RECAT: RECURRENT COMPOSABLE ACTIVATION TRANSITION FOR TEST-TIME SCALING OF DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion test-time scaling typically spends additional compute on generating, evaluating, or revising multiple candidates. We introduce ReCAT, a sequential scaling method that instead refines a single evolving internal state within one sampling trajectory. At a fixed intervention point, the frozen generator computes an intermediate activation once; a lightweight prompt- and state-conditioned controller then recurrently applies the same learned transition for a number of steps chosen at inference before generation resumes. Each additional refinement step requires only one controller call and no online reward queries. Reward-Ascent Matching (RAM) learns this transition from black-box rewards by fitting reward changes to activation displacements within sample groups and distilling the resulting secant directions across prompts. The resulting controller can be used directly; an optional second stage, Reward-Flow Calibration (RFC), further calibrates it on states reached by the controller's own updates using antithetic continuation probes. Neither stage backpropagates through the generator or reward model. Across three diffusion backbones and four target rewards, RAM alone yields consistent reward improvements, RFC provides additional gains, and reward gains increase with the number of refinement steps throughout the calibrated range. ReCAT also composes with search-based test-time scaling methods, improving Best-of-N search and gradient-based noise search. Code and checkpoints will be publicly available.

1 INTRODUCTION

Test-time scaling turns inference compute into better outputs, but the exchange rate depends on where that compute is spent. Simply increasing denoising steps refines the discretization of a fixed, reward-agnostic sampler and eventually saturates (Ma et al., 2025). Diffusion test-time scaling methods differ along two axes (Figure 1). The **allocation structure** asks whether added compute is spent across candidates or branches, or on dependent updates to one evolving state. The **completion count** asks whether returning one image requires several completed generation trajectories or only one. The axes are independent: parallel methods pay in population size, while prior sequential methods pay a completed generation per refinement round. Parallel strategies include Best-of-N (Xu et al., 2023; Kirstain et al., 2023), early pruning, which stays parallel while completing only the surviving trajectory (Guo et al., 2026; Huang et al., 2026; Yu et al., 2025), and adaptive population or tree search (Singhal et al., 2025; Jain et al., 2025; Ye et al., 2026). Sequential

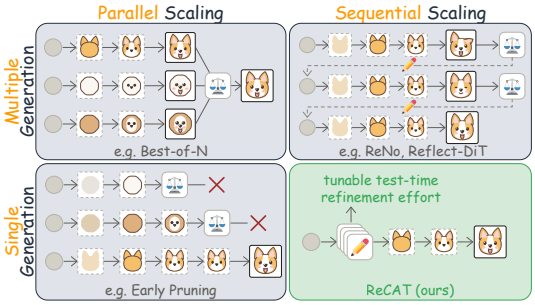

Figure 1: Test-time scaling methods on two axes: allocation structure, whether added compute is spent across candidates or on dependent updates to one evolving state, and completion count, whether returning one image requires several completed generation trajectories or only one. ReCAT occupies the sequential, single-generation cell: dependent refinement within a single sampling trajectory.

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.

methods include per-instance noise search and critique-guided revision (Eyring et al., 2024; Li et al., 2025; Zhuo et al., 2025). We introduce ReCAT for **sequential scaling within a single sampling trajectory**: dependent refinement of one persistent internal state before generation completes.



Figure 2: **Qualitative effects of ReCAT on a single sampling trajectory.** Each pair shows the frozen generator's output (left) and the ReCAT-refined output (right), illustrating how recurrent internal updates can change composition, rendering, and local detail toward better results.

This regime matters because it changes what an additional refinement step costs. Selection-based early pruning examines more candidates per unit compute (Guo et al., 2026; Huang et al., 2026; Yu et al., 2025), but with N candidates from a fixed proposal, even a perfect probe is bounded by the best of those N candidates. The expected reward of this oracle has diminishing gains as N grows (Proposition 1); pruning makes the same selection frontier cheaper to explore. Improving the candidates themselves offers a complementary route. Adaptive population methods and per-instance noise search change proposals, but construct reward-directed updates online (Singhal et al., 2025; Jain et al., 2025; Ye et al., 2026; Eyring et al., 2024; Ma et al., 2025; Mao et al., 2025). Critique-and-regenerate methods learn reusable revision models, but each revision still requires another completed generation and online feedback (Li et al., 2025; Zhuo et al., 2025). ReCAT instead amortizes an internal update rule. For N forward-generated candidates, K refinement steps per candidate add NK lightweight controller calls, without additional denoiser evaluations or reward queries inside the refinement loop. For selection, ReCAT changes the candidate distribution before ranking rather than merely examining more draws from a fixed proposal. Its internal refinement depth can be chosen separately from the outer candidate or optimization budget, allowing composition with Best-of-N random search and gradient-based noise search.

ReCAT. We implement this update rule with a compact prompt- and state-conditioned controller operating at a fixed denoising step and block of a frozen generator. The intermediate activation provides prompt-conditioned information predictive of final quality (Guo et al., 2026), while leaving substantial frozen computation downstream through which an update can influence the image. With the diffusion timestep held fixed, the same controller repeatedly reads the current activation and prompt and predicts an update direction, and the updated state feeds the next step (Figure 3).

Language models provide a useful precedent for this form of sequential computation. Explicit reasoning effort chosen at inference buys quality through longer reasoning traces (Guo et al., 2025; Muennighoff et al., 2025; Snell et al., 2024), while recurrent-depth language models repeatedly apply a weight-tied block to an evolving internal state, with the iteration count chosen at inference (Geiping et al., 2026; Yang et al., 2026). This separates learning an update rule from choosing how much computation to spend applying it. A frozen diffusion sampler does not natively provide the reward-directed activation transition required by such an inner loop. ReCAT therefore learns this transition from black-box rewards, while leaving its number of applications to be chosen at inference.

Learning this inner transition from scalar, black-box rewards poses two challenges. First, reward observations must be converted into update directions in a million-dimensional activation space without differentiating through the generator, and the resulting rule must generalize across prompts. Second, recurrent application changes the transition's input distribution, so the learned directions must remain useful on states produced by its own updates.

RAM learns a deployable controller from black-box rewards; RFC provides optional further calibration. **Reward-Ascent Matching (RAM)** constructs secant teachers by regressing rewards on activation displacements within each sample group, then distills these directions into the controller. RAM alone supports effective recurrent refinement (Table 3). **Reward-Flow Calibration (RFC)** starts from the RAM checkpoint and calibrates the controller on states reached by its own unrolls. At each visited state, RFC scores two continuations from symmetric perturbations along the predicted direction and uses their reward gap to supervise whether that direction should be retained or reversed. Both stages train only the controller, without differentiating through the generator or reward model. Our contributions are:

- **Sequential scaling within a single sampling trajectory.** A lightweight, weight-tied activation transition provides an inference-time refinement axis inside a frozen generation, without online reward queries or additional denoiser evaluations per step. Its refinement effort is separately adjustable from outer search, and composition improves both Best-of-N random search and gradient-based noise search.
- **Reward-to-action learning with RAM.** RAM uses reward differences and activation geometry within sample groups to construct secant-direction supervision, then distills these directions into a controller that is effective on its own. Supervision-target and intervention-state controls establish the value of this construction.
- **On-policy calibration with RFC.** RFC uses antithetic continuation probes to further train the controller on its own induced states. At the same refinement effort, it improves all twelve backbone–reward combinations.

2 Related Work

Diffusion refinement methods use several forms of feedback. Selection and population search compare candidates using scalar reward scores or learned estimates (Kirstain et al., 2023; Xu et al., 2023; Guo et al., 2026; Huang et al., 2026; Bucciarelli et al., 2026; Singhal et al., 2025; Jain et al., 2025; Ye et al., 2026). Per-instance noise optimization turns reward gradients or black-box perturbation measurements into update directions (Eyring et al., 2024; Ma et al., 2025; Mao et al., 2025). Critique-and-regenerate models use textual feedback about completed images to guide a new generation (Li et al., 2025; Zhuo et al., 2025).

Amortization moves the construction of an update into training. Several methods learn an inexpensive initial-noise transformation applied once before generation (Zhou et al., 2025; Ahn et al., 2026), such as Noise Hypernet's amortization of per-instance search (Eyring et al., 2024; 2026). Fine-tuning also transfers reward information into the generator's parameters (Black et al., 2024; Fan et al., 2023; Wallace et al., 2024; Liu et al., 2026).

Intermediate activations offer a state on which a learned update can operate during generation. Quality probes show that these states contain information predictive of the final output (Kim et al., 2025; Qu et al., 2026; Zhang et al., 2025). Representation-level interventions show that editing attention, embeddings, or hidden activations can influence image content (Hertz et al., 2023; Mokady et al., 2023; Brack et al., 2023; Rodriguez et al., 2025); steering vectors similarly alter language-model

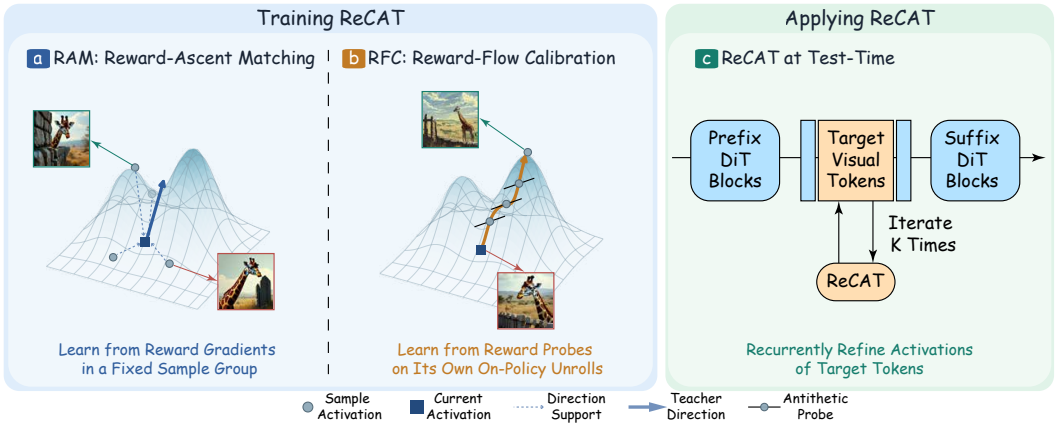

Figure 3: **ReCAT overview.** Surface height depicts the value V_c over activations of one prompt; thumbnails show decoded images. (a) RAM: gray dots form a sample group, the activations of one prompt under S seeds. For a source activation from the group (square), a locality-weighted regression of reward gaps on the displacements to the other samples (dashed lines) estimates the reward gradient within their span without backpropagation; the direction of this estimate is the secant teacher (blue arrow), which ReCAT learns to predict. (b) RFC: ReCAT unrolls its own transition (orange); at each visited state, an antithetic probe decodes and scores the pair $a \pm h\,\widehat{g}_\theta$ along the predicted direction, and the sign of the reward gap keeps or reverses that direction. (c) At inference, the frozen prefix computes the activation once, ReCAT refines it recurrently, and the frozen suffix completes the remaining blocks and denoising steps to produce one image, with no reward call.

outputs (Li et al., 2023; Turner et al., 2023; Zou et al., 2023). ReCAT learns a reward-directed update on this internal state from black-box feedback and reuses it recursively, making the number of controller applications an inference-time refinement choice. Section F gives the detailed comparison along the two axes of Figure 1.

3 Method

ReCAT learns a reusable transition on an intermediate generation state (Figure 3). At inference, the frozen image generator runs its prefix once to produce an intermediate state, ReCAT updates that state by recurrently applying the controller-defined transition, and the frozen continuation runs once from the final state. The ReCAT controller g_θ is a compact prompt- and state-conditioned network. Its normalized output defines the reusable transition $\mathcal{T}_\theta$. Applied repeatedly to its own output, this transition drives a persistent state toward higher-reward regions over a calibrated refinement horizon before the generator completes its trajectory.

RAM constructs secant teachers from grouped activations and rewards, and distills them into the controller (Section 3.2). RFC further fits the controller on states reached by its own unrolls, using reward probes to label its predicted directions (Section 3.3). Both stages train from black-box forward evaluations. Inference integrates the frozen learned direction field via explicit Euler steps (Section 3.4). Algorithm 1 states the whole procedure and Section D the implementation defaults.

3.1 Problem Setup

We operate on a pretrained text-to-image diffusion or flow model (Ho et al., 2020; Song et al., 2021; Rombach et al., 2022) with frozen weights. The construction assumes only a conditional generator with an accessible intermediate state and a reward on the completed output. Let the frozen generator G map a conditioning input c and noise $z \sim \mathcal{N}(0, I)$ to image $x = G(z, c)$ with reward $r(z, c) = R(G(z, c), c)$. In our experiments, c is a text prompt. ReCAT acts on the activation $a = A_{t,\ell}(z, c)$ of an image-stream block. This activation is an array over image tokens with p total coordinates and an induced measure $\mu_c = (A_{t,\ell})_\sharp \mathcal{N}(0, I)$. Such intermediate states carry prompt-conditioned evidence while leaving a substantial frozen suffix for the intervention to act through; Table 2 compares the three states a frozen sampler exposes.

The frozen suffix $G_{>(t,\ell)}$ completes the trajectory from a, reusing its remaining state (such as the latent, the text stream, and the solver history) unchanged; we write $V_c(a) = R(G_{>(t,\ell)}(a, c), c)$ for the early-state value, with ∇V_c taken with respect to a.

All inner products and norms below use the per-coordinate geometry

$$\langle u, v\rangle_a = p^{-1}\,\mathrm{vec}(u)^\top\,\mathrm{vec}(v), \qquad \|u\|_a = \sqrt{\langle u, u\rangle_a} = \mathrm{rms}(u). \tag{1}$$

Here vec flattens the token array, and the isometric coordinates $\widetilde{u} = p^{-1/2}\,\mathrm{vec}(u)$ reduce Equation (1) to the standard dot product, $\langle u, v\rangle_a = \widetilde{u}^\top\widetilde{v}$. Matrix expressions below use these coordinates. Accordingly, $\nabla V_c(a)$ represents the differential of V_c in this geometry: it is the unique vector satisfying $\langle \nabla V_c(a), u\rangle_a = \mathrm{d}V_c(a)[u]$ for any displacement u, making it p times the Euclidean gradient. Under this geometry, a unit direction always changes the activation by the same per-coordinate RMS, independent of token count or network width.

