FROM PREFERENCE TO PROCESS: COUNTERFACTUAL TRAJECTORY REWARD MODELING FOR DIFFUSION RL

Anonymous authors
Paper under double-blind review

ABSTRACT

Terminal preference rewards guide diffusion reinforcement learning, but do not specify how gains should be allocated across interacting denoising transitions. We define counterfactual trajectory credit by selectively replacing stochastic transitions with deterministic updates and replaying their downstream continuations under a fixed policy and indexed noise tape. A frozen preference judge compares the resulting outcomes against a shared deterministic reference, and Shapley values average each intervention's marginal contribution across combinations of retained transitions. To avoid enumerating these outcomes during policy updates, we first adapt clean preference perception to noisy latents, then train the evaluator on offline counterfactual labels to predict step credits and terminal gain from completed trajectory pairs. The predicted credits determine bounded, nonnegative weights on a terminal group advantage. On Stable Diffusion 3.5 Medium, learned allocation improves GenEval from 0.704 to 0.753 while retaining the same learned terminal evaluator. Latent initialization reduces step-credit mean absolute error from 0.028 to 0.019, and permutation controls support assigning credits to their corresponding transitions. The results support learned counterfactual credit as useful supervision for allocating terminal feedback across denoising decisions.

1 INTRODUCTION

Diffusion reinforcement learning uses a preference score for a completed image to update the denoising decisions that produced it (Black et al., 2024; Liu et al., 2025). Yet a favorable image can arise from stochastic transitions with unequal, or even opposing, contributions to the outcome. Applying the same terminal advantage across steps leaves these differences implicit. We study how to use transition contributions to allocate the strength of policy updates across the denoising process.

Recent methods already provide temporal feedback. DenseGRPO and Stepwise-Flow-GRPO use rewards from intermediate completions to estimate successive gains (Deng et al., 2026; Savani et al., 2026); TempFlow and TreeGRPO evaluate exploratory continuations through deterministic suffixes or sampled branches (He et al., 2026; Ding & Ye, 2026). Beyond local gains and sampled-future feedback, we ask: *how much does retaining a stochastic transition, rather than replacing it with a deterministic update, contribute to terminal preference across different combinations of other retained transitions?* This question specifies both the alternative transition and the contexts over which its contribution is evaluated.

Endpoint scores alone do not uniquely determine this attribution. Figure 1(1) illustrates two games with identical baseline and terminal values but different step contributions. Replacing a transition also changes the states on which later transitions act, requiring replay of the affected continuation. A leave-one-out comparison measures its effect in only one context. When individual and joint effects coexist, averaging across contexts can change the ranking of contributions and hence the allocation of terminal advantages across steps (Section 6). This motivates a replay-defined target that accounts for interactions among denoising interventions.

Building on counterfactual and Shapley credit assignment (Cao et al., 2025; Li et al., 2026a), we define an intervention game for coupled stochastic differential equation (SDE) and ordinary differential equation (ODE) sampling. A short SDE window lies between ODE segments. Each coalition retains

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

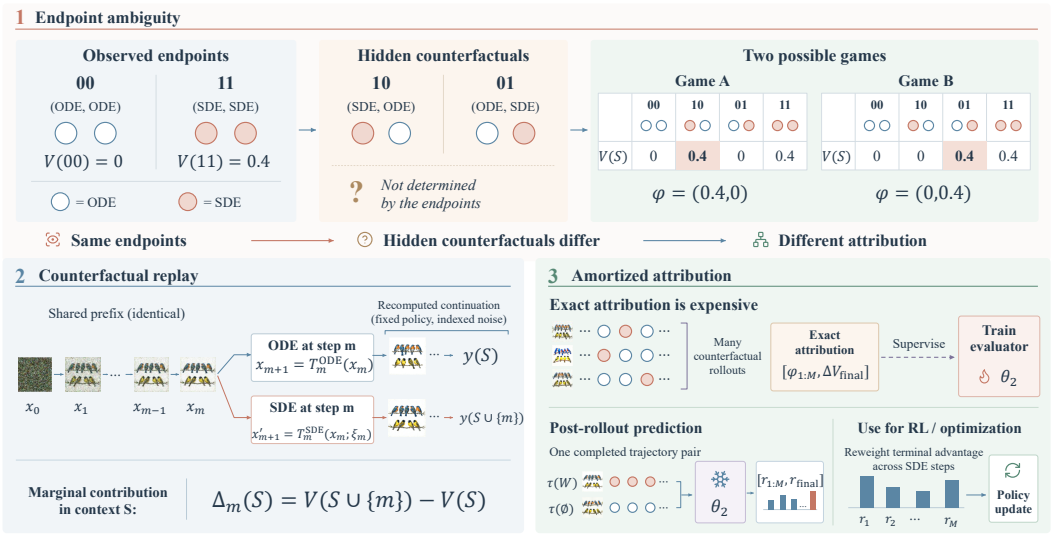


Figure 1: From endpoint preference to trajectory credit. (1) Two games share $V(00) = 0$ and $V(11) = 0.4$, but different hidden coalition values yield credits $(0.4, 0)$ and $(0, 0.4)$. (2) An ODE/SDE switch requires replaying the downstream continuation. (3) Counterfactual labels train a predictor used after rollout for credit-informed policy updates. The two-game example is algebraic; thumbnail placement illustrates the mechanism (Appendix A.5).

selected SDE transitions and replaces the others with ODE updates, recomputing every affected continuation under the same policy, prompt, initial noise, schedule, and indexed noise tape. The ODE replacement provides a reference for deterministic progression under the same policy. A frozen judge compares each outcome with a shared all-ODE image; Shapley values allocate the resulting preference-score gain by averaging marginal effects across coalition contexts.

We amortize this target to make it available for policy updates. Exact labels require 2^M completed outcomes for an M-step window. Following the principle of learned attribution (Jethani et al., 2022), we train a two-stage evaluator to predict the labels from one completed mixed/all-ODE pair. Stage I jointly trains a latent encoder and adapter for noisy-latent preference perception; Stage II learns step credits and terminal gain from paired histories and final images. Additional coalition outcomes supply offline supervision, while deployment relies on statistical regularities learned across labeled games. After rollout, the predicted credits set bounded, nonnegative weights on the terminal group advantage. The weights preserve its sign and mean across steps, assigning more reinforcement, or less penalty, to higher-credit transitions.

On Stable Diffusion 3.5 Medium, learned allocation improves GenEval from 0.704 to 0.753 with the same learned terminal evaluator, isolating the comparison from changes in terminal scoring. Latent initialization reduces Step MAE from 0.028 to 0.019 and raises within-window Spearman correlation from 0.51 to 0.70. A separate allocation diagnostic fixes the clean-teacher terminal value, while permutation controls test whether credits help when assigned to their corresponding transitions. In a direct human evaluation against TreeGRPO, our system receives 53.3% preference. These evaluations connect prediction of the replay-defined target to its use in policy updates and to generation quality.

Our contributions are:

- A replayable ODE/SDE game that attributes common-reference preference gains to interacting denoising interventions.
- A two-stage evaluator that learns from exact offline counterfactual targets to predict credits from completed trajectory pairs.
- A bounded advantage allocation with an error-sensitivity analysis at fixed terminal advantage, evaluated under fixed terminal evaluators and through credit-permutation controls.

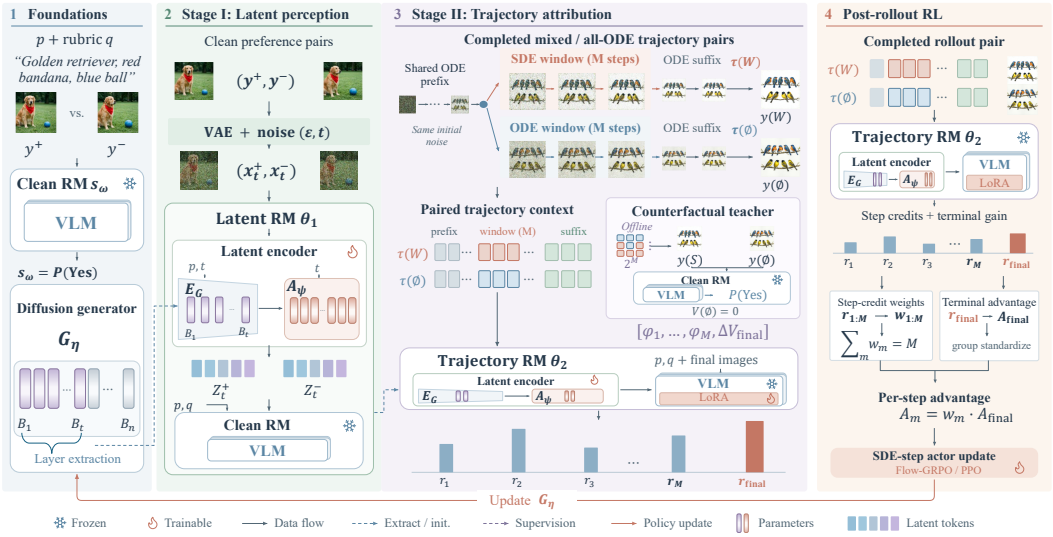


Figure 2: Two-stage trajectory reward modeling and post-rollout RL. A frozen clean judge scores replayed outcomes. Stage I jointly trains E_{G} and A_ψ on clean-pair preferences. Stage II continues training both with language LoRA on completed mixed/all-ODE pairs to predict a step/final array. The resulting evaluator is frozen for actor updates and supplies credits and a terminal value for group standardization. $\sum_m w_m = M$ preserves the mean scalar advantage, not its policy gradient. Thumbnail placement is schematic (Appendix A.5).

2 Related Work

Latent and temporal diffusion feedback. DiNa-LRM learns noisy-latent preferences, DRM compares sibling stochastic successors, and StitchVM transfers a clean evaluator to diffusion features (Liu et al., 2026; Zhang et al., 2026; Go et al., 2026). Completion gains and branching provide temporal feedback (Deng et al., 2026; Savani et al., 2026; He et al., 2026; Ding & Ye, 2026). STAR and CoCA allocate terminal feedback using attention or latent progression; PCPO corrects sampler-induced timestep weighting (Shen et al., 2026; Liao et al., 2026; Lee & Ye, 2026). Our Stage I builds on latent perception; Stage II changes the supervised target to each intervention's marginal contribution averaged over replayed coalition contexts, accounting for its interactions with other retained transitions.

Counterfactual and amortized attribution. RUDDER decomposes sequence returns (Arjona-Medina et al., 2019); SCAR applies Shapley allocation to preference rewards over masked text (Cao et al., 2025). Counterfactual Shapley Credit Assignment evaluates action coalitions (Li et al., 2026a), while FastSHAP trains an attribution predictor directly from coalition values (Jethani et al., 2022). We specialize this lineage to coupled ODE/SDE replacements with downstream replay, learning serialized credit targets from completed trajectory pairs. This differs from FastSHAP's coalition-regression objective. We evaluate attribution fidelity and subsequent advantage allocation separately. Appendix G provides the full comparison.

3 Counterfactual Trajectory Reward Modeling

We define a replayable attribution target and learn to predict it from completed mixed/reference trajectories. The evaluator is retrospective: it supplies credits after rollout and before an actor update. The target separates a trajectory's total preference change from the allocation of that change among its stochastic interventions. Figure 2 summarizes the pipeline.

3.1 A replayable ODE/SDE intervention game

Fix policy snapshot $\bar{\eta}$, prompt p, initial noise, solver schedule, and a step-indexed noise tape $\xi_{1:M}$. Let $W = \{1, \ldots, M\}$ index the intervention window between an ODE prefix and suffix. A coalition

$S \subseteq W$ uses SDE at its selected steps and ODE elsewhere:

$$x_{m+1}(S) = \begin{cases} T^{\mathrm{SDE}}_{\bar{\eta},m}(x_m(S); p, \xi_m), & m \in S, \\ T^{\mathrm{ODE}}_{\bar{\eta},m}(x_m(S); p), & m \notin S. \end{cases} \tag{1}$$

Every coalition recomputes downstream states and the suffix from its own parent states; omitting an SDE step never reindexes later noise draws. Pasting a state from the full trajectory into a reduced coalition would violate this replay rule. All continuations use the frozen snapshot. The switch replaces the entire transition map, including any SDE drift correction, so credit is relative to this ODE replacement. Write $y(S)$ for the decoded outcome and $y^{\mathrm{ODE}} = y(\emptyset)$ for the shared all-ODE reference.

3.2 Teacher attribution as the prediction target

A frozen preference-trained vision–language model (VLM) scores candidate y^a against reference y^b under prompt p and rubric q:

$$s_\omega(y^a, y^b; p, q) = P_\omega(Y \mid p, q, E_{\mathrm{V}}(y^a), E_{\mathrm{V}}(y^b)), \tag{2}$$

where E_{V} is its native visual encoder and Y is the complete affirmative answer event (Appendix A). The same frozen judge and reference define

$$V(S) = s_\omega(y(S), y^{\mathrm{ODE}}; p, q) - s_\omega(y^{\mathrm{ODE}}, y^{\mathrm{ODE}}; p, q), \qquad V(\emptyset) = 0. \tag{3}$$

The stored self-score sets the origin without assuming probability $1/2$ or calibrating the preference scale. Shapley credit averages each step's marginal over all coalition contexts:

$$\phi_m = \sum_{S \subseteq W \setminus \{m\}} \frac{|S|!(M - |S| - 1)!}{M!} [V(S \cup \{m\}) - V(S)], \qquad \sum_{m=1}^{M} \phi_m = \Delta V_{\mathrm{final}}, \tag{4}$$

with $\Delta V_{\mathrm{final}} = V(W)$ and target array $\mathbf{a} = [\phi_1, \ldots, \phi_M, \Delta V_{\mathrm{final}}]$. This is attribution of the chosen teacher statistic for a realized noise-tape game, rather than cardinal human utility; nonlinear score transformations can change credit signs (Appendix A.4). Exact offline labels require 2^M coalition completions, including eight for $M = 3$; shared prefixes reduce repeated computation. Appendix A.1 and Figure 6 detail exact construction; permutation estimates average complete marginal chains, which telescope to the same total while retaining coordinate-wise sampling uncertainty. These are teacher-derived process targets; no human annotates individual denoising steps. Figure 3A illustrates why a single replacement marginal differs from the context-averaged credit.

3.3 Two-stage learning from latent and trajectory evidence

Stage I: latent preference perception. For a clean preference pair (y^+, y^-), the generator's VAE produces latents $z^\pm$, and its native noise schedule gives $x_t^\pm = \alpha_t z^\pm + \sigma_t \epsilon^\pm$. The reward-side encoder E_{G} and adapter A_ψ map noisy latents and timestep information to tokens for a preference-trained Qwen3.5-9B VLM. Both train on the source pair's preference label to obtain θ_1; the preference backbone and native head stay frozen. Latent tokens replace image inputs with prompt and rubric unchanged. All noisy views of a source pair remain in one data split. Appendix B.1 details noise coupling, pair-order randomization, timestep-stratified evaluation, and the preference loss.

Stage II: completed trajectory evaluation. Starting from θ_1, Stage II continues training E_{G} and A_ψ with language LoRA on offline targets ($\theta \equiv \theta_2$). Its context $\mathcal{C}$ contains prompt, rubric, shared prefix, both window paths and suffixes, and both final images, with time, step, ODE/SDE role, and pre/post-state evidence. The trained reward encoder may differ from the changing actor; activation reuse requires matching encoders and preprocessing. The shared language-model head generates

$$P_\theta(\mathbf{v} \mid \mathcal{C}) = \prod_{\ell=1}^{L} P_\theta(v_\ell \mid \mathcal{C}, v_{<\ell}), \quad \mathrm{parse}(\mathbf{v}) = [r_1, \ldots, r_M, r_{\mathrm{final}}]. \tag{5}$$

The step coordinates approximate $\boldsymbol{\phi}$, while the final coordinate predicts $V(W)$; these roles remain distinct in the actor update. For target $\mathbf{a}$ in Eq. 4, a fixed numeric quantizer Q_δ and chronological

serialization give $\mathbf{v}^* = \mathrm{serialize}(Q_\delta(\mathbf{a}))$. Stage II minimizes the token negative log likelihood

$$\mathcal{L}_{\mathrm{II}} = -\mathbb{E} \sum_{\ell=1}^{L} \log P_\theta(v_\ell^* \mid \mathcal{C}, v_{<\ell}^*). \tag{6}$$

The clean teacher remains frozen; additional coalition outcomes are used only to construct training labels. Appendix B.2 details the context and output contract. At deployment, one autoregressive call uses the completed pair, without unseen coalition outcomes. Paired collection and reward-side encoding remain costs (Appendix E). The two observed paths support statistical prediction of the offline target: the model learns regularities across labeled games, rather than reconstructing unseen coalition outcomes uniquely from each pair (Appendix A.3). Since the decoder does not enforce Shapley efficiency, we evaluate step accuracy, terminal accuracy, and output validity separately.

4 Post-Rollout Advantage Reweighting

The frozen evaluator scores completed trajectories before actor updates. For $G \geq 2$ trajectories sharing a prompt, initial noise, and all-ODE baseline but using independent SDE tapes, group standardization (Liu et al., 2025) gives

$$A_{i,\mathrm{final}} = \frac{r_{i,\mathrm{final}} - \mu}{\sigma + \varepsilon}, \quad \mu = G^{-1} \sum_i r_{i,\mathrm{final}}, \quad \sigma^2 = G^{-1} \sum_i (r_{i,\mathrm{final}} - \mu)^2. \tag{7}$$

We redistribute this advantage across the M SDE steps using

$$\begin{aligned} u_{i,m} &= \frac{\mathrm{sign}(A_{i,\mathrm{final}}) r_{i,m}}{\sqrt{M^{-1} \sum_k r_{i,k}^2} + \varepsilon}, \\ w_{i,m} &= (1-\lambda) + \lambda M \, \frac{\exp(u_{i,m}/T)}{\sum_k \exp(u_{i,k}/T)}, \qquad A_{i,m} = w_{i,m} A_{i,\mathrm{final}}, \end{aligned} \tag{8}$$

where $T > 0$, $\lambda \in [0, 1]$, and $\varepsilon > 0$. Nonnegative weights preserve each trajectory's terminal-advantage sign and satisfy

$$\sum_m w_{i,m} = M, \qquad M^{-1} \sum_m A_{i,m} = A_{i,\mathrm{final}}. \tag{9}$$

The weights lie in $[1-\lambda,\ 1+(M-1)\lambda]$: λ bounds the redistribution strength, while T controls its concentration. Setting $\lambda = 0$ or assigning constant credits recovers uniform allocation. When $\lambda > 0$, for $A_{i,\mathrm{final}} > 0$ a higher credit receives more reinforcement. For $A_{i,\mathrm{final}} < 0$, it receives a smaller penalty; the sign factor in $u_{i,m}$ reverses the weight ordering before multiplication by the negative advantage (Appendix D). Each step still retains the trajectory's terminal sign: a negative credit cannot reverse a positive terminal advantage. The final value enters the reward once; adding step credits would double-count the exact target's total.

Actor update and reference collection. Only SDE transitions with nondegenerate densities enter the clipped PPO update (Schulman et al., 2017). Define the likelihood ratio $\rho_{i,m} = \pi_\eta(x_{m+1}^i \mid x_m^i, p) / \pi_{\bar{\eta}}(x_{m+1}^i \mid x_m^i, p)$. The actor minimizes

$$\begin{aligned} \mathcal{L}_\pi = \; & -\mathbb{E}_{i,m} \min\{\rho_{i,m} \, \mathrm{sg}[A_{i,m}], \\ & \mathrm{clip}(\rho_{i,m}, 1-\epsilon_{\mathrm{clip}}, 1+\epsilon_{\mathrm{clip}}) \, \mathrm{sg}[A_{i,m}]\} + \kappa \mathcal{L}_{\mathrm{ref}}, \end{aligned} \tag{10}$$

where sg stops gradients, $0 < \epsilon_{\mathrm{clip}} < 1$ is the clipping radius (distinct from ε), and $\mathcal{L}_{\mathrm{ref}}$ regularizes toward a reference policy. The ODE segments supply context and outcomes. The paired ODE reference stays fixed within a collection block and is recomputed when the behavior snapshot changes (Appendix B.3). Equation 9 conserves the scalar budget, not the policy gradient: step gradients differ and credits use future evidence.

From credit error to allocation error. For predicted credits $\mathbf{r}_i$ and exact credits $\boldsymbol{\phi}_i$, write $a_i = \|\mathbf{r}_i\|_2/\sqrt{M}$, $b_i = \|\boldsymbol{\phi}_i\|_2/\sqrt{M}$, and $e_i = \|\mathbf{r}_i - \boldsymbol{\phi}_i\|_2/\sqrt{M}$. Holding $A_i = A_{i,\mathrm{final}}$, T, ε, and λ fixed for both credit vectors gives

$$\|w(\mathbf{r}_i, A_i) - w(\boldsymbol{\phi}_i, A_i)\|_1 \leq \lambda M \min\left\{2, \frac{\sqrt{M/2}\, e_i}{T\sqrt{(a_i+\varepsilon)(b_i+\varepsilon)}}\right\}. \tag{11}$$

At $A_i = 0$, both allocations are uniform. The bound explains why low absolute Step MAE can coexist with poor allocation when credit magnitudes are small or temperature is low. It motivates the mean weight discrepancy E_w (Appendix B.4) and bounds perturbation of the chosen exact-credit surrogate at fixed data and parameters. The proof in Appendix D concerns update fidelity, not policy improvement, and excludes terminal-value errors and changes to group normalization.

5 EVALUATION

We evaluate attribution fidelity, allocation under a fixed terminal evaluator, and end-to-end generation quality and learning time.

Experimental setup. We use Stable Diffusion 3.5 Medium (SD3.5-M) as the generator, as in Flow-GRPO and TreeGRPO (Liu et al., 2025; Ding & Ye, 2026). Our reward model is trained from Qwen3.5-9B (Qwen Team, 2026) and adapted using the two-stage procedure in Section 3. We use the same random seed across experiments. For human evaluation, 30 experts assess 100 image pairs. Win rate is the percentage of judgments favoring the evaluated method; ties remain in the denominator and count as non-wins.

Evaluation design. Human win rates compare against teacher-uniform in the allocation diagnostic, a fixed mixed-window reference (Mix) in the system table, and TreeGRPO in a separate direct comparison.

The allocation diagnostic fixes the clean teacher terminal while varying credit rules. Within the learned system, the $\lambda = 0$ arm retains the predicted terminal output but removes step-dependent weighting. These controls separate allocation from terminal evaluation; full-system comparisons assess their combination alongside alternative methods.

5.1 ATTRIBUTION UNDER A TEACHER-TERMINAL ANCHOR

Table 1: Fixed clean-teacher terminal, varying step allocation. Human win rates are against teacher-uniform (self-reference: 50.0%). Dark/light pink marks the best/second-best value per metric column.

Allocation	Human vs teacher-uniform (%)
Uniform teacher-terminal control	50.0
Leave-one-out	54.0
Exact Shapley	**57.4**
Learned credit	56.1

Table 1 holds the terminal value at the clean teacher while varying allocation. Exact Shapley reaches 57.4% preference against teacher-uniform, compared with 54.0% for leave-one-out and 56.1% for learned credit. Thus the learned predictor approaches the exact-credit result in this diagnostic, with a 1.3 percentage-point gap between their common-reference rates. The permutation comparison in Figure 3C separately tests the assignment of credits to transitions.

5.2 ATTRIBUTION PREDICTION AND LATENT INITIALIZATION

Table 2: Stage II prediction of the Shapley array. Spearman is within-window correlation; Top is the accuracy of selecting the largest absolute step contribution. All rows target the same array task. StitchVM-style + Shapley is a constructed initialization control.

Method	Step MAE ↓	Spearman ↑	Sign (%) ↑	Top (%) ↑	Final MAE ↓
Final-image-only	0.034	0.28	66.1	49.0	**0.012**
Endpoint-feature-matched	0.031	0.36	69.2	54.4	0.013
Diffusion-native temporal	0.027	0.49	74.1	60.5	0.017
StitchVM-style + Shapley	0.025	0.55	77.4	64.0	0.016
Ours Stage II only	0.028	0.51	75.3	62.2	0.015
Ours Stage I + II	**0.019**	**0.70**	**83.2**	**74.6**	0.014

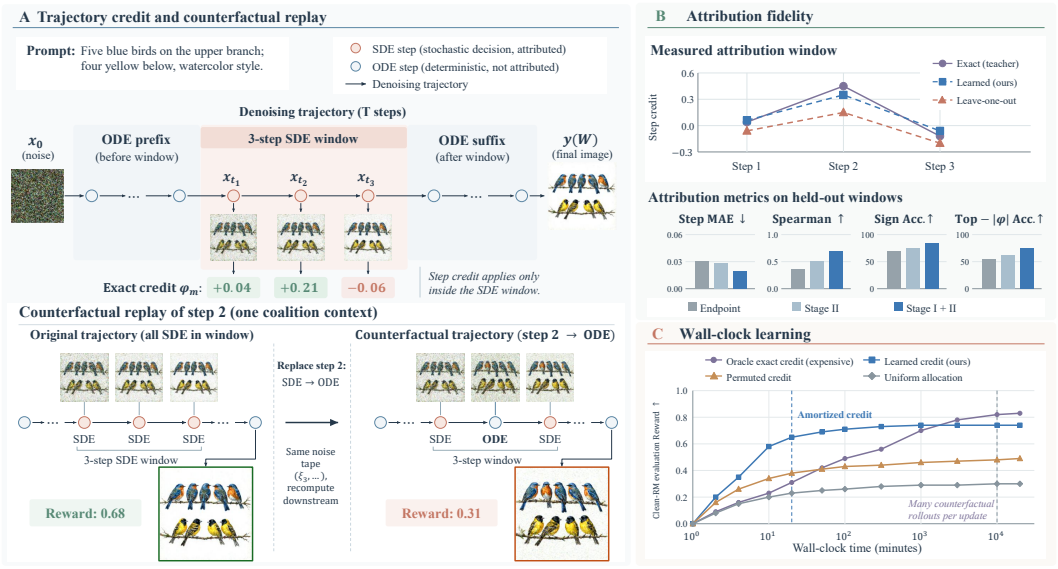


Figure 3: Measured trajectory credit, attribution fidelity, and learning. (A) Exact credits $(0.04, 0.21, -0.06)$ for a three-step window. Replacing step 2 changes reward from 0.68 to 0.31; this single-context marginal differs from its Shapley credit 0.21. (B) Exact, learned, and leave-one-out credits on another window (top), and the Endpoint-feature-matched, Stage II, and Stage I+II results from Table 2 (bottom). (C) Measured Clean-RM reward for final images against a shared fixed reference, versus wall-clock time: learned credit improves reward earlier; exact credit reaches higher reward later. Appendix A.5 explains the numerical comparisons.

All learners in Table 2 predict the same step/final array. The image-only and endpoint-feature controls assess prediction from end-of-trajectory evidence; the temporal learner and initialization variants examine trajectory representations and latent pretraining. Step MAE measures absolute prediction error, within-window Spearman measures relative ordering, and sign and largest-absolute-step accuracies capture which interventions help or dominate the teacher's score change. Final MAE evaluates the separately predicted terminal coordinate.

Stage I+II obtains 0.019 Step MAE and 0.70 within-window Spearman, versus 0.028 and 0.51 for Stage II alone (Table 2). The lower bars in Figure 3B visualize these aggregate comparisons with the endpoint-feature baseline. Sign and largest-absolute-step accuracies are 83.2% and 74.6%. The endpoint-feature learner obtains 0.031 Step MAE and 0.36 Spearman. The final-image-only learner has the lowest Final MAE, 0.012 versus 0.014 for ours, but a higher Step MAE of 0.034. Better endpoint prediction therefore does not imply better contribution prediction.

Role of latent initialization. The Stage I preference benchmark appears in Appendix C, Table 4. Its macro accuracy is 75.175%, versus 74.800% for same-clean-checkpoint StitchVM-style transfer. The larger downstream attribution improvement supports its role as initialization rather than a claim of uniform preference-model superiority. Figure 5(a) reports declining accuracy with noise, from 76.2% at $t = 0.1$ to 55.5% at $t = 0.9$. The first tested budgets attaining Step MAE ≤ 0.025 in Figure 5(b) are 5k windows with Stage I and 20k without it, a fourfold difference in the number of labeled Stage II windows needed to reach this threshold on the tested grid. Together, the results support latent adaptation as an initialization for trajectory credit.

5.3 Trajectory Replay and Wall-Clock Learning

Figure 3A reports positive credits for steps 1 and 2 and negative credit for step 3, summing to a terminal gain of 0.19. The step-2 replacement reduces the reward by 0.37 in the full-coalition context; averaging across contexts gives its Shapley credit of 0.21. In the separate window in panel B, learned credits track the exact attribution more closely than leave-one-out credits.

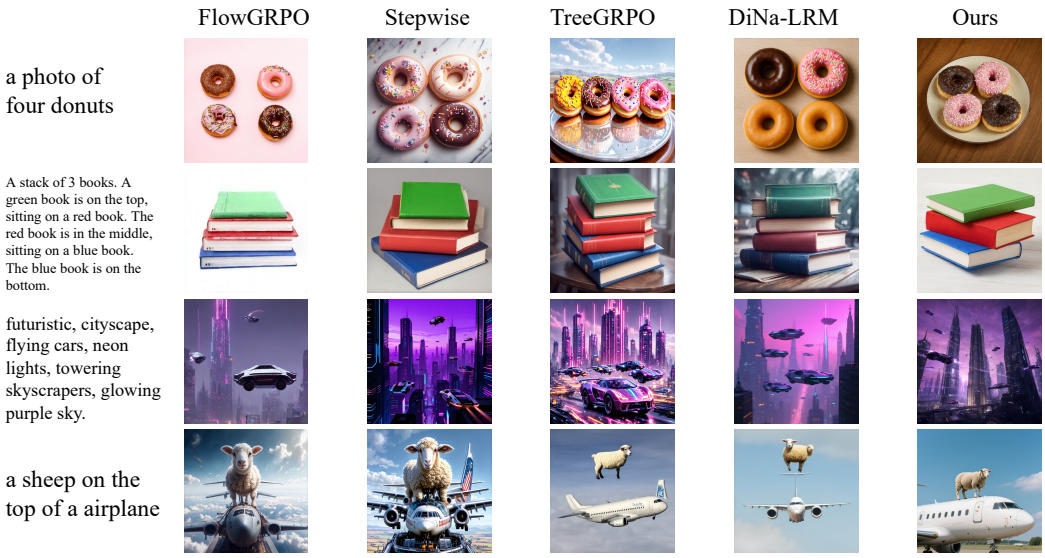


Figure 4: Qualitative generation results. Columns compare FlowGRPO, Stepwise, TreeGRPO, DiNa-LRM, and Ours under the prompts shown at left. Rows cover four-donut counting, a three-book stack with specified color order, a futuristic cityscape, and a sheep on an airplane. The examples show object count, attribute binding, scene composition, and spatial placement. Additional results appear in Appendix F, Figure 7.

Panel C plots evaluation reward from the clean reward model (Clean RM) against measured wall-clock time. The same Clean RM scores final images against a fixed reference image shared by all compared methods. This reward differs from the GenEval metric in Table 3. Learned credit improves reward earlier than exact attribution, which overtakes it later. After the common starting point, learned credit also exceeds permuted credit and uniform allocation, supporting the value of assigning credits to their corresponding transitions under this evaluator.

5.4 Complete-system comparisons

Table 3: End-to-end generation results. Human evaluation reports pairwise win rates against the same fixed mixed-window reference (Mix); 50.0 is the self-reference anchor. HPS denotes HPSv2.1 multiplied by 100.

Method	Human vs Mix (%) ↑	GenEval ↑	HPS ↑	ImageReward ↑
Flow-GRPO	48.7	0.694	29.8	1.13
Mixed-window uniform	50.0	0.706	30.0	1.17
DenseGRPO-style gain	52.1	0.722	30.2	1.22
Stepwise	53.6	0.729	30.2	1.23
TempFlow	54.8	0.736	30.4	1.25
TreeGRPO	55.5	0.741	30.5	1.28
DiNa-LRM-based process	53.9	0.727	**30.8**	1.24
Ours array, $\lambda = 0$	49.6	0.704	30.0	1.17
Ours array, learned allocation	**58.2**	**0.753**	30.6	**1.31**

Table 3 reports GenEval for text–image alignment, HPSv2.1 and ImageReward for automatic preference scoring, and human win rates against Mix (Ghosh et al., 2023; Wu et al., 2023; Xu et al., 2023). Learned allocation reaches 0.753 GenEval (mixed-window uniform: 0.706; TreeGRPO: 0.741) and the highest ImageReward, 1.31. DiNa-LRM has the highest HPS, 30.8 versus 30.6 for ours.

With the learned terminal output retained but $\lambda = 0$, GenEval is 0.704, ImageReward 1.17, and preference against Mix 49.6%; learned allocation obtains 0.753, 1.31, and 58.2%, respectively. These rows compare uniform and learned allocation within the learned-array system. A separate direct evaluation reports 53.3% preference for ours against TreeGRPO.

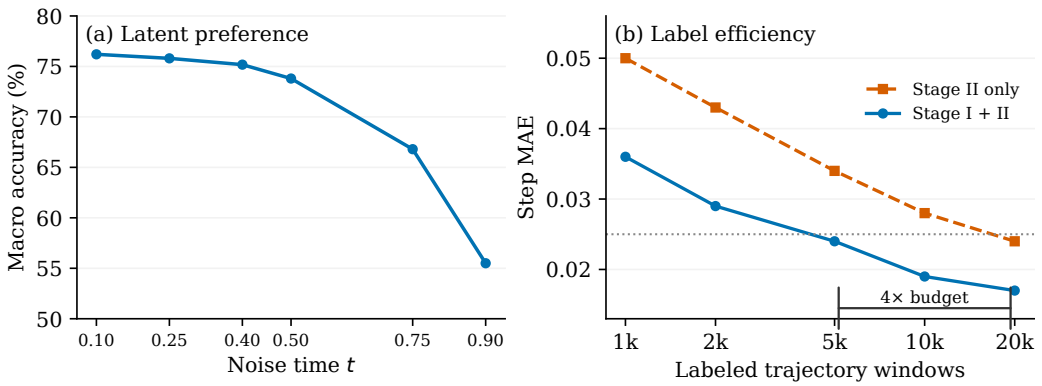


Figure 5: Latent perception and attribution initialization. (a) Preference accuracy by noise level. (b) Step MAE over tested label budgets; the dotted line marks 0.025. The first passing budgets are 5k and 20k windows. Lines connect observed aggregate points.

Qualitative comparison. Figure 4 illustrates counting, ordered attributes, spatial placement, and scene composition. Appendix F adds text rendering and material appearance.

Output validity and policy transfer. Array parsing succeeds for 99.6% of outputs. On an out-of-distribution policy, Step MAE decreases from 0.026 to 0.021 after refresh (Appendix C), separating format validity from numerical accuracy under policy change.

6 Discussion

When coalition context changes allocation. The value of averaging coalition contexts depends on how interactions affect the update, not merely on whether interactions exist. For a pure order-k interaction, leave-one-out credits equal k times the Shapley credits; RMS normalization largely removes this common scale when the stabilizer is small. Mixed-order effects can instead change which step receives the largest weight. For example, an individual contribution of 0.10 at step 1 and a joint contribution of 0.15 from steps 2 and 3 give Shapley credits $(0.10, 0.075, 0.075)$, but leave-one-out credits $(0.10, 0.15, 0.15)$. This algebraic example reverses the highest-credit step without changing the intervention game (Appendix A.2). It motivates evaluating the allocation induced by predicted credit, alongside attribution error alone.

What useful credit captures. The learned-versus-permuted Clean-RM reward comparison in Figure 3C supports assigning credits to the corresponding transitions. Output permutation preserves the weight multiset and, under uniform permutation, recovers the uniform clipped surrogate in expectation at fixed data and parameters (Appendix D.2). Its training trajectory can nevertheless differ. The measured advantage of learned credit over permutation therefore complements prediction accuracy: useful allocation requires attaching the predicted coordinates to the transitions whose contributions they represent.

Limitations and future directions appear in Appendix H.

7 Conclusion

Counterfactual trajectory reward modeling attributes preference gains to interacting ODE/SDE interventions through downstream replay. A two-stage evaluator predicts these offline targets from completed trajectory pairs, and bounded reweighting allocates terminal feedback using the predicted credits. On Stable Diffusion 3.5 Medium, learned allocation improves GenEval from 0.704 to 0.753 with the same learned terminal evaluator. Attribution and permutation diagnostics further assess credit fidelity and transition assignment, supporting replay-defined attribution as useful supervision for diffusion policy updates.

AI USE STATEMENT

Generative AI tools were used to assist with literature exploration and summarization, language editing and polishing, suggestions on manuscript organization and presentation, and the design and refinement of scientific figures. All AI-assisted outputs were critically reviewed, verified, and revised by the authors before being incorporated into the paper. The authors take full responsibility for the scientific content, claims, analyses, and presentation of the final manuscript.

REFERENCES

Jose A. Arjona-Medina, Michael Gillhofer, Michael Widrich, Thomas Unterthiner, Johannes Brandstetter, and Sepp Hochreiter. RUDDER: Return decomposition for delayed rewards. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/16105fb9cc614fc29e1bda00dab60d41-Abstract.html.

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

Meng Cao, Shuyuan Zhang, Xiao-Wen Chang, and Doina Precup. SCAR: Shapley credit assignment for more efficient RLHF. arXiv preprint arXiv:2505.20417, 2025. URL https://arxiv.org/abs/2505.20417.

Hugh Chen, Scott M. Lundberg, and Su-In Lee. Explaining a series of models by propagating shapley values. *Nature Communications*, 13:4512, 2022. doi: 10.1038/s41467-022-31384-3. URL https://www.nature.com/articles/s41467-022-31384-3.

Haoyou Deng, Keyu Yan, Chaojie Mao, Xiang Wang, Yu Liu, Changxin Gao, and Nong Sang. DenseGRPO: From sparse to dense reward for flow matching model alignment. In *International Conference on Learning Representations*, 2026. URL https://iclr.cc/virtual/2026/poster/10007484.

Zheng Ding and Weirui Ye. TreeGRPO: Tree-advantage GRPO for online RL post-training of diffusion models. In *International Conference on Learning Representations*, 2026. URL https://iclr.cc/virtual/2026/poster/10011603.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. GenEval: An object-focused framework for evaluating text-to-image alignment. In *Advances in Neural Information Processing Systems*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/a3bf71c7c63f0c3bcb7ff67c67b1e7b1-Abstract-Datasets_and_Benchmarks.html.

Hyojun Go, Hyungjin Chung, Prune Truong, Goutam Bhat, Li Mi, Zhaochong An, Zixiang Zhao, Dominik Narnhofer, Serge Belongie, Federico Tombari, and Konrad Schindler. Stitched value model for diffusion alignment. arXiv preprint arXiv:2605.19804, 2026. URL https://arxiv.org/abs/2605.19804.

Yuanshen Guan, Zipeng Feng, Chengru Song, Zhiwei Xiong, and Peiqin Sun. Latent reward registers for diffusion preference alignment. arXiv preprint arXiv:2608.03929, 2026. URL https://arxiv.org/abs/2608.03929.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330. PMLR, 2017. URL https://proceedings.mlr.press/v70/guo17a.html.

Xiaoxuan He, Siming Fu, Yuke Zhao, Wanli Li, Jian Yang, Dacheng Yin, Fengyun Rao, and Bo Zhang. TempFlow-GRPO: When timing matters for GRPO in flow models. In *International Conference on Learning Representations*, 2026. URL https://iclr.cc/virtual/2026/poster/10011263.

Neil Jethani, Mukund Sudarshan, Ian Covert, Su-In Lee, and Rajesh Ranganath. FastSHAP: Real-time shapley value estimation. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2107.07436.

Jeongjae Lee and Jong Chul Ye. PCPO: Proportionate credit policy optimization for preference alignment of image generation models. In *International Conference on Learning Representations*, pp. 126948–126982, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/ce182e31662883d4decc84a0255335b6-Abstract-Conference.html.

Junzhe Li, Yutao Cui, Tao Huang, Weijie Kong, Yiming Cheng, Chuxuan Zeng, Yinping Ma, Chun Fan, Miles Yang, Zhao Zhong, and Liefeng Bo. MixGRPO: Unlocking flow-based GRPO efficiency with mixed ODE-SDE. arXiv preprint arXiv:2507.21802, 2025. URL https://arxiv.org/abs/2507.21802. Accepted at ECCV 2026.

Mingxuan Li, Kai-Zhan Lee, and Elias Bareinboim. Counterfactual shapley credit assignment. *Reinforcement Learning Journal*, 7, 2026a. doi: 10.48550/arXiv.2607.16999. URL https://rlj.cs.umass.edu/2026/papers/Paper45.html.

Rui Li, Yuanzhi Liang, Ke Hao, Ziqiao Weng, Haibin Huang, Chi Zhang, and XueLong Li. Sample-adaptive latent rewards for uncertainty-guided diffusion post-training. arXiv preprint arXiv:2608.06125, 2026b. URL https://arxiv.org/abs/2608.06125.

Yuming Li, Yikai Wang, Yuying Zhu, Zhongyu Zhao, Ming Lu, Qi She, and Shanghang Zhang. BranchGRPO: Stable and efficient GRPO with structured branching in diffusion models. In *International Conference on Learning Representations*, 2026c. URL https://iclr.cc/virtual/2026/poster/10009356.

Xinyao Liao, Wei Wei, Xiaoye Qu, Qiyuan He, Angela Yao, and Yu Cheng. CoCA: Step-level reward for free in RL-based T2I diffusion model fine-tuning. arXiv preprint arXiv:2505.19196, 2026. URL https://arxiv.org/abs/2505.19196.

Pengyang Ling, Jiazi Bu, Yujie Zhou, Yibin Wang, Zhenyu Hu, Zihan Zhang, Yi Jin, Huaian Chen, and Yuhang Zang. Pave-GRPO: Beyond instantaneous guidance through principled average velocity decomposition. arXiv preprint arXiv:2606.01636, 2026. URL https://arxiv.org/abs/2606.01636.

Gongye Liu, Bo Yang, Yida Zhi, Zhizhou Zhong, Lei Ke, Didan Deng, Han Gao, Yongxiang Huang, Kaihao Zhang, Hongbo Fu, and Wenhan Luo. Beyond VLM-based rewards: Diffusion-native latent reward modeling. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.11146.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-GRPO: Training flow matching models via online RL. In *Advances in Neural Information Processing Systems*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/3a10c46572628d58cb44fb705f25cbbf-Abstract-Conference.html.

Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. HPSv3: Towards wide-spectrum human preference score. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15086–15095, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Ma_HPSv3_Towards_Wide-Spectrum_Human_Preference_Score_ICCV_2025_paper.html.

Thomas Mesnard, Théophane Weber, Fabio Viola, Shantanu Thakoor, Alaa Saade, Anna Harutyunyan, Will Dabney, Tom Stepleton, Nicolas Heess, Arthur Guez, Éric Moulines, Marcus Hutter, Lars Buesing, and Rémi Munos. Counterfactual credit assignment in model-free reinforcement learning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 7654–7664, 2021. URL https://proceedings.mlr.press/v139/mesnard21a.html.

Alexander Meulemans, Simon Schug, Seijin Kobayashi, Nathaniel D. Daw, and Gregory Wayne. Would i have gotten that reward? long-term credit assignment by counterfactual contribution analysis. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://papers.nips.cc/paper_files/paper/2023/file/d8bd445c2abe1343cce0e14b361b2fb3-Paper-Conference.pdf.

Qwen Team. Qwen3.5: Towards native multimodal agents. Official model-family release, 2026. URL https://qwen.ai/blog?id=qwen3.5.

Yash Savani, Branislav Kveton, Yuchen Liu, Yilin Wang, Jing Shi, Subhojyoti Mukherjee, Nikos Vlassis, and Krishna Kumar Singh. Stepwise credit assignment for GRPO on flow-matching models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://arxiv.org/abs/2603.28718.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017. URL https://arxiv.org/abs/1707.06347.

Lloyd S. Shapley. A value for n-person games. In Harold W. Kuhn and Albert W. Tucker (eds.), *Contributions to the Theory of Games II*, volume 28 of *Annals of Mathematics Studies*, pp. 307–317. Princeton University Press, 1953. doi: 10.1515/9781400881970-018.

Jinjie Shen, Wei Deng, Xian Hu, Daiguo Zhou, and Jian Luan. STAR: Spatiotemporal adaptive reward allocation for text-to-image RL post-training. arXiv preprint arXiv:2606.17979, 2026. URL https://arxiv.org/abs/2606.17979.

TIGER-AI-Lab. GenAI-Bench: Evaluation code and human-preference data. Official project repository, 2024. URL https://github.com/TIGER-AI-Lab/GenAI-Bench.

Jie Wu, Yu Gao, Zilyu Ye, Ming Li, Liang Li, Hanzhong Guo, Jie Liu, Zeyue Xue, Xiaoxia Hou, Wei Liu, Yan Zeng, and Weilin Huang. Rewarddance: Reward scaling in visual generation. arXiv preprint arXiv:2509.08826, 2025. URL https://arxiv.org/abs/2509.08826.

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

Xiaole Xian, Xilin He, Wenting Chen, Wenshuang Liu, Wenqi Mu, Yancheng He, Liang Li, Yi Zhang, and Xiangyu Yue. Consistent noisy latent rewards for trajectory preference optimization in diffusion models. In *International Conference on Learning Representations*, pp. 6315–6340, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/0b408293619f725fd30162af057e531a-Abstract-Conference.html.

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

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

Jaxon Zhang, Binxin Yang, Hubery Yin, Chen Li, and Jing Lyu. DRM: Diffusion-based reward model with step-wise guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://arxiv.org/abs/2605.25661.

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 *Advances in Neural Information Processing Systems*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/98d112a7111970b1638dafc97d96d9ee-Abstract-Conference.html.

A TEACHER SCORING AND COUNTERFACTUAL LABELS

Answer events. For a disjoint, prefix-free set Y of affirmative answer sequences with explicit termination, the clean score is

$$P_\omega(Y \mid c) = \sum_{v \in Y} \prod_{\ell=1}^{|v|} P_\omega(v_\ell \mid c, v_{<\ell}), \qquad c = (p, q, y^a, y^b). \tag{12}$$

A single-token score equals this event probability only when the exact tokenizer represents the complete verdict as one token, including its whitespace and capitalization. The raw affirmative probability and the conditional ratio $P(Y)/(P(Y)+P(N))$ are different statistics; switching between them changes the attribution target. The scoring rule uses a common template, preprocessing, answer events, and statistic across all coalitions. The same stored self-comparison score centers every member of a family.

A.1 EXACT AND PERMUTATION CONSTRUCTION

Figure 6 summarizes the replay rule, exact teacher, and amortized predictor. Its shorthand $w(S) = |S|!(M - |S| - 1)!/M!$ denotes the Shapley coefficient, and $\Delta_m V(S) = V(S \cup \{m\}) - V(S)$ the marginal; $w(S)$ differs from the RL allocation weights $w_{i,m}$.

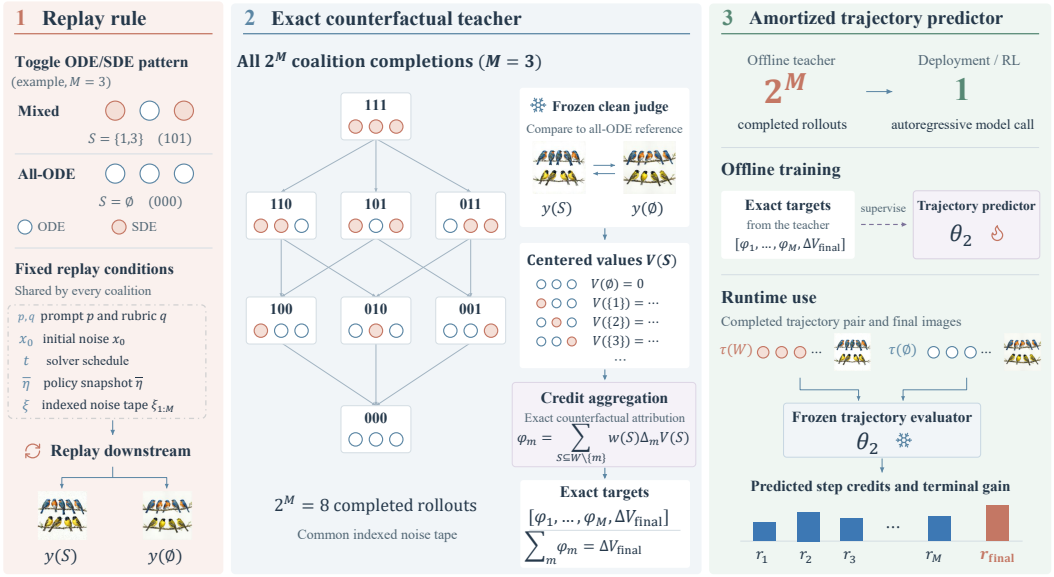


Figure 6: ODE-referenced counterfactual attribution. (1) Coalition bits select SDE (1) or ODE (0), with prompt, rubric, initial noise, solver, policy, and indexed noise tape fixed; all affected downstream states are replayed. (2) For $M = 3$, all eight completions are compared with the shared all-ODE image. Centered values yield Shapley targets satisfying $\sum_m \phi_m = \Delta V_{\text{final}}$. (3) These offline targets supervise a predictor used in one autoregressive call per completed path pair. Paired sampling and encoding remain runtime costs, and generated arrays need not satisfy the target identity.

The exact construction fixes the policy, teacher, sampler, prompt, initial latent, window, and indexed random tape within each family. After generating the all-ODE and full mixed paths, each remaining coalition reuses only its identical prefix and recomputes the first changed transition and every subsequent state. Decoded outcomes define Eq. 3, and Eq. 4 gives the unrounded targets before serialization. Outcomes are invariant to coalition enumeration order. Each stochastic transition uses its assigned tape entry even when preceding stochastic steps are omitted; reindexing draws after omission defines a different game.

For a uniformly sampled permutation π, let P_m^π contain the interventions before m. Its marginal is $d_m^\pi = V(P_m^\pi \cup \{m\}) - V(P_m^\pi)$. Across K independent permutations,

$$\widehat{\phi}_m = K^{-1} \sum_{k=1}^{K} d_m^{\pi_k}, \qquad \mathbb{E}_\pi[\widehat{\phi}_m] = \phi_m. \tag{13}$$

For each permutation, the marginals telescope from the empty to the full coalition, so $\sum_m \widehat{\phi}_m = V(W)$ when every permutation is completed and endpoint scores are shared. This does not remove the coordinate-wise Monte Carlo error. Permutation uncertainty, teacher-score variability, and variation across actor seeds are separate quantities.

A.2 Interactions that change the deployed allocation

For a centered game, define its Harsanyi dividends recursively by $d(S) = V(S) - \sum_{R \subsetneq S} d(R)$, with $d(\emptyset) = 0$. Then $V(S) = \sum_{R \subseteq S} d(R)$ and

$$\phi_i = \sum_{S \ni i} \frac{d(S)}{|S|}, \qquad \mathrm{LOO}_i = \sum_{S \ni i} d(S). \tag{14}$$

For $M = 3$, let $a_i = d(\{i\})$, $b_{ij} = d(\{i, j\})$, and $c = d(\{1, 2, 3\})$. Thus

$$\phi_i = a_i + \tfrac{1}{2} \sum_{j \neq i} b_{ij} + \tfrac{1}{3} c, \qquad \mathrm{LOO}_i = a_i + \sum_{j \neq i} b_{ij} + c. \tag{15}$$

A pure triple interaction gives constant credit vectors and hence uniform weights under either rule. More generally, if only dividends of order k are nonzero, $\mathrm{LOO} = k\boldsymbol{\phi}$. For nonzero credits the RMS-normalized weights coincide as $\varepsilon \to 0$; at finite ε, scaling by k replaces ε with ε/k. Even heterogeneous pure pairwise interactions can therefore yield nearly identical allocation.

Mixed orders can instead change rankings. In a constructed three-player game with $a = (0.10, 0, 0)$, $b_{23} = 0.15$, and other dividends zero, $\boldsymbol{\phi} = (0.10, 0.075, 0.075)$ whereas $\mathrm{LOO} = (0.10, 0.15, 0.15)$. Step 1 changes from highest to lowest credit. Adding a self-score of 0.5 gives valid coalition probabilities in $[0.5, 0.75]$. This is a mathematical example, not a measured trajectory. A common triple component cancels from pairwise credit differences but may still change weight sharpness through the RMS denominator. The mapping discrepancy

$$D_w^{\phi,\mathrm{LOO}} = M^{-1} \| w(\boldsymbol{\phi}, A) - w(\mathrm{LOO}, A) \|_1 \tag{16}$$

compares the two allocations under a common terminal advantage A. Interaction magnitude alone does not determine this discrepancy.

What the oracle measures. The leave-one-out target is $V(W) - V(W \setminus \{m\})$; the add-in target is $V(\{m\}) - V(\emptyset)$. Neither averages over all coalition contexts. The oracle experiment compares credit rules under a teacher terminal anchor. It is an optimization diagnostic, not a proof that Shapley attribution equals an unbiased action advantage.

A.3 Why endpoints do not identify process credit

Consider two abstract two-intervention games with the same endpoint scores, $V(\emptyset) = 0$ and $V(\{1, 2\}) = 0.4$. The first has $(V(\{1\}), V(\{2\})) = (0.4, 0)$ and the second $(0, 0.4)$. Equation 4 gives credits $(0.4, 0)$ and $(0, 0.4)$, respectively. Both games fit a bounded preference statistic with the same self-comparison offset. This is an algebraic illustration, not a diffusion experiment: terminal scores alone cannot determine the coalition marginals. Even identical observed paths need not specify how an unobserved coalition would evolve without a known dynamics model. The learned evaluator instead exploits regularities of labeled trajectory families. Better endpoint prediction does not guarantee better attribution, and accurate attribution on one distribution does not establish causal identification or transfer to another sampler.

A.4 Teacher scale is part of the attribution target

Subtracting the same self-score from every coalition sets $V(\emptyset) = 0$ but cancels from each marginal. It is not calibration. Probability, log-odds, and loss explain different output quantities (Chen et al., 2022). Positive affine score transformations scale Shapley credits by a positive constant; a nonlinear monotone transformation need not preserve credit signs because a player's positive and negative marginals may be rescaled differently.

For a constructed two-player example, take $(p_\emptyset, p_1, p_2, p_{12}) = (0.50, 0.55, 0.98, 0.96)$. Then $\phi_1 = \frac{1}{2}[(0.55 - 0.50) + (0.96 - 0.98)] = 0.015$. The strictly increasing transformation $f_\tau(p) = \mathrm{sigmoid}(\mathrm{logit}(p)/\tau)$ at $\tau = 2$ yields $(0.5, 0.525063, 0.875, 0.830479)$ and $\phi_1^{(2)} \simeq -0.009729$, despite preserving every outcome ranking. This is a mathematical counterexample, not observed teacher instability. If all of a player's marginals have the same sign, strictly increasing transforms preserve that sign.

A transformed game applies the score transformation to every coalition and its shared self-score before recomputing Shapley values. This differs from transforming the existing credit vector. A common fixed A separates the resulting allocation change from terminal-advantage effects. The scalar transform above is not generally equivalent to changing an LM's decoding temperature: an affirmative answer event need not form one class of an exhaustive binary softmax. Probability calibration is a separate statistical property; post-hoc calibration fits a transform to a defined prediction event using held-out labels (Guo et al., 2017).

A.5 Interpretation of the figure examples

Figure 1(1) is the algebraic example in Appendix A.3; Figures 2 and 6 describe the training and attribution procedures. Intermediate thumbnails in Figures 1 and 2 are selected for schematic presentation and do not define intervention timestamps. Figure 3 reports experimental measurements. Its bird case uses a prompt for five blue birds on an upper branch and four yellow birds below, with coalition bits ordered by intervention time.

In Figure 3A, the exact credits sum to a centered terminal gain of 0.19. The reward difference, $0.68 - 0.31 = 0.37$, is the single marginal $V(\{1,2,3\}) - V(\{1,3\})$; it differs from the averaged step-2 Shapley credit 0.21 and from the gain relative to the all-ODE baseline. The upper plot in panel B reports a separate attribution window; the lower bars reproduce the aggregate rows in Table 2. Panel C reports measured Clean-RM evaluation reward over wall-clock time for exact, learned, permuted, and uniform credit. All methods' final images are scored by the same Clean RM against the same fixed reference image. This reward and the GenEval scores in Table 3 are distinct evaluation metrics. Appendix E relates online learning cost to offline training and model refresh.

B Training and Post-Rollout Update Details

B.1 Stage I: latent corruption and preference supervision

For a clean preference pair (y^+, y^-), a generator's variational autoencoder (VAE) and native noise schedule produce

$$z^\pm = E_{\mathrm{VAE}}(y^\pm), \quad x_t^\pm = \alpha_t z^\pm + \sigma_t \epsilon^\pm, \quad Z_t^\pm = A_\psi(E_{\mathrm{G}}(x_t^\pm, t, p), t). \tag{17}$$

Here E_{G} is the trainable reward-side latent encoder, and A_ψ maps its spatial features and timestep information into VLM-compatible tokens. Stage I jointly optimizes E_{G} and A_ψ under Eq. 18, while keeping the preference backbone and native output head frozen; θ_1 denotes the adapted encoder–adapter model. For a linear flow path, $\alpha_t = 1 - t$ and $\sigma_t = t$. The formulation samples standard normal noise $\epsilon^\pm$ and draws t from a chosen training distribution $p_{\mathrm{train}}(t)$; the pair's noise coupling is specified by its input protocol. Replacing the two image inputs by Z_t^a, Z_t^b preserves the prompt and rubric. With randomized pair order and corresponding preference fraction h, the objective is

$$\mathcal{L}_{\mathrm{I}} = \mathbb{E}\left[b(t)\left\{-h \log P_{\theta_1}(Y \mid p, q, Z_t^a, Z_t^b) - (1-h)\log P_{\theta_1}(N \mid p, q, Z_t^a, Z_t^b)\right\}\right], \tag{18}$$

where N is the negative answer event and $b(t) \geq 0$ controls noise weighting. The source preference supervises perception, not rollout credit. At high noise, a preference need not be recoverable from

a particular view, so evaluation is stratified by t. All views of a source pair belong to one split. Shared pair noise cancels from the pre-encoder latent difference, $x_t^+ - x_t^- = \alpha_t(z^+ - z^-)$, whereas independent noise does not. The two noise couplings define different input distributions; cancellation does not imply exact recovery after nonlinear encoding or at pure noise.

B.2 Stage II: context, serialization, and training

The Stage II model, $\theta \equiv \theta_2$, receives completed mixed and all-ODE paths, their final images, prompt, and rubric:

$$\mathcal{C} = [p, q, Z_{\text{prefix}}, Z_{1:M}^{\text{pair}}, Z_{\text{suffix}}^{\text{pair}}, E_{\text{V}}(y(W)), E_{\text{V}}(y^{\text{ODE}})]. \tag{19}$$

Transition tokens contain pre/post-state evidence with time, step-index, and ODE/SDE role information. After path divergence, paired feature differences compare different parent states and are not local causal credits. All context is available after rollout and before array decoding. Stage II continues training the Stage I-initialized E_{G} and A_ψ together with language LoRA. Once trained, the reward encoder is frozen for actor updates and need not match the changing actor; reusing actor activations is valid only when encoder and preprocessing coincide. Otherwise, separate reward-side feature extraction contributes to evaluation cost.

The shared language-model head in Eq. 5 learns the quantized target sequence using Eq. 6. The chronological array contains M step credits followed by the final value; the quantizer Q_δ is fixed before serialization. The reward-side latent encoder, trajectory adapter, and language LoRA learn this output task; the clean teacher remains frozen. Only training labels require coalition outcomes. Inference uses the observed trajectory pair and images in one autoregressive call. Joint generation does not enforce numerical accuracy or conservation, so raw parsing, attribution error, and final prediction error are separate evaluation quantities (Appendix C). Autoregressive serialization provides a shared output interface for the step and final coordinates.

B.3 Actor objective and reference collection

Equation 10 updates only SDE transitions with nondegenerate policy densities, using stop-gradient advantages. The clipping radius ϵ_{clip} is distinct from the normalization stabilizer ε. ODE segments provide context and outcomes, and the paired reference is recomputed when the behavior snapshot changes. Reference collection and full input-token processing contribute to online cost.

For fixed normalized credits, the softmax in Eq. 8 solves an entropy-regularized allocation over steps. