w, then

$$\frac{w}{\mathbb{E}_b[w \mid c]} = \frac{w'}{\mathbb{E}_b[w' \mid c]} = \rho.$$

Equivalent weights therefore yield the same coefficient and the same `RootSolve` field. Within the behavior-controlled family of Section 3.1, the relative residual coefficient is fixed by identity calibration and the common field scale is set to one.

`RootSolve` stochastic objective. Section 3.1 fixes how a supplied coefficient enters the stochastic field, while Proposition 1 identifies that coefficient as ρ. Substituting the probability-defined coefficient $w = \rho$ into Eq. (5) gives

$$F_{\mathrm{solve}}(\rho; v) = \rho(v - u_t) + (u_t - v_b) = \rho(v - v_b) - (\rho - 1)(u_t - v_b). \tag{10}$$

The first form makes the calibrated residual construction explicit, while the second separates ratio-weighted displacement from behavior and the signed endpoint-ratio correction along $u_t - v_b$.

When the requested endpoint distribution equals behavior, $q^\star = p_b$, the identified ratio is $\rho = 1$. Equation (10) then reduces to

$$F_{\mathrm{solve}}(1; v) = v - v_b, \qquad F_{\mathrm{solve}}(1; v_b) = 0.$$

Thus the probability-defined identity coefficient produces no residual samplewise forcing at the behavior solution. Appendix C.5 analyzes this cancellation and the representation dependence of conditional noise.

Substituting $w = \rho$ into Eq. (4) gives

$$Z_\rho(s) = \mathbb{E}_b[\rho \mid s], \qquad v^\star_\rho(s) = \frac{\mathbb{E}_b[\rho u_t \mid s]}{Z_\rho(s)}. \tag{11}$$

This is the conditional flow induced by $q^\star$ through the same forward kernel. Endpoint normalization gives $\mathbb{E}_b[\rho \mid c] = 1$, while the conditional flow after forward noising depends on the state-dependent normalizer $Z_\rho(s)$.

Using Eq. (6) with $w = \rho$ yields

$$\mathbb{E}_b[F_{\mathrm{solve}}(\rho; v) \mid s] = Z_\rho(s)\big(v - v^\star_\rho(s)\big). \tag{12}$$

We use the following scalar objective whose output gradient equals the sample field:

$$\ell_{\mathrm{solve}}(\rho; v_\theta) = \frac{\rho}{2}\|v_\theta - v_b\|^2 - (\rho - 1)\langle v_\theta - v_b, u_t - v_b\rangle, \tag{13}$$

where $\nabla_v \ell_{\mathrm{solve}} = F_{\mathrm{solve}}$. The following theorem identifies its conditional population target.

Theorem 2 (`RootSolve` conditional exactness). *Suppose Assumption 1 holds, $\rho = q^\star/p_b$ is positive and normalized, and the conditional moments defining the risk are finite. For almost every s, $\mathbb{E}_b[\ell_{\mathrm{solve}}(\rho; v) \mid s]$ is strictly convex in v with unique minimizer $v^\star_\rho(s)$, the conditional flow induced by the target density $q^\star = \rho p_b$.*

Proof is given in Appendix C.4. Theorem 2 is a population statement. For a finite rollout group, $\widehat{\rho}_i$ exactly specifies the realized empirical endpoint-mass redistribution, while consistency between the frozen behavior prediction and that empirical law remains approximate. Appendix C.3 clarifies this distinction.

Structural properties. The construction above gives `RootSolve` four useful properties. (i) *Probability identification and representation invariance*: $\rho = q^\star/p_b$ is the unique positive representative with behavior mean one. Canonicalizing endpoint-equivalent weights therefore yields the same `RootSolve` field and removes the parameter-gradient perturbation associated with their free condition-wise scale (Appendix C.2). (ii) *Conditional finite-tilt exactness*: under Theorem 2's conditions, the base objective is conditionally minimized at $v^\star_\rho$, which is exactly the flow induced by the prescribed endpoint target. (iii) *Reward-translation invariance*: at fixed tilt, condition-wise centering $A = R - \zeta(c)$ leaves both ρ and F_{solve} unchanged (Appendix A.3). (iv) *Identity consistency*: $q^\star = p_b$ implies $\rho = 1$, $F_{\mathrm{solve}}(1; v) = v - v_b$, and $F_{\mathrm{solve}}(1; v_b) = 0$ samplewise.

3.3 RootTrace: Moving-Reference Rollout Reuse

Fixed behavior reference during reuse. Reusing scored endpoints keeps their reward feedback fixed while the live model changes. `RootSolve` combines two contributions,

$$F_{\mathrm{solve}}(\rho; v) = (v - v_b) + (\rho - 1)(v - u_t).$$

The behavior-restoring term $v - v_b$ contributes a descent direction toward the original behavior prediction in output space. The reward-dependent correction $(\rho - 1)(v - u_t)$ accounts for the increase or decrease in endpoint mass. It is exactly the difference between the fields with ratio ρ and ratio one,

$$F_{\mathrm{solve}}(\rho; v) - F_{\mathrm{solve}}(1; v) = (\rho - 1)(v - u_t). \tag{14}$$

Repeated `RootSolve` retains the behavior-restoring term after each update. Thus, an endpoint whose mass is unchanged can still contribute $v - v_b$ once the model has moved. We use the additional passes to form reward-dependent corrections at the current prediction. This motivates moving the velocity reference with the model while keeping the endpoint ratios fixed.

Moving-reference updates. Let θ_{k-1} denote the live parameters before pass k. We use the behavior prediction on the first pass and the current prediction on subsequent passes,

$$\bar{v}_{k-1}(s) = \begin{cases} v_b(s), & k = 1, \\ \mathrm{sg}[v_{\theta_{k-1}}(s)], & k \geq 2, \end{cases} \tag{15}$$

where sg denotes stop-gradient and holds the reference fixed during differentiation. The scored endpoints, behavior distribution, and ratio $\rho = q^\star/p_b$ remain fixed throughout the outer iteration. We replace the behavior-restoring term $v - v_b$ with the moving-reference term $v - \bar{v}_{k-1}$, while retaining the same reward-dependent correction $(\rho - 1)(v - u_t)$. The resulting field is

$$\begin{aligned} F^{(k)}_{\mathrm{trace}}(\rho; v, \bar{v}_{k-1}) =& (v - \bar{v}_{k-1}) + (\rho - 1)(v - u_t) \\ =& F_{\mathrm{solve}}(\rho; v) - F_{\mathrm{solve}}(1; \bar{v}_{k-1}) \\ =& \rho(v - \bar{v}_{k-1}) - (\rho - 1)(u_t - \bar{v}_{k-1}). \end{aligned} \tag{16}$$

The corresponding pass-specific scalar objective is

$$\ell^{(k)}_{\mathrm{trace}}(\rho; v, \bar{v}_{k-1}) = \frac{\rho}{2}\|v - \bar{v}_{k-1}\|^2 - (\rho - 1)\langle v - \bar{v}_{k-1}, u_t - \bar{v}_{k-1}\rangle. \tag{17}$$

Only the live prediction receives gradients, giving $\nabla_v \ell^{(k)}_{\mathrm{trace}}(\rho; v, \bar{v}_{k-1}) = F^{(k)}_{\mathrm{trace}}(\rho; v, \bar{v}_{k-1})$. Appendix D.1 derives its relation to the `RootSolve` objective.

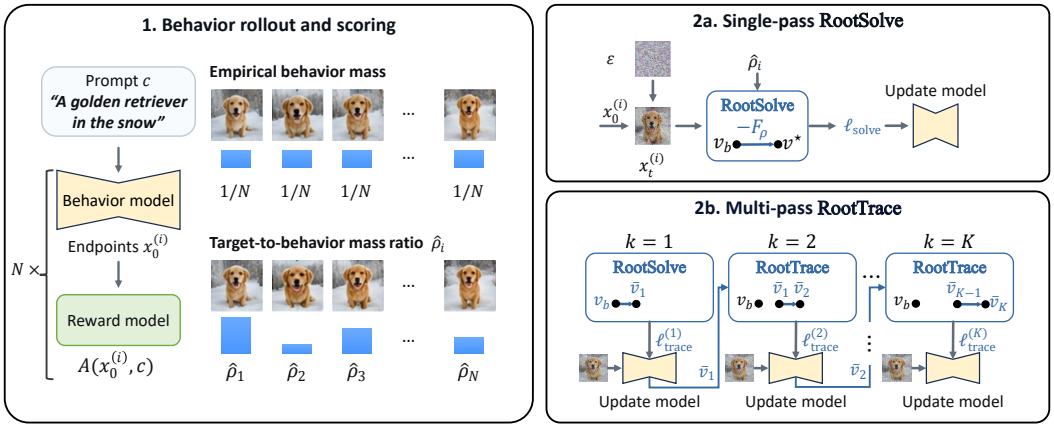


Figure 2: ROOT training workflow. Behavior-generated endpoints are scored once and converted to complete-group empirical mass ratios $\widehat{\rho}_i$. RootTrace uses RootSolve on the first pass and moving-reference updates on reuse passes, keeping the scored endpoints and ratios fixed while resampling forward time and noise.

Effect on reused updates. On passes $k \geq 2$, the reference is the detached live prediction, $\bar{v}_{k-1} = \mathrm{sg}[v_{\theta_{k-1}}(s)]$. Thus, at the point where the pass-k gradient is evaluated, v and $\bar{v}_{k-1}$ have the same numerical value, although only v carries gradients. Substituting $v = \bar{v}_{k-1}$ into Eq. (16) gives

$$F_{\text{trace}}^{(k)}(\rho; \bar{v}_{k-1}, \bar{v}_{k-1}) = (\rho - 1)(\bar{v}_{k-1} - u_t). \tag{18}$$

Hence, a reuse pass applies exactly the reward-dependent correction of RootSolve at the current prediction, with the behavior-restoring contribution removed. Equivalently, it realizes the direct reuse field $(\rho-1)(v-u_t)$ at the current prediction. Appendix D.2 gives the corresponding parameter-gradient relation and stop-gradient computation. Each pass reuses the fixed scored endpoints and ratios while resampling forward time and noise. The reference on reuse passes is obtained by detaching the live prediction from the same forward evaluation, requiring no additional model evaluation. On the first pass, $\bar{v}_0 = v_b$, so Eq. (17) reduces exactly to the RootSolve objective for every live prediction. Consequently, $K = 1$ recovers RootSolve exactly (Appendix D.3).

3.4 Training Procedure

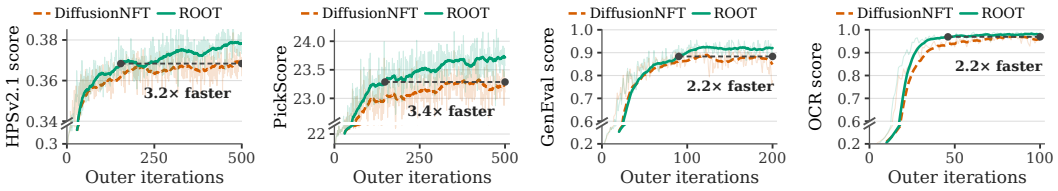
Figure 2 illustrates the training workflow of ROOT. During each outer iteration, we score a complete group of behavior-generated endpoints once and compute fixed complete-group ratios. RootSolve uses one forward-training pass. RootTrace reuses the same scored endpoints and ratios for K passes, resampling forward time and noise at each pass. Its first pass is exactly RootSolve, and later passes use the current predictions at freshly noised states as their velocity references. The behavior model remains frozen within the outer iteration and is soft-updated via Exponential Moving Average (EMA) afterward. Full pseudocode (Algorithm 1) and implementation details are provided in Appendix F.

4 Experiments

We evaluate ROOT across three dimensions: (1) head-to-head single-reward optimization (Section 4.2), (2) sequential multi-reward training (Section 4.3), and (3) further comparisons, ablations, and qualitative illustrations (Appendix F).

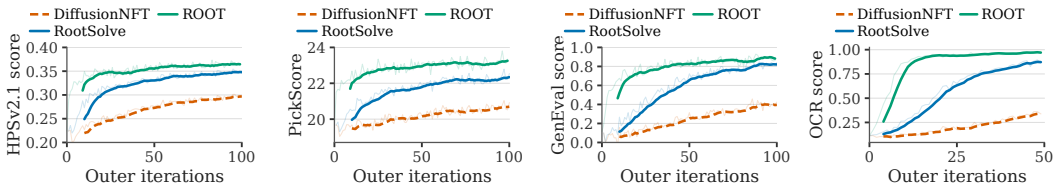
4.1 Experimental Setup

Backbone and comparison protocol. All experiments use Stable Diffusion 3.5 Medium (SD3.5-M) (Esser et al., 2024; Rombach et al., 2022) at 512×512 resolution. We fine-tune the model with LoRA (Hu et al., 2022) ($r = 32$, $\alpha = 64$) and use DiffusionNFT (Zheng et al., 2026) as the



(a) HPSv2.1 (b) PickScore (c) GenEval (d) OCR

Figure 3: ROOT versus DiffusionNFT on single-reward single-pass ($K = 1$) training.



(a) HPSv2.1 (b) PickScore (c) GenEval (d) OCR

Figure 4: ROOT versus DiffusionNFT on single-reward multi-pass ($K = 2$) training.

primary baseline. Unless otherwise stated, the backbone, rollout budget, optimization settings, and evaluation protocol are matched to those of DiffusionNFT for controlled comparison.

Tasks and rewards. We consider four single-reward tasks covering both model-based human preference and rule-based alignment. The former includes HPSv2.1 (Wu et al., 2023) and PickScore (Kirstain et al., 2023), while the latter includes GenEval (Ghosh et al., 2023) for compositional image–text alignment and OCR for text rendering. We use the standardized advantage value $A = \frac{R-\text{mean}(R)}{\text{std}(R)}$ for the endpoint score. Training and evaluation prompt sets follow DiffusionNFT.

Training and evaluation. Each rollout collects 48 prompts with a group size of $N = 24$ using 10 sampling steps, and final evaluation uses a 40-step first-order Ordinary Differential Equation (ODE) sampler. Single-pass experiments use $K = 1$, while multi-pass experiments reuse the same collected samples and reward evaluations across $K > 1$ passes, without additional rollouts or reward queries. For multi-reward experiments, we apply a 3-stage, 1000-outer training scheme. Additional optimization and task-specific details are provided in Appendix F.

4.2 Single-Reward Head-to-Head Training

We first compare ROOT with DiffusionNFT under single-reward single-pass optimization. Figure 3 reports the training trajectories on HPSv2.1, PickScore, GenEval, and OCR. Across the four objectives, ROOT reaches the final DiffusionNFT training-reward level (95th percentile of its reward) $3.2\times$, $3.4\times$, $2.2\times$, and $2.2\times$ faster, respectively, while maintaining a higher trailing-mean reward later in training.

Figure 4 further compares DiffusionNFT, repeated RootSolve, and the full ROOT construction under a matched $K = 2$ setting. DiffusionNFT and repeated RootSolve retain the behavior prediction as the reference across passes and show limited improvement from the second pass. With RootTrace, ROOT moves the reference to the current prediction after the first pass and reaches the DiffusionNFT reward level around 10–$20\times$ faster.

4.3 Multi-Reward Sequential Training

We further evaluate ROOT under sequential multi-reward training. Following the reward definitions of DiffusionNFT, we first optimize model-based rewards on Pick-a-Pic, including PickScore, CLIPScore (Hessel et al., 2021), and HPSv2.1, and then optimize the rule-based GenEval and OCR rewards. We use the same reward definitions and evaluation protocol as DiffusionNFT. Additional training details and qualitative comparison are provided in Appendix F.

Table 1: Sequential multi-reward evaluation. † denotes local evaluation of the official released DiffusionNFT multi-reward checkpoint. **Bold**: best; underline: second best.

Model	#Iter	Task Metric			DrawBench Metric					
		GenEval	OCR	PickScore	HPSv2.1	CLIPScore	Aesthetic	ImgRwd	PickScore	UniRwd
SD3.5-M (w/o CFG)	–	0.261	0.103	19.93	0.200	0.237	5.14	−0.646	20.59	1.900
+ CFG	–	0.627	0.555	21.78	0.280	0.284	5.38	0.870	22.40	2.959
+ DiffusionNFT†	1.7k	0.929	**0.890**	23.35	**0.330**	**0.294**	**6.00**	1.481	**23.81**	3.359
+ ROOT	1k	**0.930**	0.888	**23.36**	**0.330**	**0.294**	**6.00**	**1.487**	**23.81**	**3.372**

Table 1 shows that ROOT preserves strong performance under sequential multi-reward training. For DiffusionNFT, we locally evaluate its official released multi-reward checkpoint using the same evaluation pipeline as ROOT. At 1k training iterations, ROOT closely matches the 1.7k-iteration DiffusionNFT checkpoint across all evaluated metrics, with equal or higher scores on eight of the nine metrics at the reported precision. This achieves comparable multi-reward performance with approximately 41% fewer training iterations. Qualitative comparisons are provided in Appendix F.5.

5 Related Work

Forward-process reward tuning. Recent forward-process methods reuse reward-scored endpoints through fresh noising to construct velocity-training samples. DiffusionNFT transforms group-centered rewards into optimality probabilities and uses them to define reward-dependent velocity branches (Zheng et al., 2026). Tilt Matching derives explicit and implicit velocity updates from exponential reward tilting (Potaptchik et al., 2026). Reward-weighted conditional flow matching provides another route through weighted regression and regularization (Fan et al., 2025), and recent adjoint and path-space views study reward tuning through related gradient and transport formulations (Bergmeister et al., 2026; Xu et al., 2026). These methods differ in both the reward coefficient supplied to stochastic velocity learning and the relation between the resulting field and the finite-tilt target flow. Appendix E gives a unified comparison along these two axes. ROOT identifies its coefficient from the normalized endpoint target and realizes it with a finite-tilt exact field.

Reward improvement and generative-model alignment. KL-constrained policy-improvement methods connect reward optimization to normalized exponential reweighting or weighted regression (Peters et al., 2010; Abdolmaleki et al., 2018; Peng et al., 2019). For diffusion and flow models, trajectory-level reward optimization includes DDPO, DPOK, and Flow-GRPO (Black et al., 2024; Fan et al., 2023; Liu et al., 2025), with recent variants studying sparse rewards, structured branching, flow-specific updates, and optimization stability (Hu et al., 2025; Li et al., 2026; Zhou et al., 2026; He et al., 2026; Wang et al., 2026). A separate line propagates differentiable rewards through generation (Clark et al., 2024; Wu et al., 2024; Guo et al., 2025; Liang et al., 2026), while preference-based methods optimize pairwise or calibrated preference signals (Wallace et al., 2024; Yang et al., 2024; Lee et al., 2025; Croitoru et al., 2025; Liang et al., 2025). ROOT studies how a normalized endpoint redistribution determines the coefficients used in stochastic velocity training from reward-scored samples.

6 Conclusion

We present ROOT, an endpoint-ratio formulation for forward-process reward tuning of flow models. We show that endpoint-equivalent positive weights can induce different stochastic fields even when they define the same normalized endpoint target. The target-to-behavior density ratio selects the unique mean-one representative, making ROOT invariant to endpoint-equivalent coefficient rescaling. Combined with the identity-calibrated stochastic field, RootSolve has a conditional population optimum that matches the flow induced by the reward-improved endpoint distribution under behavior-flow consistency. RootTrace extends the formulation to rollout reuse through moving-reference updates that retain reward-dependent corrections at the current prediction. Experiments on SD3.5-M show faster single-reward optimization and comparable sequential multi-reward performance relative to DiffusionNFT. Overall, ROOT provides a probability-defined interface between endpoint reward redistribution and stochastic velocity learning.

AI USE STATEMENT

Generative AI tools were used only for language polishing. All technical content, experimental results, analyses, and conclusions were produced and verified by the authors.

ETHICS STATEMENT

Reward tuning may inherit biases and failure modes from reward models, and generative image models may be misused to produce misleading or harmful content.

REPRODUCIBILITY STATEMENT

We will release the code, training configurations, and evaluation scripts upon acceptance to facilitate reproduction of our results.

REFERENCES

Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018. URL https://openreview.net/forum?id=S1ANxQW0b.

Michael Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025. URL https://www.jmlr.org/papers/v26/23-1605.html.

Michael Samuel Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=li7qeBbCR1t.

Andreas Bergmeister, Stefanie Jegelka, Nikolas Nüsken, Carles Domingo-Enrich, and Jakiw Pidstrigach. Reinforce adjoint matching: Scaling RL post-training of diffusion and flow-matching models. arXiv preprint arXiv:2605.10759, 2026. URL https://arxiv.org/abs/2605.10759.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pp. 4965–4987, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/14f75513f0f1ca01de1e826b52e6b840-Abstract-Conference.html.

Kevin Clark, Paul Vicol, Kevin Swersky, and David Fleet. Directly fine-tuning diffusion models on differentiable rewards. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pp. 4793–4822, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/13e8be77982beb73d7ed0bbf122f9f3c-Abstract-Conference.html.

Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, Nicu Sebe, and Mubarak Shah. Curriculum direct preference optimization for diffusion and consistency models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2824–2834, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Croitoru_Curriculum_Direct_Preference_Optimization_for_Diffusion_and_Consistency_Models_CVPR_2025_paper.html.

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 International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12606–12633. PMLR, 2024. URL https://proceedings.mlr.press/v235/esser24a.html.

Jiajun Fan, Shuaike Shen, Chaoran Cheng, Yuxin Chen, Chumeng Liang, and Ge Liu. Online reward-weighted fine-tuning of flow matching with wasserstein regularization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pp. 60053–60113, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/9690d4746230cfea3d067fca695ba648-Abstract-Conference.html.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. DPOK: Reinforcement learning for fine-tuning text-to-image diffusion models. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 79858–79885, 2023. doi: 10.52202/075280-3497.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. GenEval: An object-focused framework for evaluating text-to-image alignment. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 52132–52152, 2023. doi: 10.52202/075280-2270.

Xiefan Guo, Miaomiao Cui, Liefeng Bo, and Di Huang. ShortFT: Diffusion model alignment via shortcut-based fine-tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 678–687, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Guo_ShortFT_Diffusion_Model_Alignment_via_Shortcut-based_Fine-Tuning_ICCV_2025_paper.html.

Dailan He, Guanlin Feng, Xingtong Ge, Yazhe Niu, Yi Zhang, Bingqi Ma, Guanglu Song, Yu Liu, and Hongsheng Li. Neighbor GRPO: Contrastive ODE policy optimization aligns flow models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6033–6042, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/He_Neighbor_GRPO_Contrastive_ODE_Policy_Optimization_Aligns_Flow_Models_CVPR_2026_paper.html.

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 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7514–7528. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.595. URL https://aclanthology.org/2021.emnlp-main.595/.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Zijing Hu, Fengda Zhang, Long Chen, Kun Kuang, Jiahui Li, Kaifeng Gao, Jun Xiao, Xin Wang, and Wenwu Zhu. Towards better alignment: Training diffusion models with reinforcement learning against sparse rewards. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23604–23614, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Hu_Towards_Better_Alignment_Training_Diffusion_Models_with_Reinforcement_Learning_Against_CVPR_2025_paper.html.

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 *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 36652–36663, 2023. doi: 10.52202/075280-1594.

Kyungmin Lee, Xiahong Li, Qifei Wang, Junfeng He, Junjie Ke, Ming-Hsuan Yang, Irfan Essa, Jinwoo Shin, Feng Yang, and Yinxiao Li. Calibrated multi-preference optimization for aligning diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18465–18475, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Lee_Calibrated_Multi-Preference_Optimization_for_Aligning_Diffusion_Models_CVPR_2025_paper.html.

Yuming Li, Yikai Wang, Yuying Zhu, Zhongyu Zhao, Ming Lu, Qi She, and Shanghang Zhang. BranchGRPO: Stable and efficient GRPO with structured branching in diffusion models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/233d16f17f809981763db2f01b7f9603-Abstract-Conference.html.

Zhanhao Liang, Yuhui Yuan, Shuyang Gu, Bohan Chen, Tiankai Hang, Mingxi Cheng, Ji Li, and Liang Zheng. Aesthetic post-training diffusion models from generic preferences with step-by-step preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13199–13208, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Liang_Aesthetic_Post-Training_Diffusion_Models_from_Generic_Preferences_with_Step-by-step_Preference_CVPR_2025_paper.html.

Zhanhao Liang, Tao Yang, Jie Wu, Chengjian Feng, and Liang Zheng. LeapAlign: Post-training flow matching models at any generation step by building two-step trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/papers/Liang_LeapAlign_Post-training_Flow_Matching_Models_at_Any_Generation_Step_by_CVPR_2026_paper.pdf.

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

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-GRPO: Training flow matching models via online RL. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 38, pp. 40783–40818, 2025. doi: 10.52202/085713-1362.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. arXiv preprint arXiv:1910.00177, 2019. URL https://arxiv.org/abs/1910.00177.

Jan Peters, Katharina Mulling, and Yasemin Altun. Relative entropy policy search. *Proceedings of the AAAI Conference on Artificial Intelligence*, 24(1):1607–1612, 2010. doi: 10.1609/aaai.v24i1.7727.

Peter Potaptchik, Cheuk-Kit Lee, and Michael S. Albergo. Tilt matching for scalable sampling and fine-tuning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2026. URL https://icml.cc/virtual/2026/poster/62780.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Rombach_High-Resolution_Image_Synthesis_With_Latent_Diffusion_Models_CVPR_2022_paper.html.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 36479–36494, 2022. doi: 10.52202/068431-2643.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model

alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8228–8238, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Wallace_Diffusion_Model_Alignment_Using_Direct_Preference_Optimization_CVPR_2024_paper.html.

Jing Wang, Jiajun Liang, Jie Liu, Henglin Liu, Gongye Liu, Jun Zheng, Wanyuan Pang, Ao Ma, Zhenyu Xie, Xintao Wang, Meng Wang, Pengfei Wan, and Xiaodan Liang. GRPO-Guard: Mitigating implicit over-optimization in flow matching via regulated clipping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5988–5998, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Wang_GRPO-Guard_Mitigating_Implicit_Over-Optimization_in_Flow_Matching_via_Regulated_Clipping_CVPR_2026_paper.html.

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.

Xiaoshi Wu, Yiming Hao, Manyuan Zhang, Keqiang Sun, Zhaoyang Huang, Guanglu Song, Yu Liu, and Hongsheng Li. Deep reward supervisions for tuning text-to-image diffusion models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 108–124. Springer, 2024. doi: 10.1007/978-3-031-73010-8_7. URL https://www.ecva.net/papers/eccv_2024/papers_ECCV/html/10922_ECCV_2024_paper.php.

Yixian Xu, Yuanrui Zhang, Shengjie Luo, Liwei Wang, and Di He. Designing reinforcement learning for diffusion models: A unified path-space view. arXiv preprint arXiv:2608.14430, 2026. URL https://arxiv.org/abs/2608.14430.

Kai Yang, Jian Tao, Jiafei Lyu, Chunjiang Ge, Jiaxin Chen, Weihan Shen, Xiaolong Zhu, and Xiu Li. Using human feedback to fine-tune diffusion models without any reward model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8941–8951, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Yang_Using_Human_Feedback_to_Fine-tune_Diffusion_Models_without_Any_Reward_CVPR_2024_paper.html.

Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. DiffusionNFT: Online diffusion reinforcement with forward process. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pp. 134129–134150, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/d8e68ddfe22520b45b8fb8d5cbde5a21-Abstract-Conference.html.

Yujie Zhou, Pengyang Ling, Jiazi Bu, Yibin Wang, Yuhang Zang, Jiaqi Wang, Li Niu, and Guangtao Zhai. Fine-grained GRPO for precise preference alignment in flow models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20045–20054, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Zhou_Fine-Grained_GRPO_for_Precise_Preference_Alignment_in_Flow_Models_CVPR_2026_paper.html.

A Endpoint Reweighting and Target Specification

This appendix supports Section 2.2. We first state the reward-improved endpoint density, then propagate a general endpoint reweighting through the shared forward process, and finally distinguish target-preserving reward translation from reward scaling that changes the effective tilt.

We use $p_b(\cdot \mid c)$ for the behavior endpoint density and $q(\cdot \mid c)$ for a candidate endpoint density. Whenever a density ratio is written, it is understood almost everywhere on the behavior support. Conditional identities at a noisy state $s = (x_t, t, c)$ use the same forward process as in Section 2 and hold almost surely whenever the stated moments exist.

A.1 KL-Regularized Endpoint Improvement

For a fixed condition c, endpoint score $A(x_0, c)$, and tilt $\tau > 0$, consider the standard KL-regularized reward-improvement problem (Peters et al., 2010; Abdolmaleki et al., 2018; Peng et al., 2019)

$$q^\star = \arg\max_q \left\{ \mathbb{E}_q[A \mid c] - \frac{1}{\tau} D_{\mathrm{KL}}\big(q(\cdot \mid c) \| p_b(\cdot \mid c)\big) \right\}, \tag{19}$$

where the maximization is over normalized endpoint densities q supported on p_b.

Lemma A.1 (KL-regularized endpoint improvement). *Fix c, $\tau > 0$, and an endpoint score A that is finite almost everywhere on the behavior support. Assume*

$$0 < Z_A(c) := \mathbb{E}_b[e^{\tau A} \mid c] < \infty, \qquad \mathbb{E}_b[|A| e^{\tau A} \mid c] < \infty.$$

Then (19) *has the unique solution*

$$\frac{q^\star(x_0 \mid c)}{p_b(x_0 \mid c)} = \frac{e^{\tau A(x_0, c)}}{Z_A(c)}, \qquad \mathbb{E}_b\left[\frac{q^\star}{p_b} \,\middle|\, c\right] = 1. \tag{20}$$

Equivalently,

$$q^\star(x_0 \mid c) = \frac{e^{\tau A(x_0, c)}}{Z_A(c)} p_b(x_0 \mid c). \tag{21}$$

Proof. Define

$$q^\star(x_0 \mid c) := \frac{e^{\tau A(x_0, c)}}{Z_A(c)} p_b(x_0 \mid c).$$

By the definition of $Z_A(c)$, $q^\star$ integrates to one. For any admissible q, substituting the expression above into the KL divergence gives

$$\begin{aligned} D_{\mathrm{KL}}(q \| q^\star) &= \int q(x_0 \mid c) \log \frac{q(x_0 \mid c)}{q^\star(x_0 \mid c)} \, dx_0 \\ &= \int q(x_0 \mid c) \left[\log \frac{q(x_0 \mid c)}{p_b(x_0 \mid c)} - \tau A(x_0, c) + \log Z_A(c) \right] dx_0 \\ &= D_{\mathrm{KL}}(q \| p_b) - \tau \mathbb{E}_q[A \mid c] + \log Z_A(c). \end{aligned} \tag{22}$$

Rearranging,

$$\mathbb{E}_q[A \mid c] - \tau^{-1} D_{\mathrm{KL}}(q \| p_b) = \tau^{-1} \log Z_A(c) - \tau^{-1} D_{\mathrm{KL}}(q \| q^\star). \tag{23}$$

Since $D_{\mathrm{KL}}(q \| q^\star) \geq 0$, with equality only when $q = q^\star$ almost everywhere, the objective is uniquely maximized by $q^\star$. Finally,

$$\mathbb{E}_b\left[\frac{q^\star}{p_b} \,\middle|\, c\right] = \int q^\star(x_0 \mid c) \, dx_0 = 1,$$

which proves (20). □

Thus KL-regularized reward improvement fixes a normalized endpoint density. Multiplying $e^{\tau A}$ by a positive factor shared within condition c does not change that density because the same factor cancels in $Z_A(c)$.

A.2 Endpoint Reweighting and the Induced Conditional Flow

We next propagate a general positive endpoint weight through the same forward process. Let $w(x_0, c) > 0$ satisfy $0 < \mathbb{E}_b[w \mid c] < \infty$ and define

$$q_w(x_0 \mid c) = \frac{w(x_0, c)}{\mathbb{E}_b[w \mid c]} p_b(x_0 \mid c). \tag{24}$$

At a noisy state $s = (x_t, t, c)$, write $Z_w(s) := \mathbb{E}_b[w \mid s]$.

For fixed c and t, write $p(s \mid x_0, c)$ for the conditional density of the noisy state induced by the shared forward process. The same $p(s \mid x_0, c)$ is used for both the behavior density p_b and the reweighted density q_w.

Proposition A.1 (Conditional flow induced by endpoint reweighting). *Let q_w be defined by (24) and use the same forward process as p_b. For almost every s such that $0 < Z_w(s) < \infty$ and $\mathbb{E}_b[w\|u_t\| \mid s] < \infty$,*

$$q_w(x_0 \mid s) = \frac{w(x_0, c)}{Z_w(s)} p_b(x_0 \mid s), \tag{25}$$

and the induced conditional velocity is

$$v_w^\star(s) = \mathbb{E}_{q_w}[u_t \mid s] = \frac{\mathbb{E}_b[w u_t \mid s]}{Z_w(s)}. \tag{26}$$

Proof. Let

$$M_w(c) := \mathbb{E}_b[w \mid c].$$

First compute the noisy-state density induced by q_w:

$$\begin{aligned} q_w(s \mid c) &= \int q_w(x_0 \mid c)\, p(s \mid x_0, c)\, dx_0 \\ &= \frac{1}{M_w(c)} \int w(x_0, c) p_b(x_0 \mid c) p(s \mid x_0, c)\, dx_0. \end{aligned} \tag{27}$$

Bayes' rule for the behavior distribution gives

$$p_b(x_0 \mid c) p(s \mid x_0, c) = p_b(s \mid c) p_b(x_0 \mid s).$$

Substituting this identity into (27),

$$\begin{aligned} q_w(s \mid c) &= \frac{p_b(s \mid c)}{M_w(c)} \int w(x_0, c) p_b(x_0 \mid s)\, dx_0 \\ &= \frac{Z_w(s)}{M_w(c)} p_b(s \mid c). \end{aligned} \tag{28}$$

Applying Bayes' rule again,

$$\begin{aligned} q_w(x_0 \mid s) &= \frac{q_w(x_0 \mid c) p(s \mid x_0, c)}{q_w(s \mid c)} \\ &= \frac{\frac{w(x_0,c)}{M_w(c)} p_b(x_0 \mid c) p(s \mid x_0, c)}{\frac{Z_w(s)}{M_w(c)} p_b(s \mid c)} \\ &= \frac{w(x_0, c)}{Z_w(s)} p_b(x_0 \mid s), \end{aligned} \tag{29}$$

which proves (25).

Because endpoint reweighting changes only the distribution of x_0 and uses the same forward process, the conditional distribution of u_t given (x_0, s) is unchanged. Therefore,

$$\begin{aligned} \mathbb{E}_{q_w}[u_t \mid s] &= \int \mathbb{E}_b[u_t \mid x_0, s]\, q_w(x_0 \mid s)\, dx_0 \\ &= \frac{1}{Z_w(s)} \int w(x_0, c) \mathbb{E}_b[u_t \mid x_0, s]\, p_b(x_0 \mid s)\, dx_0 \\ &= \frac{\mathbb{E}_b[w u_t \mid s]}{Z_w(s)}, \end{aligned}$$

where the last equality follows from the tower property. This proves (26). □

The endpoint normalizer $\mathbb{E}_b[w \mid c]$ and the noisy-state normalizer $Z_w(s) = \mathbb{E}_b[w \mid s]$ are different objects. In particular, a weight can have behavior mean one at the endpoint level while $Z_w(s) \neq 1$ after conditioning on s. Also, for any positive condition-shared factor $a(c)$, $w' = a(c)w$ gives the same q_w and $v_w^\star$ because the factor cancels from both normalizers.

A.3 Reward Translation, Reward Scaling, and Effective Tilt

Consider the condition-wise affine preprocessing

$$A(x_0, c) = \frac{R(x_0, c) - \zeta(c)}{d(c)}, \qquad d(c) > 0. \tag{30}$$

At fixed $\tau > 0$,

$$e^{\tau A(x_0,c)} = e^{-\tau\zeta(c)/d(c)} e^{(\tau/d(c))R(x_0,c)}. \tag{31}$$

Proposition A.2 (Reward translation and effective tilt). *Under the moment conditions of Lemma A.1, the target induced by* (30) *satisfies*

$$q^\star(x_0 \mid c) = \frac{e^{(\tau/d(c))R(x_0,c)}}{\mathbb{E}_b[e^{(\tau/d(c))R} \mid c]} p_b(x_0 \mid c). \tag{32}$$

Thus changing the condition-shared translation $\zeta(c)$ leaves the target unchanged for fixed $d(c)$, whereas changing $d(c)$ changes the effective tilt

$$\tau_{\text{eff}}(c) = \frac{\tau}{d(c)}. \tag{33}$$

Proof. Substituting (31) into Lemma A.1 cancels the factor $e^{-\tau\zeta(c)/d(c)}$ between numerator and denominator, leaving (32). The remaining raw-reward exponent is $(\tau/d(c))R$, which gives (33). □

The special case $d(c) = 1$ gives reward translation $A = R - \zeta(c)$ and therefore

$$\frac{e^{\tau A(x_0,c)}}{\mathbb{E}_b[e^{\tau A} \mid c]} = \frac{e^{\tau R(x_0,c)}}{\mathbb{E}_b[e^{\tau R} \mid c]}. \tag{34}$$

Hence raw and condition-wise centered rewards specify the same endpoint density and, by Proposition A.1, the same induced conditional flow at a fixed tilt. Scaling by $d(c)$ is different because it changes relative endpoint masses through τ_{eff}.

The same distinction holds for a realized finite rollout group. Conditional on the realized preprocessing statistics, define

$$\widehat{Q}_i^\star := \widehat{Q}^\star(\{x_0^{(i)}\} \mid c).$$

Then

$$\widehat{Q}_i^\star = \frac{e^{\tau A_i}}{\sum_j e^{\tau A_j}} = \frac{e^{(\tau/d(c))R_i}}{\sum_j e^{(\tau/d(c))R_j}}. \tag{35}$$

Thus a group-shared translation cancels exactly, while a group-dependent scale changes the target through the realized effective tilt. This separates target specification from the coefficient-representation question studied next.

B Details for Calibrated Stochastic Realization

Appendix A fixes a reward-improved endpoint law and its induced conditional flow. This appendix supports Section 3.1 and studies the remaining question: for a supplied positive coefficient w, how should that population target be realized by a stochastic field on behavior samples? Coefficient identification is deferred to Appendix C.

B.1 Conditional Moment of the Behavior-Controlled Family

For a supplied coefficient w, Proposition A.1 gives

$$v_w^\star(s) = \frac{\mathbb{E}_b[wu_t \mid s]}{Z_w(s)}, \qquad Z_w(s) = \mathbb{E}_b[w \mid s]. \tag{36}$$

Consider the behavior-controlled family

$$F_{w;\alpha,\beta}(v) = \alpha(s)w(v - u_t) + \beta(s)(u_t - v_b), \qquad \alpha(s) > 0, \tag{37}$$

where $\alpha(s)$ and $\beta(s)$ are fixed after conditioning on s.

Proposition B.1 (Conditional moment of the behavior-controlled family). *Under Assumption 1 and finite conditional moments,*

$$\mathbb{E}_b[F_{w;\alpha,\beta}(v) \mid s] = \alpha(s)Z_w(s)\big(v - v_w^\star(s)\big). \tag{38}$$

Hence every $\alpha(s) > 0$ has the same unique conditional root $v_w^\star(s)$, independently of $\beta(s)$.

Proof. Assumption 1 gives $\mathbb{E}_b[u_t - v_b \mid s] = 0$. Also,

$$\begin{aligned}\mathbb{E}_b[w(v - u_t) \mid s] &= v\mathbb{E}_b[w \mid s] - \mathbb{E}_b[wu_t \mid s] \\ &= Z_w(s)\big(v - v_w^\star(s)\big).\end{aligned}$$

Substitution into (37) proves (38). Since $\alpha(s)Z_w(s) > 0$, the root is unique. □

Thus population-root correctness leaves two freedoms: an overall state-wise scale $\alpha(s)$ and the relative coefficient $\beta(s)/\alpha(s)$ of the zero-mean behavior residual. The next subsection records the calibration used in Section 3.1.

B.2 Identity Calibration and the Finite-Tilt Estimating Equation

For the literal no-reweighting coefficient $w \equiv 1$, Proposition A.1 and Assumption 1 give $v_1^\star = v_b$. Evaluating (37) at this solution,

$$F_{1;\alpha,\beta}(v_b) = \big(\beta(s) - \alpha(s)\big)(u_t - v_b). \tag{39}$$

Thus requiring no samplewise stochastic forcing when no endpoint reweighting is requested gives $\beta(s) = \alpha(s)$ whenever $u_t - v_b$ is not conditionally degenerate. The family becomes

$$F_{w;\alpha}(v) = \alpha(s)\big[w(v - u_t) + (u_t - v_b)\big].$$

The remaining positive factor rescales the entire state-wise field without changing its conditional root. We fix this otherwise free field scale by setting $\alpha(s) = 1$, leaving any additional sample weighting outside the base field:

$$\begin{aligned}F_w(v) &= w(v - u_t) + (u_t - v_b) \\ &= w(v - v_b) - (w - 1)(u_t - v_b).\end{aligned} \tag{40}$$

Applying Proposition B.1 with $\alpha = \beta = 1$ gives

$$\mathbb{E}_b[F_w(v) \mid s] = Z_w(s)\big(v - v_w^\star(s)\big). \tag{41}$$

Since $Z_w(s) > 0$, the unique conditional population root is $v_w^\star(s)$.

Identity calibration therefore fixes how a supplied coefficient enters the base stochastic field. It does not determine which positive representative should be used for a prescribed normalized endpoint target; that identification problem is the subject of Appendix C.

B.3 Control-Variate Interpretation

The behavior residual $u_t - v_b$ has zero conditional mean under Assumption 1, so it can be viewed as a control variate. To make this explicit, retain a scalar coefficient $\beta(s)$ and write

$$F_{w;1,\beta}(v) = w(v - u_t) + \beta(s)(u_t - v_b). \tag{42}$$

For every finite $\beta(s)$,

$$\mathbb{E}_b[F_{w;1,\beta}(v) \mid s] = Z_w(s)(v - v_w^\star(s)),$$

so β changes the samplewise estimator without changing its conditional population root. The choices $\beta = 0$ and $\beta = 1$ recover weighted flow matching and the identity-calibrated field, respectively. Thus, the behavior residual acts as a zero-mean control variate that changes the stochastic realization while preserving the same conditional target.

B.4 Relation to Local Tilt Dynamics

For the exponential family $w_\tau = e^{\tau A}$, define the tilted conditional expectation

$$\mathbb{E}_\tau[f \mid s] := \frac{\mathbb{E}_b[e^{\tau A} f \mid s]}{\mathbb{E}_b[e^{\tau A} \mid s]}. \tag{43}$$

The induced conditional flow is therefore

$$v_\tau^\star(s) := \mathbb{E}_\tau[u_t \mid s] = \frac{\mathbb{E}_b[e^{\tau A} u_t \mid s]}{\mathbb{E}_b[e^{\tau A} \mid s]}. \tag{44}$$

For a vector u_t and scalar A, write

$$\mathrm{Cov}_\tau(u_t, A \mid s) := \mathbb{E}_\tau[A u_t \mid s] - \mathbb{E}_\tau[u_t \mid s]\mathbb{E}_\tau[A \mid s].$$

Assuming the required exponential moments permit differentiation, the quotient rule gives

$$\begin{aligned}\frac{\partial v_\tau^\star(s)}{\partial \tau} &= \frac{\mathbb{E}_b[A e^{\tau A} u_t \mid s]}{\mathbb{E}_b[e^{\tau A} \mid s]} - \frac{\mathbb{E}_b[e^{\tau A} u_t \mid s]\mathbb{E}_b[A e^{\tau A} \mid s]}{\mathbb{E}_b[e^{\tau A} \mid s]^2} \\ &= \mathbb{E}_\tau[A u_t \mid s] - \mathbb{E}_\tau[u_t \mid s]\mathbb{E}_\tau[A \mid s] \\ &= \mathrm{Cov}_\tau(u_t, A \mid s). \end{aligned} \tag{45}$$

This gives the local tilt dynamics of the target flow (Potaptchik et al., 2026). At $\tau = 0$, Assumption 1 gives $v_0^\star = v_b$, and therefore

$$v_\tau^\star(s) = v_b(s) + \tau\, \mathrm{Cov}_b(u_t, A \mid s) + O(\tau^2) \tag{46}$$

whenever the second-order expansion is valid.

Equation (46) characterizes the local response of the target flow around $\tau = 0$. Under Assumption 1,

$$\mathbb{E}_b[A(u_t - v_b) \mid s] = \mathbb{E}_b[A u_t \mid s] - \mathbb{E}_b[A \mid s]\mathbb{E}_b[u_t \mid s] = \mathrm{Cov}_b(u_t, A \mid s).$$

Therefore, for the first-order field

$$F_{\mathrm{local}}(v) := (v - v_b) - \tau A(u_t - v_b),$$

its conditional mean is

$$\begin{aligned}\mathbb{E}_b[F_{\mathrm{local}}(v) \mid s] &= v - v_b - \tau\, \mathrm{Cov}_b(u_t, A \mid s) \\ &= v - v_\tau^\star(s) + O(\tau^2). \end{aligned} \tag{47}$$

Thus, its conditional root matches the tilted flow through first order in τ. This recovers the local first-order response used by explicit Tilt Matching.

The calibrated field gives the corresponding exact finite-tilt relation. Substituting $w = e^{\tau A}$ into (40) and using (41) gives

$$\mathbb{E}_b[F_{e^{\tau A}}(v) \mid s] = \mathbb{E}_b[e^{\tau A} \mid s]\big(v - v_\tau^\star(s)\big), \tag{48}$$

so its conditional root is exactly $v_\tau^\star(s)$ whenever the required moments exist. These two relations distinguish local first-order accuracy from exact finite-tilt realization. Appendix E.2 uses this distinction to compare representative forward-training objectives.

C Details for Coefficient Identification and RootSolve

Appendix B fixes how a supplied coefficient enters the stochastic field. This appendix supports Section 3.2 and addresses the remaining identification problem: the normalized endpoint target determines relative probability mass, while an unnormalized positive coefficient remains free up to condition-wise scaling.

C.1 Proof of Theorem 1: Coefficient Ambiguity

For a positive integrable weight w, write

$$q_w(x_0 \mid c) = \frac{w(x_0, c)}{\mathbb{E}_b[w \mid c]} p_b(x_0 \mid c). \tag{49}$$

Positive weights inducing the same normalized endpoint density differ only by a positive condition-wise factor almost everywhere on the behavior support. The theorem below shows that this target-invariant factor remains in the scale-aligned samplewise field.

Theorem 1 (Coefficient ambiguity yields different stochastic fields). *Fix a positive endpoint weight w with $0 < \mathbb{E}_b[w \mid c] < \infty$ and a finite factor $a(c) > 0$. Under Assumption 1, suppose the required conditional moments exist and $0 < Z_w(s) < \infty$. Then w and $a(c)w$ define the same normalized endpoint density and the same induced conditional flow $v^\star_w(s)$. Their calibrated conditional mean fields have this same unique root. Writing $a = a(c)$, their scale-aligned sample fields satisfy*

$$a^{-1} F_{aw}(v) - F_w(v) = (a^{-1} - 1)(u_t - v_b). \tag{50}$$

Hence the scale-aligned fields generally differ samplewise whenever $a \neq 1$ and $u_t \neq v_b$.

Proof. The normalized endpoint density is invariant because

$$\frac{aw}{\mathbb{E}_b[aw \mid c]} = \frac{w}{\mathbb{E}_b[w \mid c]}.$$

Since $a(c)$ is fixed after conditioning on s, $Z_{aw}(s) = a(c) Z_w(s)$ and Proposition A.1 gives $v^\star_{aw} = v^\star_w$. Using (41) with coefficient aw gives

$$\mathbb{E}_b[F_{aw}(v) \mid s] = a(c) Z_w(s)(v - v^\star_w),$$

so the conditional root is unchanged. Finally,

$$a^{-1} F_{aw}(v) = w(v - u_t) + a^{-1}(u_t - v_b),$$

and subtracting $F_w(v) = w(v - u_t) + (u_t - v_b)$ proves (50). □

Conversely, if another positive integrable weight w' induces the same endpoint density as w, then

$$\frac{w'}{\mathbb{E}_b[w' \mid c]} p_b = \frac{w}{\mathbb{E}_b[w \mid c]} p_b.$$

Therefore, wherever $p_b > 0$,

$$\frac{w'}{\mathbb{E}_b[w' \mid c]} = \frac{w}{\mathbb{E}_b[w \mid c]},$$

and hence $w' = a(c)w$ for some $a(c) > 0$. Thus the positive representatives form the class $[w] = \{a(c)w : a(c) > 0\}$ almost everywhere on the behavior support.

The factor $a(c)$ carries no information about relative endpoint probabilities, yet it remains in the scale-aligned sample field. We next remove this representation freedom using the behavior density as the reference.

C.2 Canonical Target-to-Behavior Ratio

Proposition 1 (Canonical density-ratio representative). *For a fixed condition c, let $p_b(\cdot \mid c)$ be the behavior endpoint density and let $q^\star(\cdot \mid c)$ be a target density supported on the behavior support. Define the target-to-behavior density ratio*

$$\rho(x_0, c) := \frac{q^\star(x_0 \mid c)}{p_b(x_0 \mid c)} > 0 \tag{51}$$

on the behavior support. Then

$$\mathbb{E}_b[\rho \mid c] = 1. \tag{52}$$

Moreover, any positive coefficient w that induces the same target, i.e.,

$$q^\star(x_0 \mid c) = \frac{w(x_0, c)}{\mathbb{E}_b[w \mid c]} p_b(x_0 \mid c), \tag{53}$$

must have the form

$$w(x_0, c) = a(c)\rho(x_0, c) \tag{54}$$

for some $a(c) > 0$. Hence ρ is the unique such coefficient satisfying $\mathbb{E}_b[w \mid c] = 1$, almost everywhere on the behavior support.

Proof. By definition,

$$q^\star(x_0 \mid c) = \rho(x_0, c) p_b(x_0 \mid c).$$

Therefore,

$$\begin{aligned}\mathbb{E}_b[\rho \mid c] &= \int \rho(x_0, c) p_b(x_0 \mid c)\, dx_0 \\ &= \int q^\star(x_0 \mid c)\, dx_0 = 1.\end{aligned}$$

Now suppose a positive coefficient w induces the same target through (53). Comparing the two expressions for $q^\star$ gives

$$\frac{w(x_0, c)}{\mathbb{E}_b[w \mid c]} = \rho(x_0, c)$$

on the behavior support. Hence

$$w(x_0, c) = a(c)\rho(x_0, c), \qquad a(c) := \mathbb{E}_b[w \mid c] > 0.$$

Thus all positive coefficients inducing the same target differ from ρ only by a condition-wise positive scale. If additionally $\mathbb{E}_b[w \mid c] = 1$, then $a(c) = 1$ and therefore $w = \rho$. □

For the KL target of Lemma A.1,

$$\rho(x_0, c) = \frac{q^\star(x_0 \mid c)}{p_b(x_0 \mid c)} = \frac{e^{\tau A(x_0, c)}}{\mathbb{E}_b[e^{\tau A} \mid c]}. \tag{55}$$

Condition-shared reward translation leaves this ratio unchanged by Proposition A.2.

Finite grouped rollouts. For a realized group of N endpoint indices, define the probability simplex

$$\Delta_N := \left\{ q \in \mathbb{R}^N : q_i \geq 0,\ \sum_{i=1}^N q_i = 1 \right\},$$

with the standard convention $0 \log 0 = 0$. For empirical behavior masses $\mu_i > 0$ with $\sum_i \mu_i = 1$, consider

$$q^\star = \arg\max_{q \in \Delta_N} \left\{ \sum_{i=1}^N q_i A_i - \frac{1}{\tau} \sum_{i=1}^N q_i \log \frac{q_i}{\mu_i} \right\}. \tag{56}$$

Proposition C.1 (Empirical target-to-behavior mass ratio). *For uniform empirical behavior masses $\mu_i = 1/N$, finite scores A_i, and $\tau > 0$,* (56) *has the unique solution*

$$q_i^\star = \frac{e^{\tau A_i}}{\sum_j e^{\tau A_j}}, \qquad \widehat{\rho}_i := \frac{q_i^\star}{1/N} = \frac{e^{\tau A_i}}{N^{-1} \sum_j e^{\tau A_j}}, \qquad \frac{1}{N} \sum_i \widehat{\rho}_i = 1. \tag{57}$$

Proof. This is the finite discrete counterpart of Lemma A.1. For general positive masses μ_i satisfying $\sum_i \mu_i = 1$, define

$$Z_N := \sum_i \mu_i e^{\tau A_i}, \qquad q_i^\star := \frac{\mu_i e^{\tau A_i}}{Z_N}.$$

Then, for every $q \in \Delta_N$,

$$\sum_i q_i A_i - \frac{1}{\tau}\sum_i q_i \log\frac{q_i}{\mu_i} = \frac{1}{\tau}\log Z_N - \frac{1}{\tau}\sum_i q_i \log\frac{q_i}{q_i^\star}$$
$$= \frac{1}{\tau}\log Z_N - \frac{1}{\tau} D_{\mathrm{KL}}(q\|q^\star).$$

The objective is therefore uniquely maximized at $q = q^\star$. Setting $\mu_i = 1/N$ gives (57), and $N^{-1}\sum_i \widehat{\rho}_i = \sum_i q_i^\star = 1$. □

In particular, for $A_i = R_i - \zeta(c)$,

$$\widehat{\rho}_i = \frac{e^{\tau A_i}}{N^{-1}\sum_j e^{\tau A_j}} = \frac{e^{\tau R_i}}{N^{-1}\sum_j e^{\tau R_j}}, \tag{58}$$

so condition-shared centering cancels exactly. The same derivation also covers nonuniform empirical behavior masses:

$$q_i^\star = \frac{\mu_i e^{\tau A_i}}{\sum_j \mu_j e^{\tau A_j}}, \qquad \widehat{\rho}_i = \frac{q_i^\star}{\mu_i} = \frac{e^{\tau A_i}}{\sum_j \mu_j e^{\tau A_j}}, \qquad \sum_i \mu_i \widehat{\rho}_i = 1. \tag{59}$$

Parameter-gradient sensitivity to endpoint-equivalent representations. Theorem 1 shows that endpoint-equivalent representatives can produce different scale-aligned sample fields. Their effect in parameter space follows directly by composing the field with the model Jacobian. For the scale-aligned field

$$\widetilde{F}_a(v) := a^{-1} F_{a\rho}(v),$$

and the Jacobian $J_\theta(s) := \partial v_\theta(s)/\partial\theta$ of the live velocity prediction, the base parameter-gradient contributions before any external detached sample weighting are

$$g_\rho = J_\theta(s)^\top F_\rho(v_\theta(s)), \qquad \widetilde{g}_a = J_\theta(s)^\top \widetilde{F}_a(v_\theta(s)).$$

Corollary 1 (Gradient sensitivity to endpoint-equivalent representations). *Under Assumption 1 and finite conditional second moments,*

$$\widetilde{g}_a - g_\rho = (a^{-1}-1) J_\theta(s)^\top (u_t - v_b), \qquad \mathbb{E}_b[\widetilde{g}_a - g_\rho \mid s] = 0.$$

Moreover,

$$\mathbb{E}_b[\|\widetilde{g}_a - g_\rho\|^2 \mid s] = (a^{-1}-1)^2 \mathbb{E}_b[\|J_\theta(s)^\top (u_t - v_b)\|^2 \mid s].$$

If $\mathbb{E}_b[\|J_\theta(s)^\top (u_t - v_b)\|^2 \mid s] > 0$, then $a = 1$ is the unique endpoint-equivalent representative for which $\widetilde{g}_a - g_\rho$ vanishes.

Corollary 1 turns the field-level ambiguity of Theorem 1 into a parameter-gradient consequence. After mean-field scale alignment, endpoint-equivalent representatives have the same conditional mean update, but any $a \neq 1$ introduces a zero-mean gradient perturbation whose conditional second moment scales as $(a^{-1}-1)^2$. Hence, within the endpoint-equivalent class, the canonical representative ρ is the unique choice that removes the gradient perturbation attributable to the free coefficient scale.

Proof of Corollary 1. Taking $w = \rho$ in Theorem 1 gives

$$\widetilde{F}_a(v) - F_\rho(v) = (a^{-1}-1)(u_t - v_b).$$

In the base stochastic loss only the live output $v_\theta(s)$ receives gradients, while the remaining inputs are detached. The chain rule therefore gives

$$\widetilde{g}_a - g_\rho = (a^{-1}-1) J_\theta(s)^\top (u_t - v_b).$$

Conditioned on s, both $a(c)$ and $J_\theta(s)$ are fixed. Assumption 1 gives $\mathbb{E}_b[u_t - v_b \mid s] = 0$, so the conditional mean of $\widetilde{g}_a - g_\rho$ is zero. Because $J_\theta(s)$ is fixed conditional on s, the conditional covariance is

$$\mathrm{Cov}_b(\widetilde{g}_a - g_\rho \mid s) = (a^{-1}-1)^2 J_\theta(s)^\top \mathrm{Cov}_b(u_t - v_b \mid s) J_\theta(s).$$

Taking the squared Euclidean norm of the zero-mean vector gives

$$\mathbb{E}_b[\|\widetilde{g}_a - g_\rho\|^2 \mid s] = (a^{-1}-1)^2 \mathbb{E}_b[\|J_\theta(s)^\top (u_t - v_b)\|^2 \mid s].$$

If this conditional second moment is strictly positive, the right-hand side vanishes if and only if $a = 1$. □

C.3 Population Density Ratio and Empirical Mass Ratio

The population ratio and the finite-group ratio follow the same probability principle but operate at different levels. For a fixed population score function,

$$\rho(x_0, c) = \frac{e^{\tau A(x_0,c)}}{\mathbb{E}_b[e^{\tau A} \mid c]}, \qquad \mathbb{E}_b[\rho \mid c] = 1, \tag{60}$$

whereas a realized finite group replaces the population normalizer by its complete-group counterpart. Conditional on the realized scores, Proposition C.1 exactly solves the finite simplex problem in (56), so $\rho(x_0, c)$ is the population target-to-behavior density ratio while the empirical coefficient $\widehat{\rho}_i$ is a realized target-to-behavior mass ratio for the group.

Endpoint and noisy-state normalization play different roles. At the endpoint level, $\mathbb{E}_b[\rho \mid c] = 1$, while forward noising induces the state-dependent normalizer

$$Z_\rho(s) := \mathbb{E}_b[\rho \mid s]. \tag{61}$$

The target conditional flow is therefore

$$v^\star_\rho(s) = \frac{\mathbb{E}_b[\rho u_t \mid s]}{Z_\rho(s)}. \tag{62}$$

Mean-one normalization and behavior-flow consistency play separate roles. The empirical ratio specifies endpoint mass redistribution, while Assumption 1 connects v_b to the conditional behavior moment used by the conditional exactness results below.

C.4 Proof of Theorem 2: RootSolve Conditional Exactness

Recall

$$Z_\rho(s) = \mathbb{E}_b[\rho \mid s], \qquad v^\star_\rho(s) = \frac{\mathbb{E}_b[\rho u_t \mid s]}{Z_\rho(s)}.$$

Under Assumption 1,

$$\mathbb{E}_b[(\rho - 1)(u_t - v_b) \mid s] = Z_\rho(s)\big(v^\star_\rho(s) - v_b(s)\big). \tag{63}$$

Indeed,

$$\begin{aligned}\mathbb{E}_b[(\rho - 1)(u_t - v_b) \mid s] &= \mathbb{E}_b[\rho u_t \mid s] - \mathbb{E}_b[u_t \mid s] - \mathbb{E}_b[\rho \mid s]v_b + v_b \\ &= \mathbb{E}_b[\rho u_t \mid s] - Z_\rho(s)v_b \\ &= Z_\rho(s)\big(v^\star_\rho - v_b\big),\end{aligned}$$

where the second equality uses $\mathbb{E}_b[u_t \mid s] = v_b$.

Theorem 2 (`RootSolve` conditional exactness). *Suppose Assumption 1 holds, $\rho = q^\star/p_b$ is positive and normalized, and the conditional moments defining the risk are finite. For almost every noisy state s, $\mathbb{E}_b[\ell_{\mathrm{solve}}(\rho; v) \mid s]$ is strictly convex in v with unique minimizer $v^\star_\rho(s)$, the conditional flow induced by $q^\star = \rho p_b$.*

Proof. For fixed s, define $\mathcal{R}_{\mathrm{solve}}(v) = \mathbb{E}_b[\ell_{\mathrm{solve}}(\rho; v) \mid s]$. From the `RootSolve` objective,

$$\nabla_v \mathcal{R}_{\mathrm{solve}}(v) = Z_\rho(s)(v - v_b) - \mathbb{E}_b[(\rho - 1)(u_t - v_b) \mid s].$$

Using (63),

$$\nabla_v \mathcal{R}_{\mathrm{solve}}(v) = Z_\rho(s)\big(v - v^\star_\rho(s)\big), \qquad \nabla^2_v \mathcal{R}_{\mathrm{solve}}(v) = Z_\rho(s) I \succ 0. \tag{64}$$

Thus the unique minimizer is $v = v^\star_\rho(s)$. By Proposition A.1, this is exactly the conditional velocity induced by the endpoint law $q^\star = \rho p_b$. □

The noisy-state normalizer $Z_\rho(s)$ appears as the positive curvature of the conditional risk. Together with Proposition 1, Theorem 2 separates the two roles of ρ: probability normalization identifies the coefficient, and the calibrated objective realizes the conditional flow induced by that coefficient.

C.5 IDENTITY-TARGET SAMPLEWISE CANCELLATION

Appendix C.2 quantifies the parameter-gradient sensitivity of endpoint-equivalent representations for a general target. The identity target gives the stronger samplewise cancellation result below. For $w = a(c)\rho$, write

$$a := a(c) > 0.$$

After conditioning on $s = (x_t, t, c)$, a is fixed. Define

$$\widetilde{F}_a(v) := a^{-1} F_{a\rho}(v) = \rho(v - u_t) + a^{-1}(u_t - v_b). \tag{65}$$

Corollary C.1 (Identity-target cancellation). *If $q^\star = p_b$, then $\rho = 1$ and $v^\star_\rho = v_b$. At the behavior solution,*

$$\widetilde{F}_a(v_b) = (a^{-1} - 1)(u_t - v_b). \tag{66}$$

Consequently,

$$\mathrm{Cov}\big(\widetilde{F}_a(v_b) \mid s\big) = (a^{-1} - 1)^2 \,\mathrm{Cov}(u_t - v_b \mid s), \tag{67}$$

and the mean-one representative $a = 1$ satisfies $\widetilde{F}_1(v_b) = 0$ samplewise.

Proof. When $q^\star = p_b$, the target-to-behavior ratio is $\rho = 1$, and Assumption 1 gives $v^\star_\rho = v_b$. Substituting $\rho = 1$ and $v = v_b$ into (65) gives (66). Since a is fixed conditional on s, taking conditional covariance gives (67). Setting $a = 1$ gives the samplewise cancellation. □

The same computation gives the conditional second moment

$$\mathbb{E}_b[\|\widetilde{F}_a(v_b)\|^2 \mid s] = (a^{-1} - 1)^2 \,\mathbb{E}_b[\|u_t - v_b\|^2 \mid s].$$

Hence, if $\mathbb{E}_b[\|u_t - v_b\|^2 \mid s] > 0$, then $a = 1$ is the unique scale-aligned representative whose conditional second moment vanishes. At the identity target, the canonical representative therefore eliminates samplewise stochastic forcing at the behavior solution.

D DETAILS FOR ROOTTRACE

This appendix derives the moving-reference update in Section 3.3 and analyzes its implementation across reused passes. Within an outer iteration, scored endpoints and endpoint ratios remain fixed. Forward targets, coefficients, and reference predictions are detached during differentiation. The identities below hold samplewise; their empirical versions follow by substituting $\widehat{\rho}$ for ρ.

D.1 DERIVING THE MOVING-REFERENCE UPDATE

`RootTrace` preserves the reward-dependent correction in (14) and replaces the fixed behavior reference with the prediction available at the current pass. Keeping the reference term's unit coefficient gives $v - \bar{v}_{k-1}$. Therefore,

$$\begin{aligned} F^{(k)}_{\mathrm{trace}}(\rho; v, \bar{v}_{k-1}) &= (v - \bar{v}_{k-1}) + (\rho - 1)(v - u_t) \\ &= F_{\mathrm{solve}}(\rho; v) - (\bar{v}_{k-1} - v_b). \end{aligned}$$

The subtraction removes the behavior-restoring term evaluated at the current reference. For a fixed reference, the subtracted vector is independent of the supplied endpoint ratio. Hence, the field difference induced by reweighting remains unchanged,

$$\begin{aligned} F^{(k)}_{\mathrm{trace}}(\rho; v, \bar{v}_{k-1}) - F^{(k)}_{\mathrm{trace}}(1; v, \bar{v}_{k-1}) &= F_{\mathrm{solve}}(\rho; v) - F_{\mathrm{solve}}(1; v) \\ &= (\rho - 1)(v - u_t). \end{aligned} \tag{68}$$

For $\rho = 1$, the moving-reference field becomes $v - \bar{v}_{k-1}$ and vanishes at $v = \bar{v}_{k-1}$.

Scalar objective. The pass loss follows by removing the linear contribution of the behavior-restoring gradient at $\bar{v}_{k-1}$. Holding this reference fixed during differentiation gives

$$\ell^{(k)}_{\mathrm{trace}}(\rho; v, \bar{v}_{k-1}) = \ell_{\mathrm{solve}}(\rho; v) - \ell_{\mathrm{solve}}(\rho; \bar{v}_{k-1}) - \langle \bar{v}_{k-1} - v_b,\, v - \bar{v}_{k-1} \rangle. \tag{69}$$

The second term is constant with respect to v, and the final term subtracts $\bar{v}_{k-1} - v_b$ from the output gradient,

$$\nabla_v \ell^{(k)}_{\mathrm{trace}}(\rho; v, \bar{v}_{k-1}) = F_{\mathrm{solve}}(\rho; v) - (\bar{v}_{k-1} - v_b).$$

Expanding the expression gives

$$\ell^{(k)}_{\mathrm{trace}}(\rho; v, \bar{v}_{k-1}) = \frac{\rho}{2}\|v - \bar{v}_{k-1}\|^2 - (\rho - 1)\langle v - \bar{v}_{k-1}, u_t - \bar{v}_{k-1}\rangle,$$

which is (17) and has output gradient (16). For $\rho = 1$, it reduces to

$$\ell^{(k)}_{\mathrm{trace}}(1; v, \bar{v}_{k-1}) = \frac{1}{2}\|v - \bar{v}_{k-1}\|^2,$$

whose gradient vanishes at the current reference.

The construction continues to use the probability-defined coefficient from Section 3.2. For fixed forward samples and model references, canonicalizing endpoint-equivalent weights produces the same ρ and therefore the same `RootTrace` field.

D.2 Gradients on Reused Passes

On passes $k \geq 2$, Algorithm 1 evaluates the live prediction and constructs the reference from its detached value,

$$v = v_{\theta_{k-1}}(s), \qquad \bar{v}_{k-1} = \mathrm{sg}[v_{\theta_{k-1}}(s)].$$

The reference is held fixed during differentiation. Evaluating the resulting output gradient at this point gives

$$F^{(k)}_{\mathrm{trace}}(\rho; \bar{v}_{k-1}, \bar{v}_{k-1}) = (\rho - 1)(\bar{v}_{k-1} - u_t).$$

For comparison, `RootSolve` evaluated at the same prediction and forward sample gives

$$F_{\mathrm{solve}}(\rho; \bar{v}_{k-1}) = (\bar{v}_{k-1} - v_b) + (\rho - 1)(\bar{v}_{k-1} - u_t).$$

Their difference is exactly

$$\begin{aligned} &F_{\mathrm{solve}}(\rho; \bar{v}_{k-1}) - F^{(k)}_{\mathrm{trace}}(\rho; \bar{v}_{k-1}, \bar{v}_{k-1}) \\ &\quad = \bar{v}_{k-1} - v_b. \end{aligned} \tag{70}$$

The moving-reference update therefore removes the behavior-restoring contribution at the current prediction while retaining the reward-dependent correction.

Using the model Jacobian defined in Appendix C.2, the base parameter-gradient contribution is

$$\nabla_\theta \ell^{(k)}_{\mathrm{trace}}(\rho; v_\theta(s), \bar{v}_{k-1})\Big|_{\theta=\theta_{k-1}} = J_{\theta_{k-1}}(s)^\top (\rho - 1)(\bar{v}_{k-1} - u_t). \tag{71}$$

The corresponding gradient difference from `RootSolve` is

$$\begin{aligned} &\nabla_\theta \left[\ell_{\mathrm{solve}}(\rho; v_\theta(s)) - \ell^{(k)}_{\mathrm{trace}}(\rho; v_\theta(s), \bar{v}_{k-1})\right]\Big|_{\theta=\theta_{k-1}} \\ &\quad = J_{\theta_{k-1}}(s)^\top (\bar{v}_{k-1} - v_b). \end{aligned} \tag{72}$$

The reuse gradient applies the fixed coefficient $\rho - 1$ to the current prediction residual through the current model Jacobian. Relative to `RootSolve`, the removed contribution is $J_{\theta_{k-1}}(s)^\top (\bar{v}_{k-1} - v_b)$, the parameter gradient associated with restoring the original behavior prediction.

Endpoints with unchanged mass. For an endpoint with unchanged mass, $\rho = 1$, the reuse-pass policy gradient is zero at the current prediction. In particular, equal scores within a realized group give $\widehat{\rho}_i = 1$ for every endpoint, so the group contributes zero base policy gradient on reuse passes.

For a general ratio, the negative output field is

$$-F^{(k)}_{\mathrm{trace}}(\rho; \bar{v}_{k-1}, \bar{v}_{k-1}) = (\rho - 1)(u_t - \bar{v}_{k-1}).$$

Increased endpoint mass therefore gives an output-space direction toward the sample's forward target, while decreased mass reverses that direction. Both are evaluated at the current prediction.

Although the scalar objective evaluates to zero when the live prediction and detached reference coincide, its gradient can be nonzero because differentiation holds the reference fixed. After the parameter update, the next pass resamples forward time and noise and refreshes the reference from the updated model.

These statements concern base policy-gradient contributions. With identical finite detached sample weights, the gradient-difference identities are multiplied by those weights samplewise, and the cancellation at $\rho = 1$ is preserved. Practical weighting follows Appendix F.1.

D.3 Exact Single-Pass Reduction

Corollary 2 (Exact single-pass reduction). *For every $K \geq 1$, the first-pass `RootTrace` objective equals `RootSolve` for every live output v. Consequently, $K = 1$ gives identical scalar objectives, output gradients, and parameter gradients under the same detached inputs, weights, and reductions.*

Proof. The first-pass reference is $\bar{v}_0 = v_b$. Substituting this reference into (17) gives

$$\begin{aligned} \ell^{(1)}_{\mathrm{trace}}(\rho; v, v_b) &= \frac{\rho}{2}\|v - v_b\|^2 - (\rho - 1)\langle v - v_b, u_t - v_b\rangle \\ &= \ell_{\mathrm{solve}}(\rho; v). \end{aligned} \tag{73}$$

Equality as functions of the live output implies identical output gradients. Composition with the same network and the same detached weighting and reductions gives identical parameter gradients. This identity holds for every live prediction, including predictions that differ from the behavior output. □

For the population coefficient, Theorem 2 characterizes this first-pass objective under its stated assumptions.

E Relations among Forward-Training Realizations

The preceding appendices separate target specification, stochastic realization, and coefficient identification within `ROOT`. We now place representative forward-process reward-tuning objectives under a common velocity convention. The comparison focuses on two properties: the accuracy of the induced conditional target at finite reward tilt and the dependence of the stochastic field on the coefficient representation.

E.1 Unified Stochastic Fields

We first collect the output-gradient fields of representative objectives. Positive method-wide gradient factors, external adaptive weights, batch reductions, and optimizer effects are omitted.

Weighted Flow Matching (WFM) (Fan et al., 2025). For a positive endpoint weight w, WFM uses the sample objective

$$\ell_{\mathrm{WFM}}(v) = w\|v - u_t\|^2.$$

Its output gradient is

$$\frac{1}{2}\nabla_v \ell_{\mathrm{WFM}}(v) = w(v - u_t) = w(v - v_b) - w(u_t - v_b).$$

Hence the corresponding sample field is

$$F_{\mathrm{WFM}}(v) = w(v - v_b) - w(u_t - v_b). \tag{74}$$

DiffusionNFT (Zheng et al., 2026). DiffusionNFT defines the positive and negative velocity branches

$$v^+_\theta = (1 - \beta_{\mathrm{NFT}})v_b + \beta_{\mathrm{NFT}} v, \qquad v^-_\theta = (1 + \beta_{\mathrm{NFT}})v_b - \beta_{\mathrm{NFT}} v, \tag{75}$$

and uses the sample objective

$$\ell_{\mathrm{NFT}}(v) = r\|v^+_\theta - u_t\|^2 + (1 - r)\|v^-_\theta - u_t\|^2, \tag{76}$$

where $r \in [0,1]$ is its reward-derived optimality probability and $\beta_{\mathrm{NFT}} > 0$ is the branch coefficient. Using

$$v_\theta^+ - u_t = \beta_{\mathrm{NFT}}(v - v_b) - (u_t - v_b),$$

and

$$v_\theta^- - u_t = -\beta_{\mathrm{NFT}}(v - v_b) - (u_t - v_b),$$

the objective expands as

$$\begin{aligned}\ell_{\mathrm{NFT}}(v) &= r\,\|\beta_{\mathrm{NFT}}(v - v_b) - (u_t - v_b)\|^2 \\ &\quad + (1-r)\,\|-\beta_{\mathrm{NFT}}(v - v_b) - (u_t - v_b)\|^2 \\ &= \beta_{\mathrm{NFT}}^2\|v - v_b\|^2 - 2\beta_{\mathrm{NFT}}(2r-1)\langle v - v_b, u_t - v_b\rangle + \|u_t - v_b\|^2. \end{aligned} \tag{77}$$

Therefore,

$$\begin{aligned}\nabla_v \ell_{\mathrm{NFT}}(v) &= 2\beta_{\mathrm{NFT}}^2(v - v_b) - 2\beta_{\mathrm{NFT}}(2r-1)(u_t - v_b) \\ &= 2\beta_{\mathrm{NFT}}^2\left[(v - v_b) - \frac{2r-1}{\beta_{\mathrm{NFT}}}(u_t - v_b)\right]. \end{aligned} \tag{78}$$

After removing the positive overall factor $2\beta_{\mathrm{NFT}}^2$, the scale-aligned field is

$$F_{\mathrm{NFT}}(v) = (v - v_b) - \frac{2r-1}{\beta_{\mathrm{NFT}}}(u_t - v_b). \tag{79}$$

Explicit Tilt Matching (ETM) (Potaptchik et al., 2026). Under the aligned velocity convention, ETM uses the reward-linearized target

$$v_b + \tau A(u_t - v_b).$$

Its sample regression objective can therefore be written as

$$\ell_{\mathrm{ETM}}(v) = \|v - v_b - \tau A(u_t - v_b)\|^2. \tag{80}$$

Taking the output gradient gives

$$\frac{1}{2}\nabla_v \ell_{\mathrm{ETM}}(v) = (v - v_b) - \tau A(u_t - v_b),$$

and hence

$$F_{\mathrm{ETM}}(v) = (v - v_b) - \tau A(u_t - v_b). \tag{81}$$

Implicit Tilt Matching (ITM) (Potaptchik et al., 2026). For $w = e^{\tau A}$, ITM uses a target containing the detached live prediction,

$$\ell_{\mathrm{ITM}}(v) = \big\|v - v_b - (w-1)\big(u_t - \mathrm{sg}(v)\big)\big\|^2. \tag{82}$$

Because $\mathrm{sg}(v)$ is detached, differentiating with respect to the live output gives

$$\frac{1}{2}\nabla_v \ell_{\mathrm{ITM}}(v) = (v - v_b) - (w-1)\big(u_t - \mathrm{sg}(v)\big). \tag{83}$$

The stop-gradient has the same numerical value as v in the forward pass, so the corresponding sample field is

$$\begin{aligned}F_{\mathrm{ITM}}(v) &= (v - v_b) - (w-1)(u_t - v) \\ &= w(v - v_b) - (w-1)(u_t - v_b). \end{aligned} \tag{84}$$

Thus, for a supplied exponential coefficient $w = e^{\tau A}$, the identity-calibrated field in (40) has the same output-gradient form as ITM.

Unit-Displacement Ratio (UDR). To separate coefficient identification from the way the identified coefficient enters the stochastic field, we consider a controlled realization that uses the same normalized endpoint ratio ρ as `RootSolve` while fixing the coefficient of $v - v_b$ to one. Its sample objective is

$$\ell_{\mathrm{UDR}}(\rho; v) = \frac{1}{2}\|v - v_b\|^2 - (\rho - 1)\langle v - v_b, u_t - v_b\rangle. \tag{85}$$

Its output gradient is

$$F_{\mathrm{UDR}}(\rho; v) = (v - v_b) - (\rho - 1)(u_t - v_b). \tag{86}$$

UDR and `RootSolve` therefore use the same probability-defined endpoint ratio and the same signed endpoint correction. They differ only in the displacement coefficient, 1 versus ρ. Because UDR is defined directly from the normalized ratio ρ, it contains no free endpoint-equivalent coefficient scale. Appendix E.2 analyzes its conditional target and local tilt accuracy.

`RootSolve`. For comparison, the `RootSolve` sample objective is

$$\ell_{\mathrm{solve}}(\rho; v) = \frac{\rho}{2}\|v - v_b\|^2 - (\rho - 1)\langle v - v_b, u_t - v_b\rangle. \tag{87}$$

Its output gradient is

$$F_{\mathrm{solve}}(\rho; v) = \rho(v - v_b) - (\rho - 1)(u_t - v_b). \tag{88}$$

Table 2 collects the resulting stochastic fields.

Table 2: Forward reward-tuning fields under a common velocity convention. Positive method-wide gradient factors are omitted. The behavior-anchor correction is $-F(v_b)$, and the final column gives the coefficient multiplying $v - v_b$.

Method	Reward coefficient	Sample field $F(v)$	Behavior-anchor correction	Displacement coefficient
WFM	w	$w(v - v_b) - w(u_t - v_b)$	$w(u_t - v_b)$	w
DiffusionNFT	$(2r - 1)/\beta_{\mathrm{NFT}}$	$(v - v_b) - \frac{2r-1}{\beta_{\mathrm{NFT}}}(u_t - v_b)$	$\frac{2r-1}{\beta_{\mathrm{NFT}}}(u_t - v_b)$	1
ETM	τA	$(v - v_b) - \tau A(u_t - v_b)$	$\tau A(u_t - v_b)$	1
ITM	$w = e^{\tau A}$	$w(v - v_b) - (w - 1)(u_t - v_b)$	$(w - 1)(u_t - v_b)$	w
UDR	normalized ρ	$(v - v_b) - (\rho - 1)(u_t - v_b)$	$(\rho - 1)(u_t - v_b)$	1
`RootSolve`	normalized ρ	$\rho(v - v_b) - (\rho - 1)(u_t - v_b)$	$(\rho - 1)(u_t - v_b)$	ρ

Table 2 separates the coefficient supplied to each objective from the way that coefficient enters the stochastic field. The resulting fields differ further in whether their conditional population roots match the finite-tilt target exactly and in whether endpoint-equivalent coefficient representations remain visible in the samplewise update. We study these two properties next.

E.2 Finite-Tilt Accuracy and Coefficient Identification

The unified fields in Appendix E.1 differ along two independent axes. The first concerns target-flow accuracy, whether the conditional population root matches the flow induced by the finite endpoint tilt. The second concerns coefficient identification, whether the coefficient entering the stochastic field is uniquely determined by the normalized endpoint target.

Local and finite-tilt accuracy. The exponential family $w_\tau = e^{\tau A}$ and the normalized ratio $\rho_\tau = e^{\tau A}/\mathbb{E}_b[e^{\tau A} \mid c]$ induce the same normalized endpoint target and therefore the same target conditional flow. The expansion in Appendix B.4 gives $v^\star_\tau(s) = v_b(s) + \tau\,\mathrm{Cov}_b(u_t, A \mid s) + O(\tau^2)$. We call a realization first-order accurate when its conditional root differs from $v^\star_\tau$ by $O(\tau^2)$ around $\tau = 0$, and finite-tilt exact when the conditional roots coincide for finite τ. ETM uses the field $F_{\mathrm{ETM}}(v) = (v - v_b) - \tau A(u_t - v_b)$, whose conditional root agrees with the target through first order by the covariance identity of Appendix B.4. UDR uses $F_{\mathrm{UDR}}(\rho_\tau; v) = (v - v_b) - (\rho_\tau - 1)(u_t - v_b)$, and Proposition E.1 shows an $O(\tau^2)$ target error, so UDR is first-order accurate. DiffusionNFT retains the scale-aligned field $F_{\mathrm{NFT}}(v) = (v - v_b) - \gamma_{\mathrm{NFT}}(u_t - v_b)$ with $\gamma_{\mathrm{NFT}} = (2r - 1)/\beta_{\mathrm{NFT}}$. If its effective local coefficient satisfies $\gamma_{\mathrm{NFT}} = \tau(A - \zeta(c)) + O(\tau^2)$ for a condition-shared $\zeta(c)$, then its conditional mean has the same first-order target response, because the condition-shared term

multiplies the zero-mean residual $u_t - v_b$. DiffusionNFT defines r through its own bounded optimality construction, so this statement does not identify γ_{NFT} with the endpoint density ratio ρ_τ. For $w_\tau = e^{\tau A}$, WFM satisfies $\mathbb{E}_b[F_{\mathrm{WFM}}(v) \mid s] = \mathbb{E}_b[w_\tau \mid s](v - v^\star_\tau(s))$, so its conditional root is finite-tilt exact. By Appendix E.1 and Appendix B.2, ITM has the same output-gradient form as the calibrated field with $w = e^{\tau A}$ under the aligned convention, so its conditional root is finite-tilt exact. `RootSolve` replaces the unnormalized exponential coefficient by ρ_τ, and Theorem 2 gives a finite-tilt exact conditional root.

Coefficient representation. How coefficient non-identification affects the stochastic update depends on the realization in Appendix E.1. WFM satisfies $a^{-1}F^{\mathrm{WFM}}_{aw}(v) = F^{\mathrm{WFM}}_{w}(v)$, so a condition-wise factor appears only as a common field scale after scale alignment. For ETM, the map $w \mapsto a(c)w$ corresponds to $A \mapsto A + \log a(c)/\tau$, so its field changes samplewise by $-\log a(c)\,(u_t - v_b)$, a term with zero conditional mean under Assumption 1. For the calibrated field and ITM, Theorem 1 gives $a^{-1}F_{aw}(v) - F_w(v) = (a^{-1} - 1)(u_t - v_b)$, so the endpoint-equivalent factor remains in the scale-aligned samplewise field although the normalized endpoint target and conditional root are unchanged. UDR and `RootSolve` first identify $\rho = q^\star/p_b$, so Proposition 1 removes the condition-wise coefficient scale before the coefficient enters either field. They then differ only in the displacement coefficient, 1 for UDR and ρ for `RootSolve`. DiffusionNFT uses a separately defined bounded optimality coefficient and does not belong directly to the positive endpoint-weight equivalence class $[w] = \{a(c)w : a(c) > 0\}$. We therefore retain it as an external forward-process baseline rather than assigning it the same representation ambiguity.

Table 3: Accuracy and coefficient-representation summary for the unified fields of Appendix E.1.

Method	Effective coefficient	Target-flow accuracy	Effect of endpoint-equivalent rescaling
WFM	$w = e^{\tau A}$	Finite-tilt exact	Common scale only
DiffusionNFT	$\gamma_{\mathrm{NFT}} = (2r-1)/\beta_{\mathrm{NFT}}$	Local first-order form under coefficient matching	Separate coefficient construction
ETM	τA	First order	Samplewise
ITM	$w = e^{\tau A}$	Finite-tilt exact	Samplewise
UDR	ρ	First order	Removed
`RootSolve`	ρ	Finite-tilt exact	Removed

Exactness of the population root is distinct from estimator variance. Tilt Matching shows that, under its stated small-tilt conditions, the gradient variance of reweighted flow matching is no smaller than that of its implicit tilt formulation (Potaptchik et al., 2026).

Proposition E.1 (Conditional target and local accuracy of UDR)**.** *For $F_{\mathrm{UDR}}(\rho; v) = (v - v_b) - (\rho - 1)(u_t - v_b)$, under Assumption 1 and finite conditional moments,*

$$\mathbb{E}_b[F_{\mathrm{UDR}}(\rho; v) \mid s] = (v - v_b) - Z_\rho(s)\big(v^\star_\rho(s) - v_b(s)\big).$$

The conditional risk is strictly convex with unique minimizer

$$v^\star_{\mathrm{UDR}}(s) = v_b(s) + Z_\rho(s)\big(v^\star_\rho(s) - v_b(s)\big),$$

so that

$$v^\star_{\mathrm{UDR}}(s) - v^\star_\rho(s) = \big(Z_\rho(s) - 1\big)\big(v^\star_\rho(s) - v_b(s)\big).$$

Under the local exponential-moment and differentiability assumptions of Appendix B.4, with $\rho_\tau(x_0, c) = e^{\tau A(x_0,c)}/\mathbb{E}_b[e^{\tau A} \mid c]$,

$$v^\star_{\mathrm{UDR},\tau}(s) - v^\star_{\rho_\tau}(s) = \tau^2\big(\mathbb{E}_b[A \mid s] - \mathbb{E}_b[A \mid c]\big)\,\mathrm{Cov}_b(u_t, A \mid s) + O(\tau^3),$$

so $v^\star_{\mathrm{UDR},\tau} - v^\star_{\rho_\tau} = O(\tau^2)$.

Proof of Proposition E.1. By Appendix C.4, $\mathbb{E}_b[(\rho-1)(u_t-v_b) \mid s] = Z_\rho(s)(v^\star_\rho(s)-v_b(s))$. Taking the conditional expectation of $F_{\mathrm{UDR}}(\rho; v) = (v-v_b)-(\rho-1)(u_t-v_b)$ and using $\mathbb{E}_b[u_t-v_b \mid s] = 0$ gives

$$\mathbb{E}_b[F_{\mathrm{UDR}}(\rho; v) \mid s] = (v - v_b) - Z_\rho(s)\big(v^\star_\rho(s) - v_b(s)\big).$$

The scalar UDR loss has identity Hessian in v, so the conditional risk is strictly convex and its unique minimizer sets the conditional mean field to zero, giving $v^\star_{\mathrm{UDR}}(s)$ and the stated exact error.

For the local expansion, expanding $\rho_\tau = e^{\tau A}/\mathbb{E}_b[e^{\tau A} \mid c]$ in τ gives

$$\rho_\tau = 1 + \tau\big(A - \mathbb{E}_b[A \mid c]\big) + O(\tau^2),$$

and conditioning on s yields

$$Z_{\rho_\tau}(s) = 1 + \tau\big(\mathbb{E}_b[A \mid s] - \mathbb{E}_b[A \mid c]\big) + O(\tau^2).$$

Because condition-wise normalization does not change the induced conditional flow, the local tilt expansion of Appendix B.4 gives

$$v^\star_{\rho_\tau}(s) = v_b(s) + \tau\,\mathrm{Cov}_b(u_t, A \mid s) + O(\tau^2).$$

Substituting these two first-order factors into the exact error $v^\star_{\mathrm{UDR},\tau} - v^\star_{\rho_\tau} = (Z_{\rho_\tau}(s) - 1)(v^\star_{\rho_\tau}(s) - v_b(s))$ gives the stated $O(\tau^2)$ leading term. □

E.3 Implications for Empirical Comparison

DiffusionNFT is the primary published forward-process baseline. ITM and UDR are controlled comparisons tied directly to the analysis above.

ITM uses the same finite-tilt calibrated field form as `RootSolve` but supplies the unnormalized exponential coefficient $w = e^{\tau A}$ instead of the identified ratio ρ. The comparison between ITM and `RootSolve` therefore probes coefficient identification while retaining the finite-tilt realization.

UDR uses exactly the same normalized ratio ρ and endpoint correction as `RootSolve` while fixing the coefficient of $v - v_b$ to one. The comparison between UDR and `RootSolve` therefore probes the stochastic realization of a fixed identified coefficient, with UDR locally first-order accurate around the identity tilt and `RootSolve` finite-tilt exact.

ETM and WFM remain analytical references in Table 3. UDR provides the coefficient-matched first-order control, while Tilt Matching has characterized the variance relation between reweighted flow matching and its implicit formulation under the stated small-tilt conditions. The supplementary empirical comparison therefore focuses on DiffusionNFT, ITM, and UDR.

F Implementation Details and Additional Experiments

This appendix records the practical `ROOT` objective, the common training and evaluation configuration, and the supplementary experiments used in Section 4.

F.1 `ROOT` Training Algorithm and Practical Objective

For each condition c in the current outer-iteration batch, let $\widehat{P}_b(\cdot \mid c)$ denote the empirical behavior measure over its N scored endpoints, and let $\widehat{\rho}(x_0, c)$ denote the corresponding empirical target-to-behavior mass ratio defined in Section 3.2. The scored endpoints, rewards, empirical ratios, and behavior parameters are fixed throughout the outer iteration. `RootSolve` uses one training pass, while `RootTrace` reuses the same scored endpoints for K passes.

At each pass, fresh forward times and noises are sampled for every scored endpoint. We write the ROOT-specific policy term compactly as

$$\mathcal{L}^{(k)}_{\mathrm{policy}}(\theta) = \mathbb{E}_{c,\, x_0 \sim \widehat{P}_b(\cdot|c),\, t}\left[\ell^{(k)}_{\mathrm{policy}}(\theta)\right], \tag{89}$$

where (s, u_t) is sampled from the forward process in Eq. (1) conditional on (x_0, c, t). The empirical endpoint ratio $\widehat{\rho}(x_0, c)$ is fixed across repeated forward-noising samples of the same endpoint.

For `RootSolve` ($K = 1$),

$$\ell^{(1)}_{\mathrm{policy}}(\theta) = \ell_{\mathrm{solve}}(\widehat{\rho}(x_0, c); v_\theta(s))\,. \tag{90}$$

For `RootTrace` ($K > 1$), every pass uses the same empirical endpoint ratio $\widehat{\rho}(x_0, c)$. With the detached reference defined in Section 3.3, the sample policy term is

$$\ell^{(k)}_{\text{policy}}(\theta) = \frac{\widehat{\rho}(x_0,c)}{2}\|v_\theta(s) - \bar{v}_{k-1}(s)\|^2 - \big(\widehat{\rho}(x_0,c) - 1\big)\big\langle v_\theta(s) - \bar{v}_{k-1}(s), u_t - \bar{v}_{k-1}(s)\big\rangle. \tag{91}$$

The first-pass reference is the behavior prediction. On subsequent passes, the reference is the detached live prediction obtained from the same forward evaluation. Only the live prediction receives gradients. Corollary 2 gives exact agreement with `RootSolve` on the first pass for every K, including the single-pass case.

Algorithm 1 `ROOT` training with K passes for one outer iteration

Require: Live parameters θ, behavior parameters $\bar{\theta}$, conditions $\{c_g\}_{g=1}^{B}$, rollouts per condition N, training times per endpoint M, passes K, tilt τ

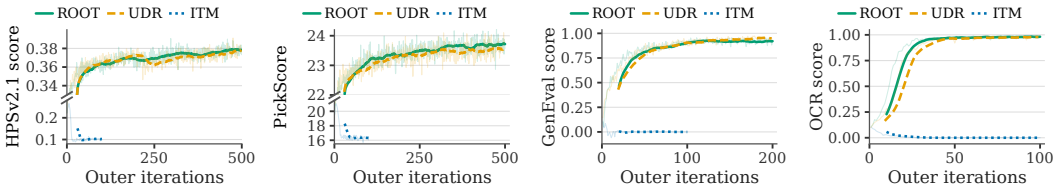
Sample $x_{0,g}^{(i)} \sim p_b(\cdot \mid c_g)$ for $g = 1, \dots, B$ and $i = 1, \dots, N$
Evaluate rewards $R_{g,i} = R(x_{0,g}^{(i)}, c_g)$ once
Construct endpoint scores $A_{g,i}$ and empirical target-to-behavior mass ratios $\widehat{\rho}_{g,i}$ within each condition
Freeze endpoints, rewards, ratios, and behavior parameters $\bar{\theta}$
for $k = 1, \dots, K$ **do**
  Reuse the fixed empirical endpoint ratios $\widehat{\rho}_{g,i}$ for all g, i
  For each (g, i), sample M forward times and noises to obtain $\{(s_{g,i,m}^{(k)}, u_{g,i,m}^{(k)})\}_{m=1}^{M}$
  Evaluate $v_\theta(s_{g,i,m}^{(k)})$ with gradients for all g, i, m
  if $k = 1$ **then**
    Set $\bar{v}_{0,g,i,m} \leftarrow v_b(s_{g,i,m}^{(1)})$ for all g, i, m
  else
    Set $\bar{v}_{k-1,g,i,m} \leftarrow \mathrm{sg}[v_\theta(s_{g,i,m}^{(k)})]$ for all g, i, m
  end if
  Compute $\ell^{(k)}_{\text{policy},g,i,m}$ from (91) using $\widehat{\rho}_{g,i}$ and $\bar{v}_{k-1,g,i,m}$ for all g, i, m
  Apply the same detached adaptive loss weighting as DiffusionNFT and average over all g, i, m to obtain $\mathcal{L}^{(k)}_{\text{policy}}$
  Update θ with $\nabla_\theta \mathcal{L}^{(k)}_{\text{policy}}$
end for
Soft-update $\bar{\theta}$ from the final live parameters θ

Following DiffusionNFT, we use detached adaptive loss weighting, EMA behavior updates, and Classifier-Free Guidance (CFG)-free optimization. These shared practical components are applied consistently across matched comparisons and are separate from the `RootSolve` and `RootTrace` stochastic fields defined above.

F.2 Training Configurations

All experiments fine-tune Stable Diffusion 3.5 Medium at 512×512 resolution using LoRA with rank 32 and scaling factor 64. Unless stated otherwise, we use Adam with learning rate 3×10^{-4}, $\tau = 1$ for `ROOT`, and $\beta = 1$ for DiffusionNFT. Each outer iteration uses $B = 48$ conditions with $N = 24$ behavior endpoints per condition. Rollouts use 10 first-order ODE sampling steps, and each endpoint is re-noised at $M = 9$ training times per pass. Training is CFG-free with guidance scale 1.0, while evaluation uses 40 sampling steps.

We use the same EMA schedule in all experiments. At outer iteration i, the EMA coefficient is $\eta_i = \min(0.001i,\, 0.5)$. For multi-pass comparisons, the scored endpoint batch and reward evaluations are shared across passes, while forward time and noise are independently resampled at every pass. Unless explicitly stated otherwise, the remaining optimization and sampling settings follow DiffusionNFT.



(a) HPSv2.1 (b) PickScore (c) GenEval (d) OCR

Figure 5: ROOT versus UDR and ITM on single-reward single-pass ($K = 1$) training.

This subsection specifies the shared base configuration used by every experiment in this paper. Individual studies extend it in two places only: the matched-comparison alignment that defines what is held fixed and what is varied across methods (Appendix F.3), and the sequential multi-reward stage schedule (Appendix F.5, Table 4). Both build on the settings above and restate only their task-specific differences.

F.3 Evaluation Protocol and Baseline Alignment

Single-reward experiments report the reward measured on behavior-generated endpoints at each outer iteration. Curves show unsmoothed measurements and their trailing means. The reported convergence speed is the number of outer iterations required to reach the reference threshold used in Section 4, with the same convention applied to all methods.

All matched comparisons in this subsection build on the shared base configuration of Appendix F.2 and specify only the task-specific settings that are held fixed or varied across methods.

For matched single-reward comparisons, ROOT and DiffusionNFT share the backbone, scored endpoint batch, reward evaluator, optimizer settings, and sampling configuration. Multi-pass comparisons additionally match the number of newly scored endpoints and optimizer updates.

For the matched multi-pass comparison, repeated RootSolve and RootTrace reuse the same scored endpoints and empirical endpoint ratios. Their first-pass objectives coincide, while subsequent passes differ only in the velocity reference used by the training objective. The rollout, reward-evaluation, and optimizer-update budgets are held fixed.

For the UDR and ITM comparisons, the rollout-group size, endpoint-score construction, and update budget are matched. Relative to RootSolve, UDR changes only the displacement coefficient, $\widehat{\rho}(v - v_b) \to (v - v_b)$, while retaining the same normalized endpoint ratio. ITM instead retains the same calibrated finite-tilt field form but uses the unnormalized exponential coefficient $w_i = \exp(\tau A_i)$ in place of its normalized representative $\widehat{\rho}_i = w_i/(N^{-1}\sum_j w_j)$. The theoretical roles of these comparisons are summarized in Appendix E.3.

For sequential multi-reward evaluation, we follow the evaluation protocol of DiffusionNFT, including the evaluation prompt sets, sample counts, metric implementations, and aggregation procedure. All models are evaluated locally at 512×512 resolution in fp16 using a 40-step first-order ODE sampler. Reward-tuned models are sampled CFG-free. The ROOT result uses the final checkpoint after 1000 outer iterations. The DiffusionNFT result is our local evaluation of its official released multi-reward checkpoint, trained with its original five-stage schedule for approximately 1.7k iterations. The same local pipeline is used for both models, and the reported multi-reward results use a single evaluation seed.

F.4 Controlled Ablations

This subsection collects supplementary results omitted from the main text.

UDR and ITM comparisons. Figure 5 compares ROOT with UDR and ITM under the matched single-pass setting. Although UDR can reach similar reward levels on some tasks, its training trajectories exhibit larger fluctuations on HPSv2.1 and PickScore, while ROOT shows stronger late-training behavior. This comparison uses the same identified ratio in both methods, and their target-

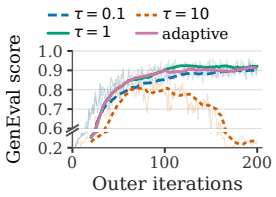


Figure 6: Tilt sensitivity.

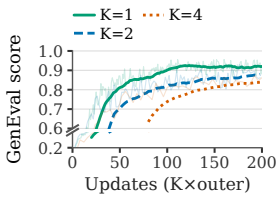


Figure 7: Reuse depth.

Stage	Training set	Active rewards	#Iter
S1	Pick-a-Pic	PickScore CLIPScore HPSv2.1	500
S2	GenEval	GenEval + S1	300
S3	OCR	OCR + S1	200

Table 4: Sequential multi-reward training schedule for ROOT.

flow relation is analyzed in Appendix E.2. ITM, which uses the unnormalized exponential coefficient $w_i = \exp(\tau A_i)$, rapidly degrades to low training rewards across all four objectives. This behavior is consistent with the representation sensitivity analyzed in Theorem 1 and Appendix E.2.

Tilt sensitivity. Appendix A.3 shows that affine score scaling changes the endpoint target through the effective tilt $\tau_{\text{eff}}(c) = \tau/d(c)$. Varying τ at fixed preprocessing therefore probes the same effective-tilt axis. We compare fixed $\tau \in \{0.1, 1, 10\}$ with an adaptive variant that sets the τ to the batch standard deviation of the endpoint rewards on GenEval, holding the remaining settings fixed. As shown in Figure 6, $\tau = 1$ provides a strong and stable fixed default. A smaller tilt leads to slower optimization, while an overly large tilt produces less stable late-stage training. The adaptive variant performs comparably to $\tau = 1$ over the training horizon. These results indicate that the main experiments do not rely on a narrowly tuned fixed tilt, and we use $\tau = 1$ for simplicity.

Reuse depth. We further vary the number of reuse passes $K \in \{1, 2, 4\}$ on GenEval. Figure 7 reports training reward against the total number of optimizer updates, measured as K times the number of outer iterations. Because each outer iteration scores one rollout batch and reuses it for K training passes, this comparison fixes the update budget while larger K uses proportionally fewer newly scored rollout batches. At matched update counts, $K = 1$ is the most update-efficient, while $K = 2$ retains most of its performance and $K = 4$ is consistently lower. This shows diminishing returns from deeper reuse of a fixed scored batch. We therefore use $K = 2$ as the main multi-pass setting, providing moderate rollout reuse without the larger loss in per-update efficiency observed at $K = 4$.

F.5 Sequential Multi-Reward Training and Evaluation

Training schedule. Building on the shared base configuration of Appendix F.2, we use the same reward definitions and implementations as DiffusionNFT, with a three-stage schedule totaling 1000 outer iterations. Within each stage, all active reward terms are equally weighted. PickScore is rescaled by $1/26$ before reward combination, while the other reward definitions follow DiffusionNFT. The complete schedule is summarized in Table 4.

The three stages use the same reward combinations as the corresponding DiffusionNFT training stages. In contrast, the released DiffusionNFT multi-reward checkpoint continues the sequential schedule for five stages and approximately 1.7k iterations. Our final ROOT checkpoint is obtained after S3 at 1k iterations. We report both task-specific metrics and DrawBench (Saharia et al., 2022) preference and image-quality metrics using the common local evaluation in Appendix F.3.

Qualitative comparison. Figures 8–11 present per-set qualitative comparisons for the SD3.5-Medium base model (with and without classifier-free guidance), our local re-evaluation of the released DiffusionNFT multi-reward model, and ROOT. Across compositional (GenEval), text-rendering (OCR), and general-preference (PickScore, DrawBench) prompts, both DiffusionNFT and ROOT yield cleaner layouts, more legible text, and higher-fidelity images than the base model, and are visually comparable to each other. All samples use the common evaluation pipeline in Appendix F.3.

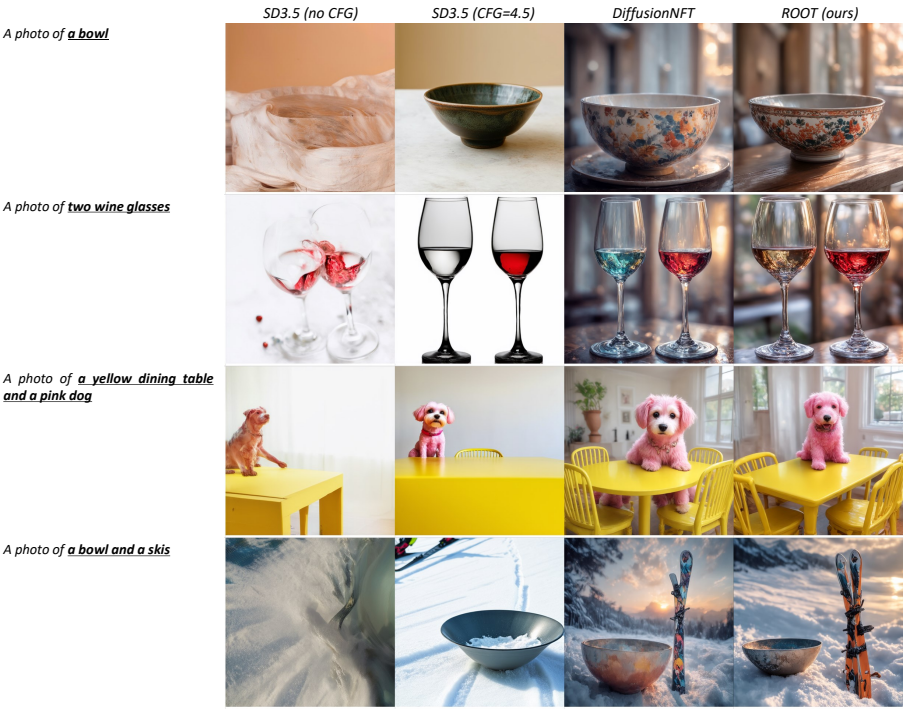


Figure 8: Qualitative comparison on GenEval prompts.

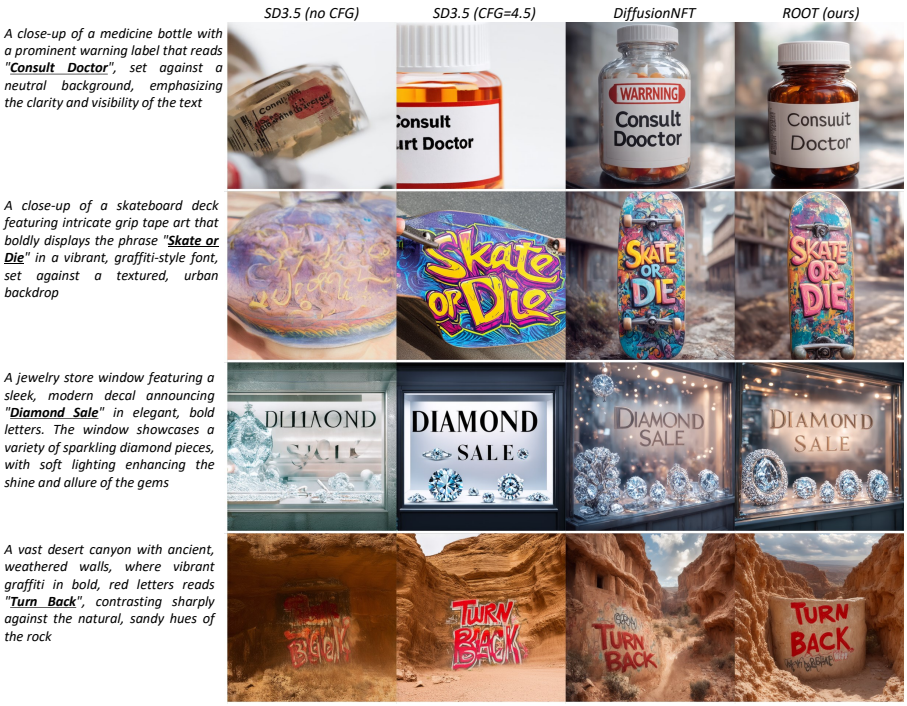


Figure 9: Qualitative comparison on OCR prompts.

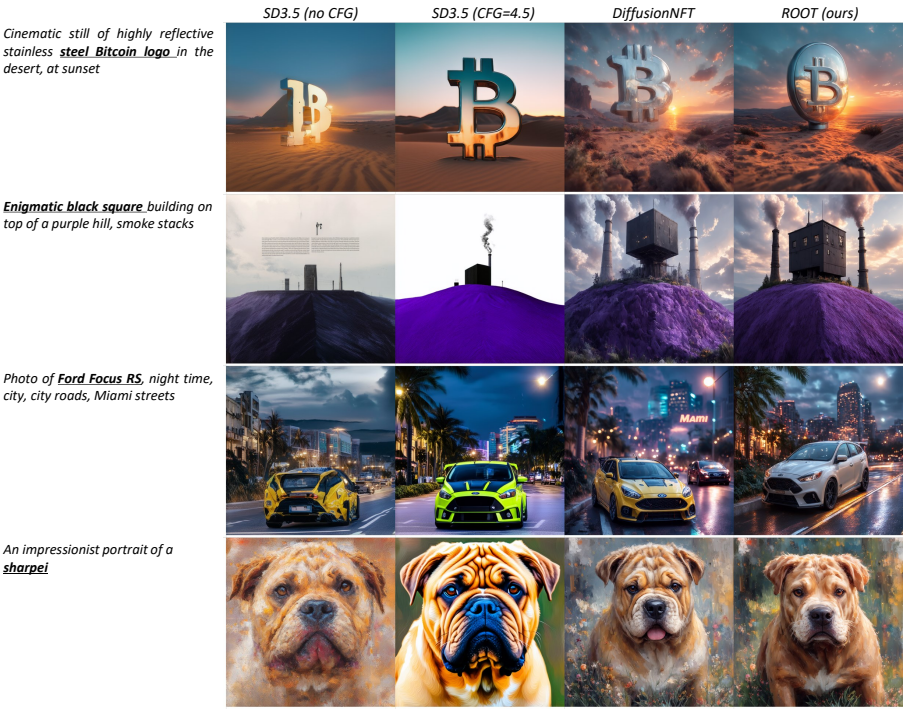


Figure 10: Qualitative comparison on PickScore prompts.

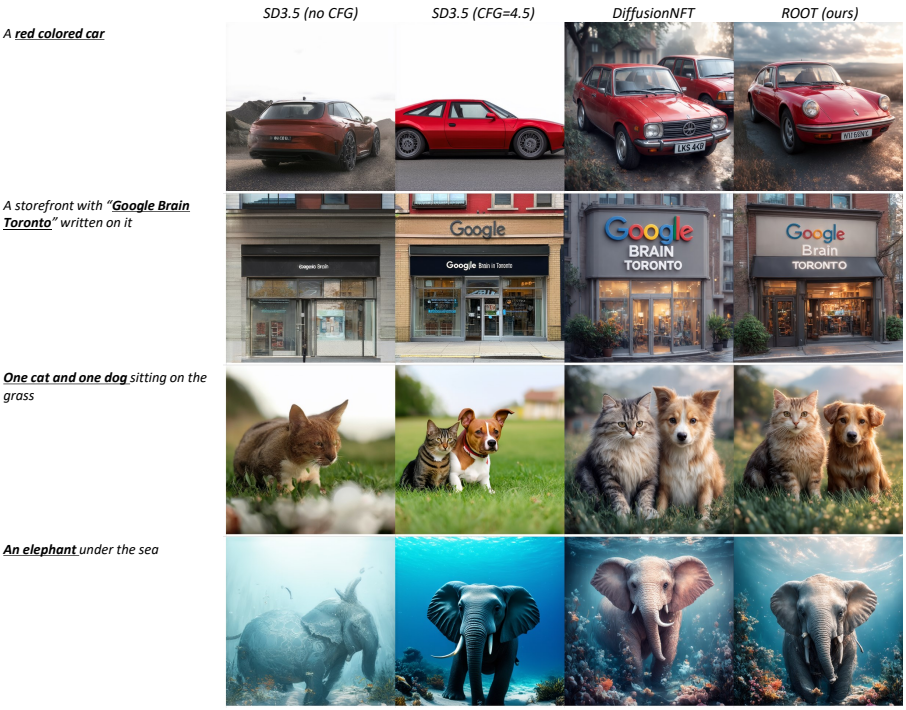


Figure 11: Qualitative comparison on DrawBench prompts.