A sample group contains S samples generated with the same conditioning input c and independent seeds $z_i \sim \mathcal{N}(0, I)$: $\{(a_i, r_i)\}_{i=1}^S$, where $a_i = A_{t,\ell}(z_i, c)$ and $r_i = r(z_i, c)$. Applying the shared transition recursively creates state feedback: each refinement step acts on the current state, and its output becomes the input to the next.

3.2 Reward-Ascent Matching from Grouped Samples

Within each sample group, RAM estimates the reward gradient restricted to the group's secant span from reward and activation differences, then distills these secant teachers into the controller.

Under a pointwise trust region $\|u(a)\|_a \le \epsilon(a)$ in the activation geometry, the first-order optimal displacement is

$$u^\star_\epsilon(a) = \epsilon(a)\frac{\nabla V_c(a)}{\|\nabla V_c(a)\|_a}. \tag{2}$$

For a refinement step, ReCAT sets $\epsilon(a) = \Delta s\ \mathrm{rms}(a)$, making Δs a relative step size: the state moves by that fraction of its own RMS. This direction also links refinement to selection. Ranking N draws reweights the fixed proposal by an order-statistic tilt (Section G), whereas a refinement step transports the proposal itself, $\rho_c^{(k+1)} = (\mathcal{T}_\theta)_\sharp \rho_c^{(k)}$. Both share a local direction: exponentially tilting the current law toward reward yields a density-ratio score proportional to ∇V_c (Equation (14) in Section H). The reward is a black box, so RAM approximates this direction from forward evaluations alone.

Constructing secant teachers. Within one sample group, for source a_i, let $d_{ij} = a_j - a_i$, set the bandwidth $h_i = \mathrm{med}_k\|d_{ik}\|_a$, and define locality weights $w_{ij} = \exp(-\|d_{ij}\|_a/h_i)$. We fit a locally weighted linear regression of reward on displacement with an unpenalized intercept, centering both sides around their locality-weighted means. Centering only the reward would spuriously fold the intercept into the slope (see Section I). Let $\widetilde{D}_i$ collect the centered displacements $\widetilde{d}_{ij}$ as rows, $\widetilde{y}_i$ collect the centered rewards, and $W_i = \mathrm{diag}(w_{ij})$. With a ridge $\rho > 0$ (Section I), the solution is

$$g_c^\star(a_i) = \left(\widetilde{D}_i^\top W_i^2 \widetilde{D}_i + \rho I\right)^{-1} \widetilde{D}_i^\top W_i^2 \widetilde{y}_i. \tag{3}$$

When V_c is affine on the sampled neighborhood and rewards equal $V_c(a_j)$, this solution recovers the projected gradient $P_{\mathcal{U}_i}\nabla V_c(a_i)$, where $\mathcal{U}_i = \mathrm{span}\{\widetilde{d}_{ij}\}$, up to a ridge shrinkage that vanishes as $\rho \to 0$ (Proposition 3). Lemma 4 characterizes how the sampled geometry amplifies Taylor remainders and reward observation errors.

With p in the millions, the $p \times p$ covariance in Equation (3) cannot be formed explicitly. The dual form gives the same solution through the $S \times S$ Gram matrix of centered displacements. Because the target lies in $\mathcal{U}_i$, S dual coefficients per state reconstruct it exactly (Section I). Each nonzero teacher is then normalized, $\widehat{g}_c^\star = g_c^\star/\|g_c^\star\|_a$, and $\mathcal{I}_c$ collects the sources retained.

Matching the field to its teachers. Teachers exist only at sampled states, so the controller $g_\theta(a, c)$ amortizes them across states and prompts by regressing onto these unit-RMS targets,

$$\mathcal{L}_{\mathrm{RAM}}(\theta) = \mathbb{E}_c \frac{1}{|\mathcal{I}_c|}\sum_{i\in\mathcal{I}_c} \left\|g_\theta(a_i, c) - \widehat{g}_c^\star(a_i)\right\|_a^2, \qquad \left\|\widehat{g}_c^\star(a_i)\right\|_a = 1. \tag{4}$$

The controller is a small ConvNeXt-style network that reads the prompt via a single cross-attention layer. One set of parameters serves every refinement step, so the unroll is a weight-tied recurrence (see Section D). g_θ supplies only the direction. Normalizing directions to unit RMS separates direction learning from the transition's step length. The transition applied at inference is

$$\mathcal{T}_\theta(a, c, \Delta s) = a + \Delta s \ \mathrm{rms}(a)\, \widehat{g}_\theta(a, c), \qquad \widehat{g}_\theta = \frac{g_\theta}{\mathrm{rms}(g_\theta)}. \tag{5}$$

This uses the radius of Equation (2) and the controller's learned direction. To first order, the value gain from one refinement step scales with the cosine between $\widehat{g}_\theta$ and the unit ascent direction, so direction error bounds the lost gain (Proposition 6, Section K).

Choice of intervened state. A frozen sampler exposes three intervention states: the initial noise, a latent at a chosen step, or a block activation inside that step. Block activations expose a prompt-conditioned representation of the current sample to the controller and leave a frozen suffix through which an edit can influence the output. The initial noise is independent of the prompt, so a controller acting there must obtain prompt information through its own conditioning. Table 2 compares the same RAM construction across these states with a single update and identical supervision cost.

RFC uses this same transition over K_{train}-step calibration unrolls with step size Δs.

3.3 Reward-Flow Calibration on ReCAT Unrolls

RAM learns the field on activations sampled from the frozen generator, but recurrent application evaluates it on states produced by its own updates. RFC therefore calibrates the RAM controller along its K_{train}-step unrolls, using black-box reward feedback to determine whether each predicted direction remains aligned with reward ascent.

Measuring reward change along the field. Two decoded continuations provide this signal; their antithetic placement lets it survive curvature and measurement error. For a state a visited during a calibration unroll, we set $h(a) = \Delta s\, \mathrm{rms}(a)$ and form the antithetic pair $a^\pm = a \pm h(a)\widehat{g}_\theta(a, c)$. Both are decoded through the frozen suffix without differentiation and scored as $r^\pm = R(G_{>(t,\ell)}(a^\pm, c), c)$. The unroll always advances through a^+; a^- acts as a measurement branch. The probe radius matches the unroll's trust-region radius, comparing continuations at the scale of one training transition.

Symmetric probes cancel the state's own value and every even-order term,

$$\frac{r^+ - r^-}{2h(a)} = \left\langle \nabla V_c(a), \widehat{g}_\theta(a, c) \right\rangle_a + O(h(a)^2). \tag{6}$$

Black-box queries alone therefore reveal whether the controller's predicted direction points uphill or downhill. By Proposition 5, this sign is reliable when the directional signal clears a floor set by curvature and by error in the measured gap. We map the gap to a bounded label $\chi(a) = \tanh\left((r^+ - r^-)/(\kappa\sigma_R)\right) \in (-1, 1)$, where σ_R is the across-seed reward standard deviation averaged over prompts and κ sets the temperature. Its sign records the measured orientation, while its magnitude downweights small gaps and saturates for large ones.

Because the probe reads reward only along $\pm\widehat{g}_\theta$, it is informative only when that direction already carries a measurable reward projection. A random unit direction in p dimensions has an expected squared alignment of $1/p$ with a fixed unit gradient direction, so the root-mean-square directional signal scales as $p^{-1/2}$ at fixed gradient norm and can fall below that floor. RFC starts from the learned directions of the RAM checkpoint, then measures their reward alignment on visited states (Table 2). The probed direction need not lie in the secant span of any RAM sample group.

On-policy calibration. Following on-policy data aggregation (Ross et al., 2011), RFC alternates between sampling the occupancy induced by the current controller and fitting the field to reward-labeled directions on those states. Let $g_{\bar\theta}$ denote a frozen collection copy of the controller and $\mathcal{D}_{\bar\theta}$ the resulting labeled states. Conditioned on this collection, the fitted controller minimizes

$$\mathcal{L}_{\mathrm{RFC}}(\theta; \bar\theta) = \frac{\mathbb{E}_{(a,c,\chi)\sim\mathcal{D}_{\bar\theta}} |\chi| \left\| g_\theta(a, c) - \mathrm{sign}(\chi)\, \widehat{g}_{\bar\theta}(a, c) \right\|_a^2}{\mathbb{E}_{(a,c,\chi)\sim\mathcal{D}_{\bar\theta}} |\chi|}. \tag{7}$$

Using the updated controller for subsequent collection refreshes the states and targets for on-policy self-distillation. Section J details the batched realization and RAM replay anchor.

The pointwise form clarifies the supervision supplied at one state. Hold (a, c) fixed, write $v = \widehat{g}_{\bar{\theta}}(a, c)$, and treat the label at that state as random under the collection process, with $\mathbb{E}[|\chi| \mid a, c] > 0$. Since $|\chi| \operatorname{sign}(\chi) = \chi$, the minimizer of Equation (7) at that state is

$$g^\star(a, c) = \frac{\mathbb{E}[\chi \mid a, c]}{\mathbb{E}[|\chi| \mid a, c]}\, v. \tag{8}$$

The coefficient records the net probe agreement with the collection direction; for a nonzero coefficient, deployment normalization retains only its sign. Across an on-policy batch, the collection direction v varies with the state. RFC jointly fits one shared controller to these signed vector targets at states sampled from the current on-policy occupancy, then collects the next round under the updated controller. Each probe therefore supplies a scalar orientation label, while parameter sharing couples these labels across sampled states and rounds.

3.4 Inference: Sequential Scaling within a Single Sampling Trajectory

At inference, the controller parameters are frozen. The generator prefix produces $a^{(0)}$ once, and ReCAT applies

$$a^{(k+1)} = \mathcal{T}_\theta(a^{(k)}, c, \Delta s), \quad k = 0, \ldots, K-1, \qquad \tau_K := K\,\Delta s, \tag{9}$$

with the frozen suffix decoding only the final state $a^{(K)}$. Each application of $\mathcal{T}_\theta$ is one refinement step, costing one controller call. The number of applications K is the refinement effort ($K = 0$ is the frozen baseline). Because the same parameters act at every step, K sets the serial depth of the computation behind one image. Inference uses the calibrated step size Δs, so a larger K carries the state farther along the learned field over the total integration horizon $\tau_K = K\,\Delta s$, where $\eta = K_{\text{train}}\Delta s$ is the horizon for which ReCAT was calibrated ($\Delta s = \eta / K_{\text{train}}$). Section A.2 sweeps K on all three backbones.

Refinement effort and integration horizon. Equation (9) is the explicit Euler discretization of the learned flow $\mathrm{d}a/\mathrm{d}s = F_\theta(a, c) := \mathrm{rms}(a)\widehat{g}_\theta(a, c)$, integrated while the sampler's clock is paused. Along this transported measure, the expected value changes at the rate of the Lie derivative $\mathcal{L}_{F_\theta} V_c := \langle \nabla V_c, F_\theta \rangle_a$ (Equation (23) in Section K). The centered difference in Equation (6), multiplied by $\mathrm{rms}(a)$, estimates this directional value change. RFC uses these measurements to calibrate the direction along the flow that inference integrates. Under L-smoothness, each refinement step gains value when its true alignment term exceeds the $O(\Delta s^2)$ curvature penalty (Proposition 7).

4 Experiments

4.1 Protocol and Baselines

Setup. Prompts are drawn from MS-COCO captions (Lin et al., 2014), with a held-out test set of 500 prompts. We evaluate SANA-1.5, SD3.5-Medium, and FLUX.2-klein under an identical protocol. Training rewards are ImageReward (Xu et al., 2023), PickScore (Kirstain et al., 2023), HPSv2 (Wu et al., 2023), and aesthetic score (Schuhmann, 2022). Training is run independently for each backbone and target reward; reward-guided baselines use the same target. Hyperparameters are selected on a disjoint validation set. On the test set, we report paired per-prompt gains (Section C). Primary ReCAT results use the calibrated endpoint $K = K_{\text{train}} = 30$.

Baselines. We compare the frozen generator against Best-of-N, Probe-Select, Zero-Order Search over initial noise, per-instance gradient noise optimization (ReNO), and Noise Hypernet noise modulation. Training and evaluation settings for learned baselines are in Section C. Throughout, N denotes a method's candidate or iterate count. In Table 1, selection and search baselines use about two generator trajectories of forward compute. Probe-Select screens six candidates after the first 20% of the trajectory and completes one. We run ReNO through each backbone's full sampling chain and return the highest-reward iterate. We train Noise Hypernet through each backbone's full sampling chain. Critique-and-regenerate methods incur a distinct outer-loop cost and are compared by their training and inference requirements in Table 8.

Table 1: **Test-time scaling across diffusion backbones.** Paired per-prompt gains Δ/σ, normalized by the frozen generator's across-seed reward standard deviation (raw units in Table 7). Rows above the thin rule use one generation per image, rows below about two generations of forward compute. The last column is the added single-image inference time on an Nvidia A10G; ReNO's is estimated from measured forward and backward times. † marks training collapse of Noise Hypernet.

Method	ImageReward	Aesthetic	HPSv2	PickScore	Infer. Time
SANA-1.5 (Linear-attention Multi-step DiT)					4.014 s
Noise Hypernet (Eyring et al., 2026)	−0.005	+0.289	+0.241	+0.171	+173 ms
ReCAT	+0.263	+1.393	+0.783	+0.442	+78 ms
Best-of-N (N=2)	+0.503	+0.662	+0.564	+0.566	+4.09 s
ReNO (N=2 rollouts) (Eyring et al., 2024)	+0.323	+0.341	+0.392	+0.428	+14.16 s
Zero-Order Search (Ma et al., 2025)	+0.327	+0.364	+0.359	+0.426	+4.09 s
Probe-Select (N=6) (Guo et al., 2026)	+0.244	+0.527	+0.445	+0.380	+4.02 s
ReCAT + Best-of-2	+0.739	+1.795	+1.212	+0.944	+4.24 s
SD3.5-Medium (Standard Multi-step MMDiT)					15.056 s
Noise Hypernet (Eyring et al., 2026)	−1.579	−0.295	−0.083	−1.163	+587 ms
ReCAT	+0.116	+0.910	+0.499	+0.295	+340 ms
Best-of-N (N=2)	+0.516	+0.350	+0.499	+0.454	+15.13 s
ReNO (N=2 rollouts) (Eyring et al., 2024)	+0.312	+0.395	+0.432	+0.434	+54.73 s
Zero-Order Search (Ma et al., 2025)	+0.399	+0.414	+0.432	+0.373	+15.13 s
Probe-Select (N=6) (Guo et al., 2026)	+0.179	+0.217	+0.393	+0.209	+15.06 s
ReCAT + Best-of-2	+0.597	+1.331	+0.962	+0.748	+15.81 s
FLUX.2-klein (Distilled Few-step MMDiT)					3.346 s
Noise Hypernet (Eyring et al., 2026)	+0.025	†	+1.462	+0.694	+831 ms
ReCAT	+0.308	+1.353	+0.520	+0.475	+385 ms
Best-of-N (N=2)	+0.521	+0.452	+0.493	+0.589	+3.42 s
ReNO (N=2 rollouts) (Eyring et al., 2024)	+0.347	+0.290	+0.329	+0.353	+10.22 s
Zero-Order Search (Ma et al., 2025)	+0.374	+0.327	+0.301	+0.314	+3.42 s
Probe-Select (N=6) (Guo et al., 2026)	+0.085	+0.311	+0.252	+0.320	+3.35 s
ReCAT + Best-of-2	+0.832	+1.727	+0.935	+0.999	+4.19 s

4.2 Results

ReCAT alone improves all twelve backbone–reward pairs over the frozen generator (Table 1), and on every pair its gains grow with refinement effort up to the calibrated effort (Figure 5).

Against Best-of-N (N=2), which spends a second generation, refinement alone trails on ImageReward and PickScore. Refining every candidate before the same selection surpasses the unrefined selection across all twelve pairs, and ReCAT+Best-of-N stays above Best-of-N across the sweep, out to a search budget of 48 generations (Figure 4).

Adding ReCAT to ReNO also improves every measured search budget in Figure 4. The ReNO result shows that the same internal refinement can be used inside an adaptive optimizer. On SANA-1.5, thirty refinement steps add 78 ms to a 4.01 s generation, a small fraction of the second generation that Best-of-N already pays. Unlike terminal selection, refinement moves every draw and leaves within-prompt diversity undiminished (Section B.3). The learned transition also transfers beyond its training distribution, retaining 89% of its SANA-1.5 gain on DrawBench prompts (Section B.2).

Table 2: **Ablation on SANA-1.5.** Paired normalized gains Δ/σ over the frozen generator. RAM controls use one update; the other blocks use calibrated controllers. Definitions are in Section B.5.

Config	IR.	Aes.	HPS.	Pick.
RAM	+0.22	+1.00	+0.59	+0.32
supervision target				
Group best	−1.24	−2.00	−1.62	−2.39
Reward-weighted mean	−1.49	−2.03	−1.90	−2.60
Entropic OT barycenter	+0.07	+0.01	+0.08	+0.05
intervention state				
Initial noise z	+0.04	+0.50	+0.46	+0.27
Sampler latent	+0.03	+0.62	+0.30	+0.16
RFC	+0.26	+1.39	+0.78	+0.44
initialization method				
Random initialization	+0.04	−0.08	+0.07	+0.01
exploration strategy				
Random transverse	+0.17	+0.94	+0.57	+0.30
Principal transverse	+0.18	+1.01	+0.56	+0.29
Inference				
w/o recurrence	+0.14	+1.22	+0.72	+0.39

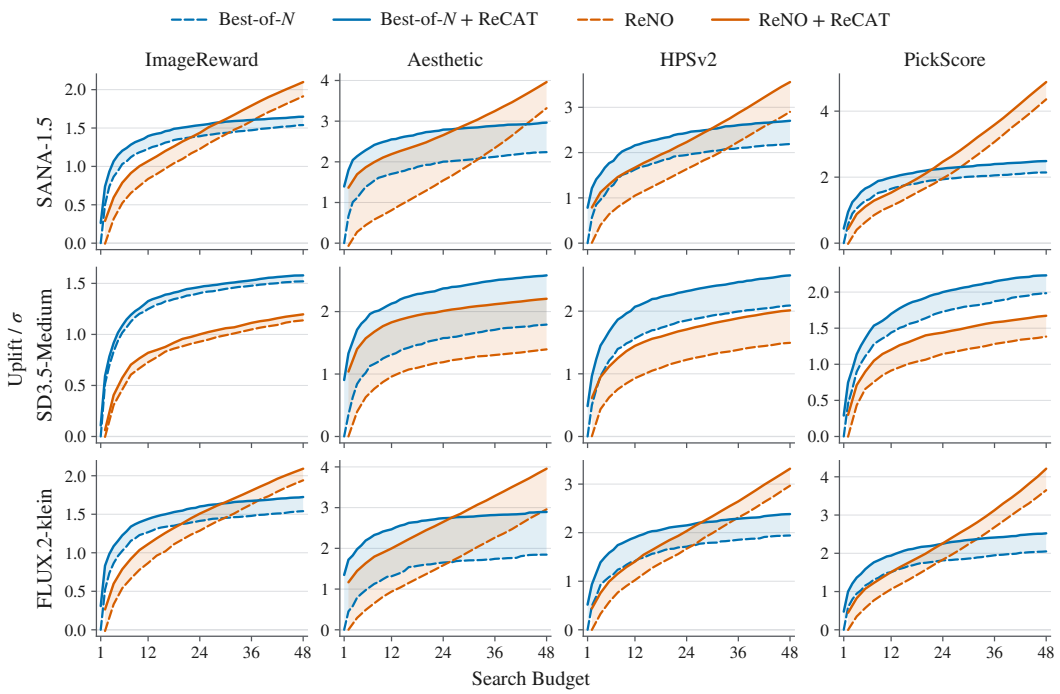

Figure 4: **ReCAT composes with both outer loops across the search budget.** Paired normalized gain Δ/σ over the frozen generator, against generation-equivalent search budget: one unit per Best-of-N candidate and two per ReNO iterate for generation and reward-gradient backpropagation. Shaded bands mark the gain ReCAT adds to each outer loop; see Table 4 for more comparisons.

Table 2 shows that RAM alone learns an effective update and compares the supervision targets and intervention states behind it. All four supervision targets use variation within each sample group, but they extract different signals. Group-best and reward-weighted mean point toward high-reward samples, while the entropic OT barycenter predicts a reward-weighted endpoint. RAM instead fits a signed combination of activation displacements to reward *changes*, so low-reward neighbors can contribute directions away from them. The OT control yields positive gains on all four rewards, but well below RAM's. The same RAM construction improves all four rewards in each of the three spaces, with block activations yielding the largest gains at identical supervision cost. We therefore use block activations for RFC and all primary experiments.

The RAM checkpoint improves all twelve backbone–reward pairs under recurrent application, and RFC further increases every reported gain at the same refinement effort (Table 3). Before RFC, recurrence is weaker than a single application on three pairs; after RFC, it is stronger on all twelve. Initializing RFC from a random controller yields limited gains. The transverse controls use the same probe budget to estimate a correction to $\widehat{g}_\theta$, yet gain less than the antithetic pair (Table 2).

5 Conclusion

A frozen diffusion generator can be improved from inside the single trajectory it is already running. By learning a reusable transition on intermediate activations and recursively applying it, ReCAT turns the number of refinement steps into an inference-time control over output quality. Reward-Ascent Matching learns effective updates by distilling secant teachers from grouped activations and black-box rewards. Reward-Flow Calibration further improves these updates using reward feedback on states visited by their own unrolls. Across three backbones and four rewards, gains rise with refinement effort, and refining every candidate before selection improves Best-of-N as well. Future work could explore alternative RAM teacher constructions and RFC on-policy calibration targets within the ReCAT framework, and extend ReCAT to editing and other generative tasks.

ReCAT amortizes the *rule* for spending test-time compute rather than the test-time budget itself. Once that rule is learned, the next unit of inference compute need not buy another sample; it can buy another informed refinement step on the one already under way.

AI USE STATEMENT

In this work, we used generative AI tools to aid and polish writing, assist with retrieval and discovery of related work, help develop proofs of mathematical claims, and assist with code writing and debugging. We did not use generative AI tools for other tasks requiring disclosure, and the remaining disclosure categories are not applicable. We reviewed and verified all AI-assisted text, references, mathematical arguments, and code. We take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

The method and training procedure are specified in the main text and summarized in Algorithm 1. Sections C and D document the implementation defaults, model configurations, prompt splits, baseline settings, and evaluation protocol; the remaining appendices provide theoretical details and additional controls. Code and model checkpoints will be publicly available.

REFERENCES

Donghoon Ahn, Jiwon Kang, Sanghyun Lee, Jaewon Min, Minjae Kim, Wooseok Jang, Hyoungwon Cho, Sayak Paul, SeonHwa Kim, Eunju Cha, Kyong Hwan Jin, and Seungryong Kim. A noise is worth diffusion guidance. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=xEWooSOgaz.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations*, volume 2024, pp. 4965–4987, 2024.

Manuel Brack, Felix Friedrich, Dominik Hintersdorf, Lukas Struppek, Patrick Schramowski, and Kristian Kersting. Sega: Instructing text-to-image models using semantic guidance. *Advances in Neural Information Processing Systems*, 36:25365–25389, 2023.

Davide Bucciarelli, Evelyn Turri, Lorenzo Baraldi, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Tiny inference-time scaling with latent verifiers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 2873–2882, June 2026.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

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

Luca Eyring, Shyamgopal Karthik, Alexey Dosovitskiy, Nataniel Ruiz, and Zeynep Akata. Noise hypernetworks: Amortizing test-time compute in diffusion models. *Advances in Neural Information Processing Systems*, 38:95429–95462, 2026.

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. *Advances in neural information processing systems*, 36:79858–79885, 2023.

Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. Erasing concepts from diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2426–2436, 2023.

Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. Scaling up test-time compute with latent reasoning: A recurrent depth approach. *Advances in Neural Information Processing Systems*, 38: 41340–41391, 2026.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Huanlei Guo, Hongxin Wei, and Bingyi Jing. Toward early quality assessment of text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 38410–38419, June 2026.

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

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

Bukun Huang, Benlei Cui, Zhizeng Ye, Xuemei Dong, Tuo Chen, Hui Xue, Dingkang Yang, Longtao Huang, Haiwen Hong, and Jingqun Tang. Diffusion probe: Generated image result prediction using cnn probes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 35926–35935, June 2026.

Vineet Jain, Kusha Sareen, Mohammad Pedramfar, and Siamak Ravanbakhsh. Diffusion tree sampling: Scalable inference-time alignment of diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=3D88hCO0Gd.

Richard Jordan, David Kinderlehrer, and Felix Otto. The variational formulation of the fokker–planck equation. *SIAM journal on mathematical analysis*, 29(1):1–17, 1998.

Joowon Kim, Ziseok Lee, Donghyeon Cho, Sanghyun Jo, Yeonsung Jung, Kyungsu Kim, and Eunho Yang. Early timestep zero-shot candidate selection for instruction-guided image editing. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18844–18854. IEEE, 2025.

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. *Advances in neural information processing systems*, 36:36652–36663, 2023.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in neural information processing systems*, 36:41451–41530, 2023.

Shufan Li, Konstantinos Kallidromitis, Akash Gokul, Arsh Koneru, Yusuke Kato, Kazuki Kozuka, and Aditya Grover. Reflect-dit: Inference-time scaling for text-to-image diffusion transformers via in-context reflection. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15657–15668. IEEE, 2025.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

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. *Advances in neural information processing systems*, 38:40783–40818, 2026.

Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, and Saining Xie. Scaling inference time compute for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2523–2534, June 2025.

Shunqi Mao, Wei Guo, Chaoyi Zhang, Jieting Long, Ke Xie, and Weidong Cai. Ctrl-z sampling: Scaling diffusion sampling with controlled random zigzag explorations. *arXiv preprint arXiv:2506.20294*, 2025.

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

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori B Hashimoto. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20275–20321, 2025.

Mihir Prabhudesai, Anirudh Goyal, Deepak Pathak, and Katerina Fragkiadaki. Aligning text-to-image diffusion models with reward backpropagation, 2024. URL https://arxiv.org/abs/2310.03739.

Xiangyan Qu, Zhenlong Yuan, Jing Tang, Rui Chen, Datao Tang, Meng Yu, Lei Sun, Yancheng Bai, Xiangxiang Chu, Gaopeng Gou, Gang Xiong, and Yujun Cai. From scale to speed: Adaptive test-time scaling for image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23272–23282, June 2026.

Pau Rodriguez, Arno Blaas, Michal Klein, Luca Zappella, Nicholas Apostoloff, marco cuturi, and Xavier Suau. Controlling language and diffusion models by transporting activations. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=l2zFn6TIQi.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

Christoph Schuhmann. Laion-aesthetics. LAION Blog, August 2022. URL https://laion.ai/blog/laion-aesthetics/.

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

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2009.

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8228–8238. IEEE, 2024.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

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

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.

Xiao-Wen Yang, Ziyu Han, Xi-Hua Zhang, Wen-Da Wei, Jie-Jing Shao, Lan-Zhe Guo, and Yu-Feng Li. Stabilizing recurrent dynamics for test-time scalable latent reasoning in looped language models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=tbdSfBMlbq.

Jinyan Ye, Zhongjie Duan, Zhiwen Li, Cen Chen, Daoyuan Chen, Yaliang Li, and Yingda Chen. Spectral evolution search: Efficient inference-time scaling for reward-aligned image generation. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=tBt12b9dxa.

Qingtao Yu, Changlin Song, Minghao Sun, Zhengyang Yu, Vinay Kumar Verma, Soumya Roy, Sumit Negi, Hongdong Li, and Dylan Campbell. Ttsnap: Test-time scaling of diffusion models via noise-aware pruning. *arXiv e-prints*, pp. arXiv–2511, 2025.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 586–595. IEEE, 2018.

Tao Zhang, Cheng Da, Kun Ding, Huan Yang, kun jin, Yan Li, Tingting Gao, Di ZHANG, Shiming Xiang, and Chunhong Pan. Diffusion model as a noise-aware latent reward model for step-level preference optimization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=YB9VGCClv9.

Zikai Zhou, Shitong Shao, Lichen Bai, Shufei Zhang, Zhiqiang Xu, Bo Han, and Zeke Xie. Golden noise for diffusion models: A learning framework. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17688–17697. IEEE, 2025.

Le Zhuo, Liangbing Zhao, Sayak Paul, Yue Liao, Renrui Zhang, Yi Xin, Peng Gao, Mohamed Elhoseiny, and Hongsheng Li. From reflection to perfection: Scaling inference-time optimization for text-to-image diffusion models via reflection tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15329–15339, October 2025.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023.

CONTENTS OF THE APPENDIX

A Recurrent Refinement: Calibration, Effort, and Long Unrolls . . . 14
A.1 Calibration at the Same Refinement Effort . . . 14
A.2 Refinement Effort at Inference . . . 15
A.3 Long Unrolls Before and After Calibration . . . 18
B Additional Results and Controls . . . 18
B.1 Larger Search Budgets . . . 18
B.2 Generalization Under Prompt Distribution Shift . . . 18
B.3 Effect on Sample Diversity . . . 19
B.4 Comparison with Fine-Tuning . . . 19
B.5 Ablation Definitions . . . 19
C Extended Experimental Protocol . . . 21
D Implementation Details . . . 22
E Choosing the Intervention Point . . . 23
F Extended Related Work . . . 25
G Selection Under a Fixed Proposal: Reweighting, Scaling, and Discovery . . . 27
H Deriving the Reward-Ascent Field: Trust Region and Restriction to the Secant Span . . . 28
I Reward-Ascent Matching (RAM): Centering, Dual Form, and Guarantees . . . 29
J Reward-Flow Calibration (RFC): Rounds, Occupancy, and Guarantees . . . 30
K Value Growth Along the Flow . . . 31

Section A collects the evidence on recurrent refinement: calibration at matched refinement effort, the refinement-effort sweep, and long unrolls before and after RFC. Section B collects additional results and controls, including larger search budgets, transfer to DrawBench prompts (Section B.2), the effect of refinement on sample diversity (Section B.3), and the control definitions behind Table 2. Sections C and D record the extended experimental protocol, including the reward scales behind Δ/σ (Table 7), and the implementation defaults, and Section E describes the block sweep behind the intervention point. Section F expands the two-axis taxonomy over prior work, and Section G characterizes the fixed-proposal selection frontier and discovery limits of selection under a fixed proposal. Section H derives the ideal reward-ascent field, Sections I and J detail the two training stages and their theoretical properties, and Section K analyzes value growth along the learned flow.

A RECURRENT REFINEMENT: CALIBRATION, EFFORT, AND LONG UNROLLS

A.1 CALIBRATION AT THE SAME REFINEMENT EFFORT

Table 3 shows that the RAM checkpoint improves all twelve backbone–reward pairs when unrolled to $K = 30$, and RFC further improves each pair at the same refinement effort. On three pairs, one RAM update outperforms the $K = 30$ RAM unroll; later updates act on states produced by earlier ones, not the frozen-generator activations used to train RAM. After RFC, the $K = 30$ unroll outperforms one update on all twelve pairs.

Table 3: **RAM, RFC, and recurrence at the same refinement effort.** Paired per-prompt normalized gains Δ/σ over the frozen generator. The *w/o recurrence* rows apply the same checkpoint once (Section B.5). Best values within each backbone are in bold.

Backbone	Config.	ImageReward	Aesthetic	HPSv2	PickScore
SANA-1.5	RAM w/o recurrence	+0.22	+1.00	+0.59	+0.32
	RAM w/ recurrence	+0.25	+1.18	+0.66	+0.41
	RFC w/o recurrence	+0.14	+1.22	+0.72	+0.39
	RFC w/ recurrence	**+0.26**	**+1.39**	**+0.78**	**+0.44**
SD3.5-Medium	RAM w/o recurrence	+0.07	+0.51	+0.30	+0.13
	RAM w/ recurrence	+0.08	+0.58	+0.28	+0.11
	RFC w/o recurrence	+0.07	+0.79	+0.44	+0.27
	RFC w/ recurrence	**+0.12**	**+0.91**	**+0.50**	**+0.30**
FLUX.2-klein	RAM w/o recurrence	+0.21	+1.32	+0.50	+0.42
	RAM w/ recurrence	+0.25	+1.34	+0.47	+0.46
	RFC w/o recurrence	+0.26	+1.32	+0.45	+0.41
	RFC w/ recurrence	**+0.31**	**+1.35**	**+0.52**	**+0.48**

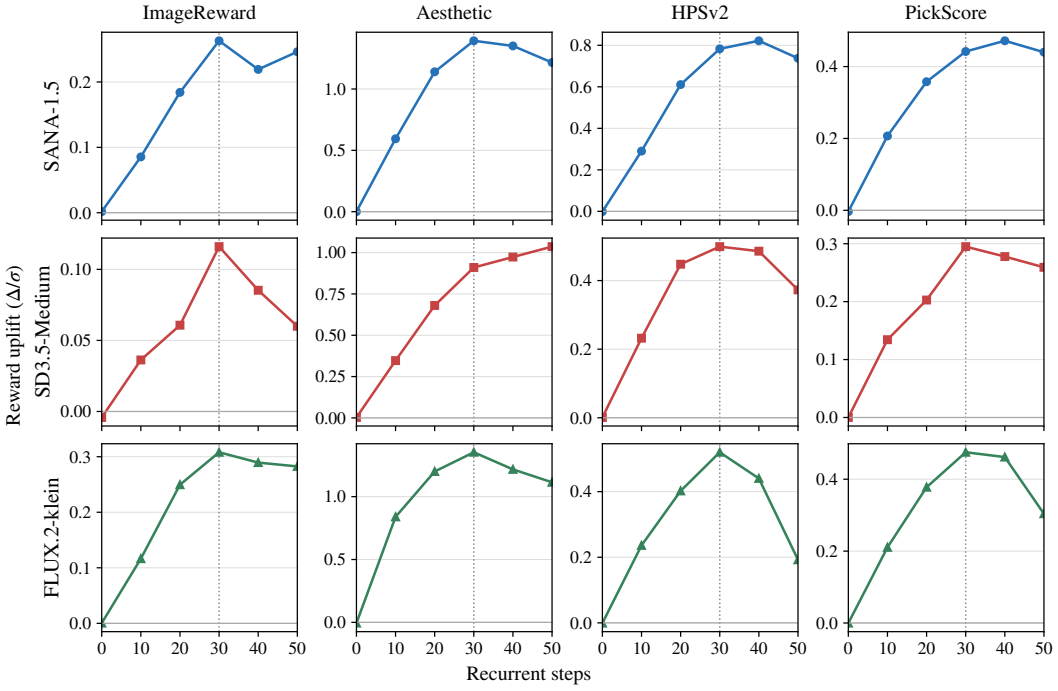

Figure 5: **Refinement effort at inference.** Paired normalized gain Δ/σ over the frozen generator against refinement effort K on the 500 test prompts. Columns are the four rewards and rows the three backbones; $K = 0$ is the frozen generator. The vertical reference lines mark $K = K_{\text{train}} = 30$, used for the primary ReCAT results.

A.2 Refinement Effort at Inference

Inference keeps Δs fixed and varies K, giving a total integration horizon $\tau_K = K\Delta s = (K/K_{\text{train}})\eta$. For $K \leq K_{\text{train}}$, refinement stays within the range used for RFC calibration. Figure 5 sweeps K on all three backbones and all four rewards. Gains rise through the calibrated range on every backbone and reward. Past K_{train}, the unroll extends beyond the calibration horizon, and the trends differ across rewards: some keep improving, most clearly Aesthetic on SD3.5-Medium, while others decline, most steeply HPSv2 on FLUX.2-klein.



Figure 6: **Long recurrent unroll before RFC calibration.** SANA-1.5 outputs at $K = 0, \ldots, 109$ from the Aesthetic-trained RAM checkpoint for one prompt and seed. The header at $K = 0$ gives the absolute Aesthetic score, and subsequent headers give the Aesthetic score change from $K = 0$; green and red denote positive and negative changes. In this illustrative rollout, the score stays below its initial value from $K = 55$ onward, after which the outputs collapse into a repetitive pattern.



Figure 7: **Long recurrent unroll after RFC calibration.** The RFC-calibrated checkpoint is evaluated with the same SANA-1.5 prompt, seed, step size, and Aesthetic objective as Figure 6. The header at $K = 0$ gives the absolute Aesthetic score, and subsequent headers give the Aesthetic score change from $K = 0$. Although the refinement sequence is non-monotonic beyond the calibrated horizon $K_{\text{train}} = 30$, the output remains coherent through $K = 109$ and ends $+0.562$ above its initial score in this illustrative rollout.

A.3 Long Unrolls Before and After Calibration

Figures 6 and 7 follow one SANA-1.5 prompt and seed through a long recurrent unroll, from the RAM checkpoint and from the RFC-calibrated checkpoint, respectively. They extend past K_{train} the RAM-to-RFC comparison that Table 3 reports for all backbone–reward pairs at K_{train}. In this rollout, the RAM unroll falls below its initial score and collapses into a repetitive pattern, whereas the RFC-calibrated unroll remains coherent through $K = 109$ and ends above its initial score.

B Additional Results and Controls

B.1 Larger Search Budgets

Table 4 reports paired gains of the outer-loop methods of Figure 4 at generation-equivalent budgets $B = 9$ and $B = 31$.

Table 4: **Outer-loop comparison at larger search budgets.** Paired normalized gains Δ/σ over the frozen generator at generation-equivalent budgets $B = 9$ and $B = 31$. These use $N = 9, 31$ for Best-of-N, $N = 41, 151$ for Probe-Select, and 4,15 updates for Zero-Order Search and ReNO, respectively. Probe-Select has no ReCAT-composed row.

	ImageReward		Aesthetic		HPSv2		PickScore	
Method	B=9	B=31	B=9	B=31	B=9	B=31	B=9	B=31
SANA-1.5								
Probe-Select	+0.203	+0.188	+0.720	+0.722	+0.622	+0.752	+0.428	+0.497
Best-of-N	+1.158	+1.440	+1.607	+2.061	+1.507	+2.041	+1.523	+2.018
ReCAT + Best-of-N	+1.316	+1.586	+2.453	+2.841	+2.024	+2.532	+1.885	+2.331
Zero-Order Search	+0.925	+1.364	+1.206	+1.908	+1.193	+1.889	+1.211	+1.919
ReCAT + Zero-Order Search	+1.141	+1.479	+2.364	+2.909	+1.868	+2.486	+1.652	+2.286
ReNO	+0.746	+1.478	+0.691	+2.056	+0.925	+2.017	+1.003	+2.671
ReCAT + ReNO	+1.003	+1.673	+2.105	+3.039	+1.571	+2.649	+1.410	+3.204
SD3.5-Medium								
Probe-Select	+0.122	+0.242	+0.192	+0.233	+0.264	+0.409	+0.185	+0.135
Best-of-N	+1.169	+1.452	+1.199	+1.653	+1.429	+1.943	+1.320	+1.812
ReCAT + Best-of-N	+1.228	+1.505	+2.015	+2.445	+1.909	+2.408	+1.567	+2.089
Zero-Order Search	+1.021	+1.382	+1.188	+1.808	+1.256	+1.916	+1.206	+1.847
ReCAT + Zero-Order Search	+1.067	+1.471	+1.995	+2.505	+1.711	+2.361	+1.498	+2.138
ReNO	+0.668	+1.010	+0.872	+1.282	+0.847	+1.340	+0.848	+1.232
ReCAT + ReNO	+0.766	+1.070	+1.767	+2.088	+1.361	+1.834	+1.143	+1.536
FLUX.2-klein								
Probe-Select	+0.138	+0.120	+0.624	+0.393	+0.248	+0.211	+0.228	+0.179
Best-of-N	+1.187	+1.456	+1.175	+1.707	+1.274	+1.796	+1.355	+1.870
ReCAT + Best-of-N	+1.371	+1.649	+2.362	+2.780	+1.781	+2.233	+1.813	+2.358
Zero-Order Search	+1.014	+1.460	+0.990	+1.642	+1.101	+1.818	+1.101	+1.847
ReCAT + Zero-Order Search	+1.234	+1.636	+2.234	+2.748	+1.581	+2.265	+1.591	+2.335
ReNO	+0.779	+1.510	+0.810	+2.037	+0.901	+2.102	+0.921	+2.376
ReCAT + ReNO	+1.035	+1.707	+1.896	+3.074	+1.291	+2.444	+1.371	+2.811

B.2 Generalization Under Prompt Distribution Shift

ReCAT amortizes the transition across prompts, so a natural question is whether ReCAT transfers to prompts far from its training distribution. We evaluate ReCAT for all three backbones on the 162 DrawBench prompts (Saharia et al., 2022), disjoint from the caption-style training distribution, and score the outputs under the protocol of Table 1. Across backbones, the DrawBench four-reward macro-average is 61.4–89.2% of its value on the held-out MS-COCO prompts (Table 5).

Table 5: **Zero-shot transfer to DrawBench prompts.** ReCAT gains on the 162 DrawBench prompts outside the Reddit category, normalized by each reward's σ from the held-out MS-COCO prompts as in Table 1. The last column reports the DrawBench-to-MS-COCO ratio of the four-reward macro-average Δ/σ.

Backbone	ImageReward	Aesthetic	HPSv2	PickScore	Macro retention
SANA-1.5	+0.211	+1.328	+0.587	+0.443	89.2%
SD3.5-Medium	+0.105	+0.777	+0.388	+0.129	76.9%
FLUX.2-klein	+0.229	+0.694	+0.314	+0.391	61.4%

B.3 Effect on Sample Diversity

Figure 8 compares within-prompt pairwise LPIPS before and after refinement on SANA-1.5. Mean LPIPS increases under all four objectives.

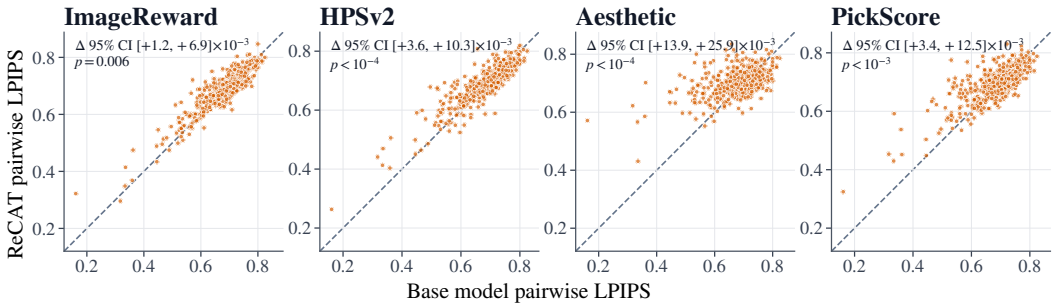

Figure 8: **Sample diversity under refinement.** Within-prompt pairwise LPIPS (Zhang et al., 2018) across seeds before and after ReCAT refinement on SANA-1.5, one panel per training reward. Diversity is evaluated on the 500 test prompts; each point is one prompt, and each panel reports the paired bootstrap 95% confidence interval and two-sided p-value for the difference, refined minus base. Points on or above the dashed identity line are prompts whose refined samples are at least as mutually distinct as the frozen ones.

B.4 Comparison with Fine-Tuning

Supervised fine-tuning and preference optimization (Wallace et al., 2024) update the generator weights and produce a fixed fine-tuned model; they do not provide an inference-time choice of refinement depth (Table 8). We train both by full-parameter fine-tuning on the same prompts and reward as ReCAT, and evaluate them under the protocol of Table 1 (Table 6). They update all 1.6B–4B generator parameters, whereas ReCAT trains a 9–13M-parameter controller and leaves the generator frozen. On the multi-step backbones, DPO exceeds ReCAT on Aesthetic and HPSv2, and on SANA-1.5 also on ImageReward; SFT exceeds it on ImageReward alone. Neither exceeds it on PickScore on either backbone. On the distilled FLUX.2-klein, SFT lowers all four rewards and DPO lowers three of the four, while ReCAT, which leaves the generator frozen, improves all four rewards on the same backbone.

Fine-tuning settings. For each training prompt, the target sample is the highest-reward sample among the same 48 seeds that supply RAM's teachers; DPO pairs it with the lowest-reward seed and scores both against the frozen backbone as reference. SFT minimizes the flow-matching loss on the target sample alone. Both optimize all generator parameters with AdamW at learning rate 10^{-5} and weight decay 10^{-4}, cosine-decayed to zero over 20 epochs, on eight NVIDIA H100 GPUs in bfloat16 with fully sharded data parallelism. The reported checkpoint is selected on the same validation prompts. Noise levels are drawn logit-normally over the scheduler's training grid. For FLUX.2-klein they are drawn from the four levels its distilled sampler visits at 1024×1024, which its scheduler obtains by shifting that grid per call.

B.5 Ablation Definitions

The training controls in Table 2 change the supervision target, the intervention state, the initialization method, or the exploration strategy, while retaining the same controller architecture. The inference-

Table 6: **Comparison with fine-tuning.** Paired per-prompt normalized gains Δ/σ over each frozen generator, for supervised fine-tuning (SFT) and preference optimization (DPO) of each backbone. The ReCAT rows repeat Table 1. Trainable-parameter counts are reward-independent and rounded to three significant digits.

Method	ImageReward	Aesthetic	HPSv2	PickScore	Trainable params.
SANA-1.5					
SFT	+0.432	−0.412	+0.392	−0.181	1.60B
DPO	+0.701	+1.434	+0.819	+0.326	1.60B
ReCAT	+0.263	+1.393	+0.783	+0.442	9.05M
SD3.5-Medium					
SFT	+0.660	−0.860	+0.333	−0.090	2.47B
DPO	−0.082	+1.137	+1.315	+0.137	2.47B
ReCAT	+0.116	+0.910	+0.499	+0.295	9.45M
FLUX.2-klein					
SFT	−1.289	−1.243	−2.990	−2.086	3.88B
DPO	+0.063	−2.323	−2.859	−1.451	3.88B
ReCAT	+0.308	+1.353	+0.520	+0.475	13.4M

update control changes how the trained RFC controller is applied. Write $\mathrm{unit}(v) = v/\|v\|_a$ for a nonzero activation-space direction and $d_{ij} = a_j - a_i$.

RAM target controls. For the group-best-direction control, let $j^\star = \arg\max_j r_j$, with ties broken by seed index, and use $\mathrm{unit}(d_{ij^\star})$ as the target for every $i \neq j^\star$. The zero displacement at $i = j^\star$ is omitted.

For the reward-weighted mean control, let $\sigma_c^2 = S^{-1}\sum_j (r_j - \bar{r}_c)^2$ and use the same locality weights w_{ij} as RAM. Given temperature $\tau > 0$, define

$$\psi_{ij} = \frac{w_{ij}^2 \exp\big(r_j/(\tau\sigma_c)\big)}{\sum_k w_{ik}^2 \exp\big(r_k/(\tau\sigma_c)\big)}, \qquad d_i^{\mathrm{rw}} = \mathrm{unit}\left(\sum_j \psi_{ij} d_{ij}\right).$$

This is the direction toward a soft reward-weighted local mean, and the softmax normalization in ψ_{ij} makes it invariant to a shift of the rewards.

For the entropic OT barycenter, set $q_j \propto \exp((r_j - \bar{r}_c)/(\tau\sigma_c))$, the empirical form of the reward tilt of Equation (14), and let $C_{ij} = \|a_i - a_j\|_a^2$, the ground cost under the geometry of Equation (1). Compute the entropically regularized coupling

$$\pi^\star = \underset{\substack{\pi\mathbf{1}=\mathbf{1}/S \\ \pi^\top\mathbf{1}=q}}{\arg\min} \sum_{ij} \pi_{ij} C_{ij} + \varepsilon \sum_{ij} \pi_{ij}(\log \pi_{ij} - 1).$$

This control regresses the barycentric projection (Villani et al., 2009) itself rather than a direction,

$$\mathcal{L}_{\mathrm{OT}}(\theta) = \mathbb{E}_c \frac{1}{S} \sum_i \left\| m_\theta(a_i, c) - S \sum_j \pi_{ij}^\star a_j \right\|_a^2,$$

where m_θ has the controller architecture but predicts a point, and at inference it interpolates once toward that point,

$$a' = (1-\omega)\, a + \omega\, m_\theta(a, c).$$

The regression target is one empirical transport step from the group toward its reward-tilted reweighting. The group-best and reward-weighted mean controls use the unit-direction regression of Equation (4), with zero targets omitted. We select ω, the temperatures, and ε on the same validation prompts as the RAM intervention strength.

RFC initialization and exploration controls. At a visited state, let $v = \widehat{g}_{\bar{\theta}}(a, c)$. RFC evaluates the radial antithetic pair $a \pm h(a)v$. The random-initialization row applies this same operation to a controller that never saw RAM teachers.

The transverse controls compare two directions tilted around v, with the transverse axis chosen from the same subspace. Let $b_i = a_i - \bar{a}$ be the group's mean-centered seed activations, and let $\Pi_{v^\perp} u = u - \langle u, v\rangle_a v$ project out the parent direction. The principal transverse control takes u along the leading principal direction of $\{\Pi_{v^\perp} b_i\}$, obtained from the top eigenvector of the $S \times S$ matrix with entries $\langle \Pi_{v^\perp} b_i, \Pi_{v^\perp} b_j\rangle_a$. The random control instead draws fresh coefficients $\xi \sim \mathcal{N}(0, I_S)$ at every visited state and takes u along $\sum_i \xi_i \Pi_{v^\perp} b_i$, so its axis is random within the cross-seed span rather than isotropic in activation space. Both then normalize u and evaluate the cone pair

$$w^{\pm} = \mathrm{unit}(v \pm \vartheta u), \qquad r^{\pm} = R\big(G_{>(t,\ell)}\big(a + h(a)w^{\pm}, c\big), c\big), \qquad \vartheta = 0.25,$$

and, writing $\zeta = (r^+ - r^-)/(2\vartheta\sigma_R)$, fit the transverse-coordinate objective

$$\frac{1}{2}\left(\frac{\langle \widehat{g}_\theta, u\rangle_a}{\langle \widehat{g}_\theta, v\rangle_a} - \gamma\zeta\right)^2 + \lambda\big(1 - \langle \widehat{g}_\theta, v\rangle_a\big),$$

where γ normalizes the slope RMS on the training bank and the anchor retains the parent direction when transverse evidence is weak. The two controls therefore differ only in how the transverse axis is chosen. Both transverse controls and radial antithetic probing use two frozen-suffix evaluations per visited state.

Intervention-state controls. The two controls move RAM from a block activation to either the initial noise or the sampler latent. They keep the RAM objective, controller architecture, one update of Equation (5), and validation-based strength selection fixed; only the edited state changes. Each control also has the same supervision cost: one sample group of size S, with each generation scored once. The noise control acts on the initial noise z, before any denoising step, so group displacements are $z_j - z_i$ and the geometry of Equation (1) is read in the coordinates of z. The latent control acts on the sampler's own iterate at the same intervention point. It differs from ReCAT only in editing the state the frozen model consumes between denoising steps rather than an activation inside one block, and it leaves strictly more frozen computation downstream, since its edit additionally traverses the blocks before ℓ at that step.

Inference update. The *w/o recurrence* control uses the same RFC checkpoint and applies Equation (5) once before running the frozen suffix. Its update strength is selected on the same validation prompts as the recurrent configuration.

C Extended Experimental Protocol

Backbones. The three generators are Efficient-Large-Model/SANA1.5_1.6B_1024px_diffusers (1.6B parameters, 20 blocks), stabilityai/stable-diffusion-3.5-medium (2.5B, 24 MMDiT blocks), and black-forest-labs/FLUX.2-klein-4B (4B, 5 dual and 20 single blocks; step-distilled release). All three generate at 1024×1024. Sampling uses the diffusers pipeline default for each model: 20 steps at guidance 4.5 for SANA-1.5, 28 steps at guidance 7.0 for SD3.5-Medium, and 4 steps without guidance for the distilled FLUX.2-klein.

Prompts and splits. Training uses 5,000 captions, hyperparameter and checkpoint selection uses a disjoint set of 200, and reporting uses a disjoint set of 500. No prompt appears in more than one split.

Paired per-prompt gains. The four rewards live on incommensurable scales, so result tables report normalized gains. Each test prompt contributes the difference between the method's reward and the frozen generator's reward on that prompt. Result tables report the mean of these differences over the 500 test prompts, divided by the frozen generator's across-seed reward standard deviation, computed within each prompt and averaged over prompts (Table 7). For reference, if per-prompt scores were Gaussian across seeds, keeping the better of two draws would gain $1/\sqrt{\pi} \approx 0.56\,\sigma$.

Table 7: **Conversion key for normalized gains.** Frozen-generator mean score and across-seed standard deviation σ on the held-out test set; multiplying a Δ/σ entry by σ recovers the raw score difference.

	ImageReward		Aesthetic		HPSv2		PickScore	
	mean	σ	mean	σ	mean	σ	mean	σ
SANA-1.5	1.092	0.282	5.861	0.324	0.293	0.0063	22.792	0.414
SD3.5-Medium	1.015	0.320	5.833	0.398	0.292	0.0065	22.527	0.432
FLUX.2-klein	1.141	0.244	6.080	0.372	0.298	0.0058	23.078	0.382

Baseline settings. The learned baselines use the same training prompts and target reward as ReCAT. ReNO and Noise Hypernet are adapted from their official implementations. Where a released implementation optimizes a weighted combination of several rewards, we optimize the single target reward, as every other method in the comparison does.

Best-of-N and Zero-Order Search. Both are from Ma et al. (2025): Best-of-N draws N independent initial noises, completes each, and returns the highest-reward image. Zero-Order Search holds a pivot noise and, at each update, draws neighbors at cosine similarity 0.95 to it, completes and scores them, and promotes the best one to pivot. Its neighbor and update counts follow the compute budget of each comparison; in Table 1 that budget is one trajectory beyond the frozen generator's. In ReCAT + Best-of-N, every candidate is refined before the same selection.

Probe-Select. We implement the early-pruning probe as a linear score on a pooled, dimension-reduced form of the same intermediate activation that ReCAT edits, trained with AdamW to rank the seeds within each prompt. The reported probe is the checkpoint with the best within-prompt rank correlation on the validation prompts.

ReNO. Each prompt's initial noise is optimized with SGD at learning rate 5.0 and Nesterov momentum 0.9, gradient-norm clipping at 0.1, and the released regularizer on the noise norm at weight 0.01, backpropagating the reward through the backbone's full sampling chain. The iterate count is set by the compute budget of each comparison, and the highest-reward iterate visited is returned; Figure 4 sweeps it. For ReNO+ReCAT, every ReNO iterate is generated with the frozen controller in the forward path, and the reward gradient propagates through the controller to the optimized initial noise. ReNO and ReNO+ReCAT start from the same seed but follow separate optimization trajectories; each returns the highest-reward iterate it visits.

Noise Hypernet. The hypernetwork is a rank-128 LoRA with $\alpha = 2r$ on the generator's output projection, trained for 25 epochs over the training prompts with reward gradients backpropagated through the full sampling chain, gradient-norm clipping at 1.0, and the released ℓ_2 regularizer on the modulated noise at weight 0.5. We train the hypernetwork on eight NVIDIA H100 GPUs. The optimizer, learning rate, and LoRA scale are selected per backbone on the validation prompts. Inference is the proposed single call.

D Implementation Details

Intervention point. The intervention blocks are 13 (SANA-1.5), 19 (SD3.5-Medium), and 7 (FLUX.2-klein), each after the first 20% of the sampler's denoising steps. At the intervention point, ReCAT applies $\mathcal{T}_\theta$ to the block's output activation K times, then resumes the remaining blocks and denoising steps from the refined activation. The block sweep behind these choices is in Section E.

ReCAT controller. The controller is a ConvNeXt-style residual network over the activation's token grid, conditioned on the prompt embedding the generator already computes, with hidden width 384, four blocks, and an inverted-bottleneck ratio of 4. Its head predicts a per-token direction, which Equation (5) normalizes and scales before adding it to the activation. The controller has 9–13M parameters depending on the backbone (Table 6); these are the only trainable parameters in the pipeline. We train the controller on NVIDIA A10G GPUs in both RAM and RFC.

Reward-Ascent Matching (RAM) defaults. Each sample group contains $S = 48$ samples. The bandwidth is the median pairwise displacement norm (Section I); teachers follow the ridge regression of Equation (3) with ridge ratio $\gamma = 0.01$, and zero-norm teachers are dropped. The controller

Algorithm 1 ReCAT. RAM distills secant teachers into a controller, RFC calibrates its directions on its own unrolls, and inference applies it for K steps. Neither frozen model is ever differentiated.

Require: Frozen generator G split at the intervention point into prefix $A_{t,\ell}$ and suffix $G_{>(t,\ell)}$; black-box reward R; training prompts; group size S; calibration horizon K_{train}, step size Δs; label temperature κ; replay fraction β.

Stage 1: Reward-Ascent Matching (Section 3.2, Section I)

for each training prompt c **do**
  Draw S seeds and run the prefix and suffix once per seed to get $\{(a_i, r_i)\}_{i=1}^{S}$, with $a_i = A_{t,\ell}(z_i, c)$ and $r_i = R(G(z_i, c), c)$.
  Form the activation Gram $\Gamma_{jk} = \langle a_j, a_k \rangle_a$ and the ridge ρ from its centered spectrum; for each source i get $K^{(i)}$ by Equation (16), the bandwidth h_i from its diagonal, and $\widetilde{K}^{(i)}$ by Equation (17).
  Solve the $S \times S$ dual system Equation (18) for α_i; store $(\alpha_i, \text{seed indices})$ and drop zero-norm teachers.
end for
Fit g_θ by Equation (4), reconstructing $g_c^\star(a_i) = \widetilde{D}_i^\top \alpha_i$ on demand and normalizing it to the unit teacher $\widehat{g}_c^\star(a_i)$.

Stage 2: Reward-Flow Calibration (Section 3.3, Section J)

while held-out validation uplift keeps improving **do**
  Freeze a collection copy $g_{\bar\theta} \leftarrow g_\theta$ and a batch of prompts with seed states.
  for each seed state, $k = 0, \ldots, K_{\text{train}} - 1$ **do**
    $h \leftarrow \Delta s \,\mathrm{rms}(a^{(k)})$; $a^{\pm} \leftarrow a^{(k)} \pm h\, \widehat{g}_{\bar\theta}(a^{(k)}, c)$.
    Decode both through $G_{>(t,\ell)}(\cdot, c)$, score $r^{\pm}$, and record $\big(a^{(k)}, c, \chi = \tanh((r^+ - r^-)/(\kappa\sigma_R))\big)$ in $\mathcal{D}_{\bar\theta}$.
    Advance the unroll through the positive branch, $a^{(k+1)} \leftarrow a^+$.
  end for
  Update θ on $\mathcal{L}_{\text{RFC}}(\theta; \bar\theta)$ of Equation (7), replaying a fraction β of RAM teachers as an anchor.
end while

Inference (Section 3.4); refinement effort K chosen here, no reward call

$a^{(0)} \leftarrow A_{t,\ell}(z, c)$.
for $k = 0, \ldots, K - 1$ **do**
  $a^{(k+1)} \leftarrow a^{(k)} + \Delta s \,\mathrm{rms}(a^{(k)})\, \widehat{g}_\theta(a^{(k)}, c)$ ▷ Equation (5), one controller call
end for
return $G_{>(t,\ell)}(a^{(K)}, c)$ ▷ the suffix decodes the final state only

is fit with AdamW at learning rate 10^{-4}, weight decay 10^{-4}, in bfloat16. We keep the checkpoint with the highest mean cosine between the predicted direction and the teacher direction.

Calibration horizon and step size. We use $K_{\text{train}} = 30$ equal refinement steps over a total horizon η selected on validation prompts for each backbone–reward pair, giving step size $\Delta s = \eta / K_{\text{train}}$.

Reward-Flow Calibration (RFC) defaults. The label temperature is $\kappa = 0.15$, the replay fraction $\beta = 0.1$, and the controller is trained with AdamW at learning rate 2×10^{-5}, weight decay 10^{-4}, in bfloat16. Each round trains one epoch on the states from its own unrolls and is then evaluated on the validation prompts. The reported controller is the round with the highest validation uplift.

Optimization memory. Supervision is collected through forward evaluations of the frozen generator and reward model. Backpropagation involves only the controller, so the frozen models' activations need not be retained for a backward pass. Noise Hypernet and the fine-tuning baselines backpropagate through the generator and exceed the A10G's memory, so they are trained on NVIDIA H100 GPUs.

E Choosing the Intervention Point

Given the activation state, whose choice is ablated in Table 2, the construction of Section 3.1 still leaves the block and timestep free. The earliest blocks carry little prompt-conditioned evidence (Guo

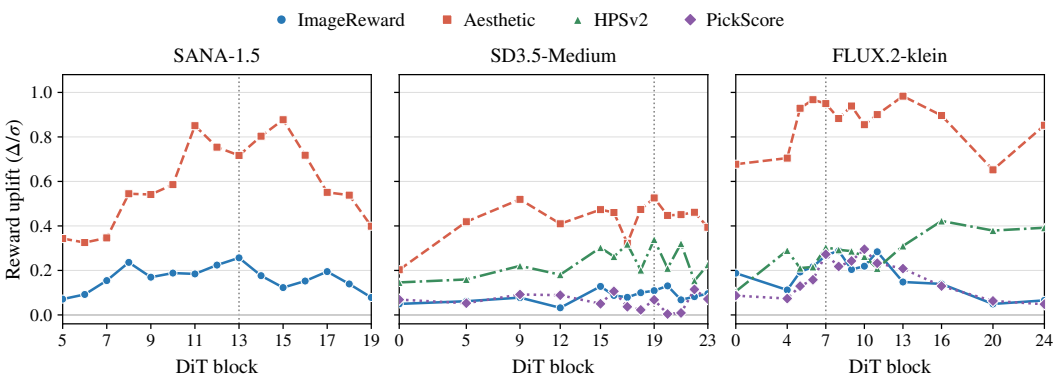

Figure 9: **RAM block sweep.** Validation reward gain Δ/σ of RAM trained on 2,000 prompts at candidate blocks of each backbone. The SANA-1.5 sweep covers ImageReward and Aesthetic; the other two backbones cover all four rewards. The vertical reference lines mark the intervention block used for each backbone across all four rewards.

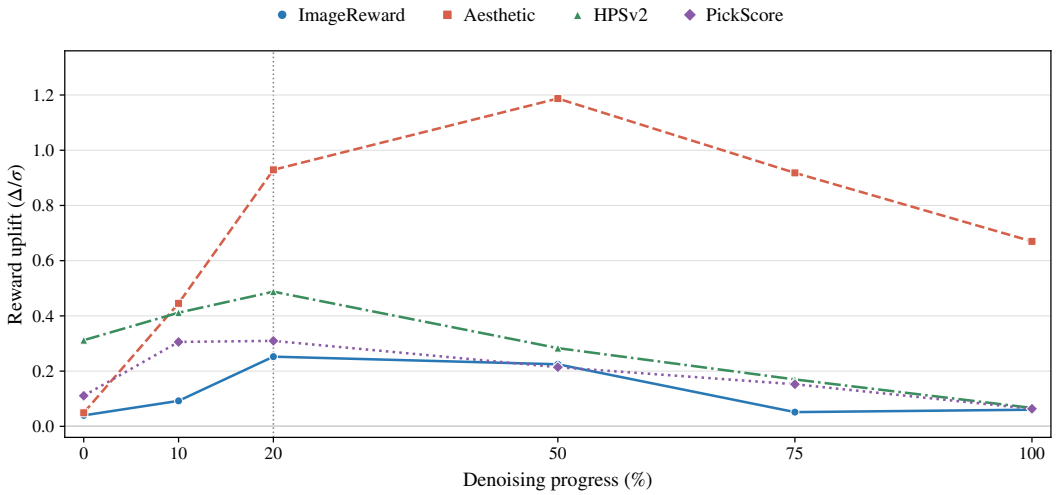

Figure 10: **Timestep sweep on SANA-1.5.** Normalized reward gain Δ/σ when the intervention is placed at a given fraction of the sampler's denoising steps. The vertical reference line marks the 20% setting used throughout.

et al., 2026; Zhang et al., 2025), and the latest blocks leave too short a frozen suffix for an edit to change the output, so the usable region is the middle of the network. Before full-scale training, we construct RAM teachers from 2,000 of the 5,000 training prompts at candidate blocks in this region on every backbone, and sweep them under a shared configuration (Figure 9). RAM yields gains across much of the region, and each reward's strongest block differs, so we fix one high-uplift block per backbone and reuse it for all four reward objectives. A single block per backbone supports gains on all four reward objectives (Table 3).

The timestep is swept on SANA-1.5 in the same way, and the resulting setting is reused for the other two backbones. Every reward peaks in the interior of the trajectory rather than at either endpoint (Figure 10): the first sampler steps act on a state that carries little of the content the reward responds to, and the last leave too short a remaining trajectory for an edit to reach the output. Three of the four rewards peak at 20% of the denoising steps, which is the setting we use for every backbone and reward. Aesthetic peaks at 50%. Both sweeps therefore indicate that tuning the intervention point to a single reward would improve on the numbers we report.

Table 8: Inference and training requirements across representative scaling and post-training methods. "Reusable" means the update is learned across prompts rather than optimized anew, and "variable depth" means the length of the dependent update chain is an inference-time choice. "Reward-only" refers to scalar reward supervision for constructing training targets or online updates, without reward gradients or textual critiques. "Reward-free" means inference queries no reward model or online verifier. G is a complete generator call, G_{pre} a trajectory prefix up to a probe point, ∇G a reward-gradient backward pass through the generator, C a complete corrector generation, V the online vision–language feedback and verification stack, P one probe call, H_{noise} one hypernetwork call, and H_θ one controller call. SFT and DPO are supervised and preference fine-tuning. "No generator backprop" also excludes backpropagation through a generator-sized diffusion corrector. A dash means not applicable.

	Training			Inference		
Method	Reusable	Reward-only	No generator backprop	Variable depth	Reward-free	Cost per image
Best-of-N	–	–	–	×	×	$N \times (G + R)$
Early pruning (Probe-Select)	✓	✓	✓	×	×	$G + (N{-}1) \times G_{\text{pre}} + N \times P$
Gradient noise optimization (ReNO)	×	×	×	✓	×	$N \times (G + \nabla G + R)$
Noise Hypernet	✓	×	×	×	✓	$1 \times H_{\text{noise}} + 1 \times G$
Fine-tuning (SFT, DPO)	✓	✓	×	×	✓	$1 \times G$
Online particle / tree search	×	✓	✓	✓	×	$N \times (G + R)$
Reflect-DiT	✓	×	×	✓	×	$K \times (G + V)$
ReflectionFlow	✓	×	×	✓	×	$G + K \times (C + V)$
ReCAT	✓	✓	✓	✓	✓	$K \times H_\theta + 1 \times G$

F Extended Related Work

We organize prior work along the two independent dimensions of Figure 1. The first asks whether added inference compute is allocated in *parallel*, across candidates, particles, or branches, or *sequentially*, along a dependent chain on one state from a single seed. The second asks whether an image is returned after a single decoded generation or only after several are decoded and scored. These dimensions separate where compute is allocated from how many generations are completed: parallel methods pay in population size, prior sequential refinement pays a generation per round, and amortized maps avoid repeated generations but provide only one update. The composition rows of Table 1 evaluate combining ReCAT's internal refinement with candidate selection.

The parallel-to-sequential shift in language models. Language-model test-time scaling began in parallel, with verifier-scored Best-of-N and majority vote over independent samples (Cobbe et al., 2021; Wang et al., 2023), and later gained a sequential scaling axis through longer explicit reasoning traces (Guo et al., 2025; Muennighoff et al., 2025). The trade between the two allocations is now studied in its own right (Snell et al., 2024). Post-training teaches the model how to use this axis, while explicit reasoning effort remains an inference-time choice. In recurrent-depth language models, a weight-tied block repeatedly updates an internal state before an output is produced, making iteration count an inference-time compute choice (Geiping et al., 2026; Yang et al., 2026). ReCAT learns a lightweight recurrent transition on the activation of a frozen generator, with its number of applications setting the refinement effort. Among the diffusion methods surveyed below, only ReCAT pairs this compute pattern with black-box training and reward-free inference (Table 8). The diffusion counterpart of post-training is fine-tuning the generator by reinforcement learning or preference optimization (Black et al., 2024; Fan et al., 2023; Wallace et al., 2024; Liu et al., 2026). It changes generator parameters, while ReCAT adds a recurrent transition whose application count is chosen at inference; Section B.4 compares the two on the same backbones and rewards.

Parallel scaling: candidates and populations under a fixed proposal. Best-of-N, terminal ranking, and early pruning draw candidates from the pretrained proposal and use scores to select, stop, or allocate suffix compute (Kirstain et al., 2023; Xu et al., 2023; Ma et al., 2025; Guo et al., 2026; Huang et al., 2026; Bucciarelli et al., 2026). Their per-prompt cost grows with the candidate population. Early pruning is especially useful below the cost of Best-of-2, where several prefixes fit before two complete generations do. Probe-Select reduces the cost of candidate selection by stopping unpromising trajectories early. ReCAT can refine candidates within either early pruning or terminal Best-of-N selection. Feynman–Kac steering repeatedly resamples promising partial trajectories and relies on stochastic continuations to branch their suffixes (Singhal et al., 2025). Tree search explicitly expands and scores branches of partial trajectories (Jain et al., 2025). Population-based noise

evolution updates distributions over initial noise across rounds (Ye et al., 2026). By the allocation criterion these remain parallel scaling: additional compute expands the candidate population or search tree. For the selection-only members of this family, the probability of discovering a rare high-value region follows the fixed-proposal rate of Proposition 2: aggregation cannot reach what the proposal never proposes. Adaptive variants can redirect generation, but every reward-directed move is constructed anew from additional candidates for each prompt.

Sequential scaling, online: per-instance chains. Gradient-based noise optimization, controlled zigzag exploration, and zero-order neighborhood search on the initial noise keep one current iterate alive and improve it through dependent updates (Eyring et al., 2024; Mao et al., 2025; Ma et al., 2025). Eyring et al. (2024) restrict ReNO to one-step models precisely because backpropagating the reward through the denoising steps yields exploding or vanishing gradients, and the multi-step noise-optimization methods that preceded it cost tens of minutes per image. Even in the one-step regime, each of ReNO's optimization steps pays a generator forward and backward pass plus a suite of reward evaluations, tens of seconds per image in total. Our ReNO comparison runs the full multi-step backbones and therefore includes this cost (Table 1). Critique-and-regenerate pipelines establish that dependent, variable-depth revision can improve diffusion outputs, but extend the chain across completed images. They train a diffusion model to revise earlier generated images using textual criticism. At test time, a vision–language model critiques each completed image, and the trained diffusion model uses that feedback to generate another complete image (Li et al., 2025; Zhuo et al., 2025).

Their training and inference costs are distinct. Reflect-DiT fine-tunes its diffusion generator to condition on previous images and textual critiques, together with the context-processing module that supplies this conditioning, and separately fine-tunes the vision–language model that writes the critiques (Li et al., 2025). ReflectionFlow trains a diffusion corrector that generates a new image from the preceding result and textual feedback, and a vision–language model that writes that feedback; its full pipeline additionally requires an image verifier. The authors train one such verifier, although reported configurations can instead use external verifiers (Zhuo et al., 2025). At inference these parameters are frozen, but each executed revision still requires another complete image generation followed by online vision–language feedback. Reflect-DiT reports that producing N samples has similar computational cost to Best-of-N: both generate N images and invoke a vision–language model N times. Unlike independent Best-of-N draws, the revisions within one chain are sequential. ReCAT keeps the dependent chain but moves it to the inner loop: the step is an internal activation transition rather than an outer regenerate-and-critique round, so inference needs no judge call, no decode per step, and no update to the generator.

Learned maps in noise space. Learned noise maps establish that prompt-conditioned actions can be amortized and applied to a frozen generator. Golden Noise, NoiseRefine, and related methods learn a single pre-generation update in noise space (Zhou et al., 2025; Ahn et al., 2026). For Noise Hypernet, this update is a residual map $z' = z + f_\phi(z, c)$, trained through the generator–reward path and applied once before generation (Eyring et al., 2026). LoRA reduces its trainable parameters and optimizer state, but the reward gradient still traverses the frozen generator and decoder. Reward-backpropagation fine-tuning similarly uses that path to update generator weights (Prabhudesai et al., 2024). ReCAT instead uses black-box supervision to amortize the projected activation-space direction derived in Section 3.2. The resulting transition reads its own previous output, making the refinement effort an inference-time control.

Internal states: observable, and editable. Probe networks estimate final quality from early denoising states (Guo et al., 2026; Huang et al., 2026), latent verifiers score candidates without a pixel decode (Bucciarelli et al., 2026), adaptive systems use intermediate evidence to stop, rank, or allocate compute (Kim et al., 2025; Qu et al., 2026), and noise-aware reward models read preference signal directly off noisy latents (Zhang et al., 2025): intermediate states are observable and predictive of reward. They are also editable: representation-level control edits attention maps, embeddings, concept directions, or hidden activations (Hertz et al., 2023; Mokady et al., 2023; Brack et al., 2023; Gandikota et al., 2023; Rodriguez et al., 2025), and inference-time interventions in language models add steering vectors that shift attributes such as truthfulness or sentiment (Li et al., 2023; Turner et al., 2023; Zou et al., 2023). Value estimates provide a selection criterion but not an update, while the cited interventions do not learn a reward-directed transition that reads its own output. ReCAT makes the activation the persistent state of a sequential computation: RAM distills secant teachers

into a prompt- and state-conditioned controller, and RFC further calibrates its directions on states visited by its own unrolls.

G Selection Under a Fixed Proposal: Reweighting, Scaling, and Discovery

Fix a conditioning input c, a state space with proposal measure μ_c (Section 3.1), and the early-state value V_c of Section 3.1.

Selection is an order-statistic tilt. Let $a_1, \dots, a_N \overset{\text{iid}}{\sim} \mu_c$ and let selection return $a_{i^\star}$ with $i^\star = \arg\max_i V_c(a_i)$. Write $F_c(v) = \mathbb{P}_{a\sim\mu_c}\big(V_c(a) \le v\big)$. If $V_c(a)$ has a continuous law under μ_c, the winner's law is

$$\nu_N(\mathrm{d}a) = N\, F_c\big(V_c(a)\big)^{N-1}\, \mu_c(\mathrm{d}a), \tag{10}$$

a reweighting of μ_c by a factor that depends on the state only through its value. This is the exact sense in which Best-of-N tilts the proposal toward reward, as invoked in Section 3.2, and it makes the limitation visible: the tilt is absolutely continuous with respect to μ_c, so states the proposal never produces remain unreachable at any N.

Oracle ceiling and diminishing returns. Set $R_i = V_c(a_i)$, $M_N = \max_{i\le N} R_i$, and

$$m_N(F_c) = \mathbb{E}[M_N].$$

This expected order statistic defines the fixed-proposal selection frontier.

Proposition 1 (Fixed-proposal selection frontier)**.** *Let a procedure observe arbitrary information about $a_1, \dots, a_N \overset{\text{iid}}{\sim} \mu_c$, choose an index $I \in \{1, \dots, N\}$, and return a_I without modifying it.*

1. *Its expected value is bounded by the oracle order statistic:*

$$\mathbb{E}[V_c(a_I)] \le m_N(F_c).$$

Equality is attained by the oracle rule $I \in \arg\max_i V_c(a_i)$.

2. *If R_i is integrable, $m_N(F_c)$ is increasing and discrete-concave:*

$$m_{N+1}(F_c) - m_N(F_c) = \mathbb{E}[(R_{N+1} - M_N)_+], \qquad m_{N+2} - m_{N+1} \le m_{N+1} - m_N. \tag{11}$$

Proof. For the first claim, pointwise $V_c(a_I) \le \max_{i\le N} V_c(a_i) = M_N$; take expectations. For the second, the identity follows from $M_{N+1} = M_N + (R_{N+1} - M_N)_+$. Since $M_{N+1} \ge M_N$,

$$\mathbb{E}[(R_{N+2} - M_{N+1})_+] \le \mathbb{E}[(R_{N+2} - M_N)_+] = \mathbb{E}[(R_{N+1} - M_N)_+],$$

where the equality uses the identical law and independence of the next draw. □

The oracle bound is independent of what a probe observes or how accurately it predicts V_c. Early pruning can therefore inspect more candidates for a fixed completion budget, but at a given candidate count its ceiling remains $m_N(F_c)$.

For a selection-only method with inference budget B, let $M_{\mathcal{A}}(B)$ be the maximum number of fixed-proposal trajectories it can initialize. Its best possible value obeys the decomposition

$$\mathcal{V}_{\mathcal{A}}(B; F_c) \le m_{M_{\mathcal{A}}(B)}(F_c). \tag{12}$$

By stopping unselected trajectories early, early pruning can initialize more candidates under the same inference budget B, increasing $M_{\mathcal{A}}(B)$. It therefore selects from a larger N, but does not change the selection frontier $m_N(F_c)$.

Tail behavior controls how quickly the frontier flattens. Proposition 1 gives diminishing gain per additional candidate without a distributional assumption. Stronger statements about multiplying the candidate count depend on the reward tail. If R_1 has finite upper endpoint $r_c^\star = \operatorname{ess\,sup} R_1$, then $m_N(F_c) \uparrow r_c^\star$, and for every fixed $q > 1$,

$$m_{\lceil qN \rceil}(F_c) - m_N(F_c) \longrightarrow 0.$$

For a familiar light-tailed calibration, write $\bar{r}_c = \mathbb{E}[R_1]$ for the reward mean and suppose $R_1 - \bar{r}_c$ is s_c-sub-Gaussian. Then

$$m_N(F_c) \le \inf_{\lambda>0} \left\{ \bar{r}_c + \frac{\log N}{\lambda} + \frac{\lambda s_c^2}{2} \right\} = \bar{r}_c + s_c\sqrt{2\log N}, \tag{13}$$

using $e^{\lambda M_N} \le \sum_i e^{\lambda R_i}$, Jensen's inequality, and the sub-Gaussian moment-generating-function bound. The infimum is attained at $\lambda = \sqrt{2\log N}/s_c$. For a Gaussian reward law this bound has the correct leading order, $m_N(F_c) \approx \bar{r}_c + \sigma_c\sqrt{2\log N}$, which gives the familiar fixed-factor increment

$$m_{\lceil qN \rceil}(F_c) - m_N(F_c) \approx \frac{\sigma_c \log q}{\sqrt{2\log N}}.$$

Proposition 2 (Proposal invariance of selection)**.** *Let $\mathcal{A}$ be any procedure that draws N independent states from μ_c and returns one of the drawn states, chosen by any rule that may depend on their values and additional independent randomness. Then: (i) the law of the returned state is absolutely continuous with respect to μ_c. (ii) for a target region A with $\mu_c(A) = p_\tau > 0$, the probability that any drawn state lies in A is $1 - (1-p_\tau)^N$. (iii) guaranteeing discovery of A with probability at least $1-\delta$ requires $N \ge (1-p_\tau)\log(1/\delta)/p_\tau$, while $N \ge \log(1/\delta)/p_\tau$ suffices.*

Proof. (i) For any measurable B with $\mu_c(B) = 0$, a union bound gives $\mathbb{P}(\exists i : a_i \in B) \le N\mu_c(B) = 0$, and the returned state is one of the a_i. (ii) Independence of the draws. (iii) The requirement $(1-p_\tau)^N \le \delta$ is equivalent to $N \ge \log(1/\delta)/(-\log(1-p_\tau))$. The bounds $p_\tau \le -\log(1-p_\tau) \le p_\tau/(1-p_\tau)$ give the two sides. □

The support argument extends to randomized weighted selection among the drawn states, since a mixture of laws absolutely continuous with respect to μ_c is again absolutely continuous.

Transport changes the frontier. Let $F_{c,\mathcal{T}}$ be the value law after applying a transition $\mathcal{T}$ to each proposal. Selection after transport follows $m_N(F_{c,\mathcal{T}})$, rather than moving farther along $m_N(F_c)$. As a sufficient condition, if $F_{c,\mathcal{T}}$ first-order stochastically dominates F_c, so $F_{c,\mathcal{T}}(v) \le F_c(v)$ for every v, then

$$F_{c,\mathcal{T}}(v)^N \le F_c(v)^N, \qquad m_N(F_{c,\mathcal{T}}) \ge m_N(F_c)$$

for every N. The two operations change different quantities in Equation (12): allocation changes the number of proposals $M_{\mathcal{A}}(B)$, while transport changes their value distribution from F_c to $F_{c,\mathcal{T}}$.

H Deriving the Reward-Ascent Field: Trust Region and Restriction to the Secant Span

The ideal transition of Section 3.2 maximizes the expected value under a pointwise bound on how far each state moves. The directional reference is the reward tilt: for any current activation law ρ and any temperature $\lambda > 0$, with rewards in their own units,

$$\nu^{\lambda}_{\rho,c}(\mathrm{d}a) = \frac{\exp\big(V_c(a)/\lambda\big)}{\mathbb{E}_{a'\sim\rho}\big[\exp\big(V_c(a')/\lambda\big)\big]}\rho(\mathrm{d}a), \qquad \nabla \log \frac{\mathrm{d}\nu^{\lambda}_{\rho,c}}{\mathrm{d}\rho}(a) = \lambda^{-1}\nabla V_c(a), \tag{14}$$

so the density-ratio score points along the reward gradient; the normalization in Equation (5) removes the positive scale factor.

Trust-region form. Displace each state by $u(a)$ with $\|u(a)\|_a \le \epsilon(a)$. To first order,

$$\mathbb{E}_\rho\big[V_c(a+u(a))\big] = \mathbb{E}_\rho[V_c] + \mathbb{E}_\rho\big[\langle \nabla V_c(a), u(a)\rangle_a\big] + o(\epsilon),$$

and the inner product is maximized pointwise, by Cauchy–Schwarz, at $u(a) = \epsilon(a)\nabla V_c(a)/\|\nabla V_c(a)\|_a$, which is Equation (2). This is the pointwise analogue of a Wasserstein proximal ascent step (Jordan et al., 1998). The maximizer is zero where the gradient vanishes. Because the constraint is pointwise, the optimal displacement at a given state does not depend on the current distribution ρ. The same ideal update rule therefore remains applicable as the activation distribution changes.

Restriction to the secant span. Restricting the displacement at a_i to the group's secant span $\mathcal{U}_i$ gives the ideal direction $P_{\mathcal{U}_i}\nabla V_c(a_i)$, normalized when nonzero. RAM constructs a computable secant teacher by regressing rewards on the observed displacements. Under the affine, noiseless conditions of Proposition 3, the exact regression recovers this projected gradient as $\rho \to 0$. RAM amortizes the secant teachers across prompts: each target lies in its group's span, while the fitted controller emits a full activation-space direction at each state.

I Reward-Ascent Matching (RAM): Centering, Dual Form, and Guarantees

The RAM regression can be solved using $S \times S$ matrices. This section derives the dual solution of Equation (3) and gives the affine recovery and finite-sample error properties of the regression.

Objective and centering. With $q_{ij} = w_{ij}^2 / \sum_k w_{ik}^2$, both sides of the regression are centered around locality-weighted means,

$$\bar{d}_i = \sum_j q_{ij} d_{ij}, \quad \bar{r}_i = \sum_j q_{ij} r_j, \quad \widetilde{d}_{ij} = d_{ij} - \bar{d}_i, \quad \widetilde{y}_{ij} = r_j - \bar{r}_i,$$

and the secant teacher is the minimizer of

$$g_c^\star(a_i) = \arg\min_g \sum_{j=1}^{S} w_{ij}^2 \big(\widetilde{y}_{ij} - \langle g, \widetilde{d}_{ij} \rangle_a\big)^2 + \rho \, \|g\|_a^2, \tag{15}$$

ridge-regularized local-linear regression with an unpenalized intercept.

Displacement Gram from the activation Gram. Let $\Gamma_{jk} = \langle a_j, a_k \rangle_a$ be the group's activation Gram. The ridge is $\rho = \gamma \operatorname{med}\{\lambda_k > 0\}$, the ridge ratio γ times the median positive eigenvalue λ_k of the centered Gram $\big(I - \frac{1}{S}\mathbf{1}\mathbf{1}^\top\big)\Gamma\big(I - \frac{1}{S}\mathbf{1}\mathbf{1}^\top\big)$, computed once per group. For source i, the raw displacement Gram $K_{jk}^{(i)} = \langle d_{ij}, d_{ik} \rangle_a$ follows without forming any p-dimensional vector:

$$K_{jk}^{(i)} = \Gamma_{jk} - \Gamma_{ji} - \Gamma_{ik} + \Gamma_{ii}. \tag{16}$$

The bandwidth h_i is the median of the diagonal square roots $\|d_{ik}\|_a = \sqrt{K_{kk}^{(i)}}$, giving the locality weights of Section 3.2 with no additional passes over the data.

Weighted double centering. With $q_i = (q_{ij})_j$ the normalized squared weights and $\mathbf{1}$ the all-ones vector, the centered displacements $\widetilde{d}_{ij} = d_{ij} - \sum_k q_{ik} d_{ik}$ have Gram

$$\widetilde{K}^{(i)} = \big(I - \mathbf{1} q_i^\top\big) K^{(i)} \big(I - q_i \mathbf{1}^\top\big), \tag{17}$$

the weighted double centering used in Section 3.2. The response is centered by the same weights, $\widetilde{y}_{ij} = r_j - \sum_k q_{ik} r_k$.

Dual solution by the push-through identity. Write $A_i = W_i \widetilde{D}_i$. The primal normal equations $(A_i^\top A_i + \rho I) g = A_i^\top W_i \widetilde{y}_i$ live in p dimensions, but the identity $(A_i^\top A_i + \rho I)^{-1} A_i^\top = A_i^\top (A_i A_i^\top + \rho I)^{-1}$ moves the inverse to the $S \times S$ side:

$$g_c^\star(a_i) = \widetilde{D}_i^\top \alpha_i, \qquad \alpha_i = W_i \big(W_i \widetilde{K}^{(i)} W_i + \rho I\big)^{-1} W_i \widetilde{y}_i, \tag{18}$$

the dual form of ridge regression. Under the affine, noiseless model, $\widetilde{y}_i$ records the reward differential on the centered displacements, and as $\rho \to 0$ the Gram solve returns its Riesz representer on $\mathcal{U}_i$, whose metric is $\widetilde{K}^{(i)}$. The reconstruction is invariant to a common translation of the group's activations, since centered displacements annihilate constant shifts. For each sample group, we store the coefficient vectors α_i and the seed indices, and reconstruct teacher directions on demand as $\widetilde{D}_i^\top \alpha_i$. Teacher reconstruction uses the same group activations used to train the controller (Equation (4)). Storing the coefficients therefore requires S additional scalars per teacher, compared with p for an explicit direction.

Guarantees. The following results concern the ridge solution of Equation (3), which tends to the minimum-norm least-squares solution as $\rho \to 0$.

Proposition 3 (Affine-model recovery). *Let $\mathcal{U}_i = \operatorname{span}\{\widetilde{d}_{ij}\}_{j=1}^{S}$. If V_c is affine on the sampled neighborhood and $r_j = V_c(a_j)$ for every j, then $g_c^\star(a_i) = (C_i + \rho I)^{-1} C_i \nabla V_c(a_i)$ with $C_i = \widetilde{D}_i^\top W_i^2 \widetilde{D}_i$. It lies in $\mathcal{U}_i$ and tends to $P_{\mathcal{U}_i} \nabla V_c(a_i)$ as $\rho \to 0$.*

Lemma 4 (Finite-sample teacher error). *Write the centered observations as $\widetilde{y}_i = \widetilde{D}_i \nabla V_c(a_i) + b_i + \epsilon_i$, where b_i collects the Taylor remainders and ϵ_i the deviations of the observed rewards from V_c. With C_i as in Proposition 3 and σ_i its smallest nonzero eigenvalue,*

$$\left\| g_c^\star - P_{\mathcal{U}_i} \nabla V_c(a_i) \right\|_a \le \frac{\|\widetilde{D}_i^\top W_i^2 (b_i + \epsilon_i)\|_a + \rho \, \|P_{\mathcal{U}_i} \nabla V_c(a_i)\|_a}{\sigma_i + \rho}. \tag{19}$$

Proof of Proposition 3 and Lemma 4. Under the affine model, centered responses equal $\widetilde{D}_i \nabla V_c(a_i)$, so Equation (3) gives $(C_i + \rho I)^{-1} C_i \nabla V_c(a_i)$. The weights are positive, so $\operatorname{range}(C_i) = \operatorname{range}(\widetilde{D}_i^\top) = \mathcal{U}_i$: on each eigendirection of C_i in $\mathcal{U}_i$, with eigenvalue λ, the component of $\nabla V_c(a_i)$ is scaled by $\lambda/(\lambda + \rho)$, and the orthogonal complement is annihilated, which gives the limit $P_{\mathcal{U}_i} \nabla V_c(a_i)$. For the error, $g_c^\star - P_{\mathcal{U}_i} \nabla V_c(a_i) = \big((C_i + \rho I)^{-1} C_i - P_{\mathcal{U}_i}\big) \nabla V_c(a_i) + (C_i + \rho I)^{-1} \widetilde{D}_i^\top W_i^2 (b_i + \epsilon_i)$. The first operator acts on $\mathcal{U}_i$ with eigenvalues $-\rho/(\lambda + \rho)$, and the second term's argument lies in $\mathcal{U}_i$, where $(C_i + \rho I)^{-1}$ is bounded by $(\sigma_i + \rho)^{-1}$. □

J Reward-Flow Calibration (RFC): Rounds, Occupancy, and Guarantees

RFC alternates collection and update in outer rounds. Round n starts from θ_n and, for every seed state a_s, constructs the unroll

$$a_{n,s}^{(0)} = a_s, \qquad a_{n,s}^{(k+1)} = \mathcal{T}_{\theta_n}(a_{n,s}^{(k)}, c, \Delta s), \quad k = 0, \dots, K_{\text{train}} - 1, \tag{20}$$

with induced state occupancy

$$d_{\theta_n, c}^{K_{\text{train}}} = \frac{1}{K_{\text{train}}} \sum_{k=0}^{K_{\text{train}}-1} \big(\mathcal{T}_{\theta_n}(\cdot, c, \Delta s)^k\big)_\sharp \mu_c. \tag{21}$$

For each round's prompts and seed states, $\mathcal{T}_{\theta_n}$ is held fixed during collection and unrolled for K_{train} refinement steps, recording the visited states. At each recorded state the antithetic pair $a^\pm$ of Section 3.3 is decoded through the frozen suffix and scored, producing the continuous tanh labels defined there. Comparing an edited state against its unedited continuation would cost the same two decodes and leave an $O(h)$ bias set by the second directional derivative. Setting $\bar{\theta} = \theta_n$ in Equation (7), the controller is updated on the labeled batch, with a fraction β of RAM teacher targets replayed as an anchor. Finally, a held-out evaluation decides early stopping, and the next round collects fresh unrolls under the updated transition $\mathcal{T}_{\theta_{n+1}}$.

Reliability of the antithetic label.

Proposition 5 (When the antithetic label is reliable). *Let $v = \widehat{g}_\theta(a, c)$, so $\|v\|_a = 1$, let $D_v V_c(a) = \langle \nabla V_c(a), v \rangle_a$, and suppose $|D_v^3 V_c(a + sv)| \le M$ for $|s| \le h$. If the observed scores track the values of the probed states to $\big|(r^+ - r^-) - \big(V_c(a^+) - V_c(a^-)\big)\big| \le 2\nu$, then*

$$\left| \frac{r^+ - r^-}{2h} - D_v V_c(a) \right| \le \frac{M h^2}{6} + \frac{\nu}{h}. \tag{22}$$

Consequently, whenever $|D_v V_c(a)| > M h^2/6 + \nu/h$, the label has the correct sign.

Proof. Expanding $V_c(a \pm hv)$ about a along v, the even-order terms cancel in the difference, leaving $V_c(a^+) - V_c(a^-) = 2h D_v V_c(a) + \frac{h^3}{6}\big(D_v^3 V_c(\xi^+) + D_v^3 V_c(\xi^-)\big)$ for some $\xi^\pm$ on the segment, which contributes at most $M h^2/6$ after division by $2h$. The score deviation contributes at most ν/h. □

K VALUE GROWTH ALONG THE FLOW

Proposition 6 (First-order value of one step)**.** *Let V_c be differentiable at a with $\nabla V_c(a) \neq 0$, and let $d_c^\star(a) = \nabla V_c(a)/\|\nabla V_c(a)\|_a$. One step of Equation (5) changes value by*

$$V_c\big(\mathcal{T}_\theta(a, c, \Delta s)\big) - V_c(a) = \Delta s \ \mathrm{rms}(a)\, \|\nabla V_c(a)\|_a \, \big\langle d_c^\star(a), \widehat{g}_\theta(a, c)\big\rangle_a + o(\Delta s).$$

Relative to the oracle step along $d_c^\star$, the first-order gain lost is $\Delta s \ \mathrm{rms}(a)\|\nabla V_c(a)\|_a\big(1-\langle d_c^\star, \widehat{g}_\theta\rangle_a\big)$, so it is controlled by the direction-cosine error of the controller's predicted direction.

Proof. Taylor expansion of V_c at a along the displacement $\Delta s \ \mathrm{rms}(a)\widehat{g}_\theta(a, c)$, whose norm is $\Delta s \ \mathrm{rms}(a)$ since $\|\widehat{g}_\theta\|_a = 1$. The oracle value follows by substituting $d_c^\star$ and $\langle d_c^\star, d_c^\star\rangle_a = 1$. □

Let φ_s denote the flow of F_θ and $\rho_{c,s} = (\varphi_s)_\sharp \mu_c$ the transported activation measure. Differentiating $\mathbb{E}_{\rho_{c,s}}[V_c] = \mathbb{E}_{\mu_c}[V_c \circ \varphi_s]$ gives

$$\frac{\mathrm{d}}{\mathrm{d}s}\mathbb{E}_{\rho_{c,s}}[V_c] = \mathbb{E}_{\rho_{c,s}}\big[\mathcal{L}_{F_\theta} V_c\big], \qquad \mathcal{L}_{F_\theta} V_c := \langle \nabla V_c, F_\theta\rangle_a. \tag{23}$$

Proposition 7 (Value growth along the flow)**.** *Suppose ∇V_c is L-Lipschitz in $\|\cdot\|_a$ on a set containing the trajectory. Each Euler step of Equation (9) satisfies*

$$V_c\big(a^{(k+1)}\big) \geq V_c\big(a^{(k)}\big) + \Delta s \, \big\langle \nabla V_c(a^{(k)}), F_\theta(a^{(k)}, c)\big\rangle_a - \frac{L\,\Delta s^2}{2} \, \mathrm{rms}\,\big(a^{(k)}\big)^2,$$

so the step gains value whenever the alignment term exceeds the $O(\Delta s^2)$ curvature penalty. If moreover $\mathcal{L}_{F_\theta} V_c \geq 0$ on the occupancy, the expected value $\mathbb{E}_{\rho_{c,s}}[V_c]$ is non-decreasing along the continuous flow of F_θ.

Proof. The first claim is the standard descent lemma applied to $-V_c$ with update $\Delta s \, F_\theta$, using $\|F_\theta(a, c)\|_a = \mathrm{rms}(a)$ from the unit-RMS normalization. The second claim is the value-evolution identity in Equation (23): the time derivative of the expected value equals the expected Lie derivative, which is nonnegative by assumption. □

The antithetic probes of Section 3.3 estimate the alignment term used by both guarantees.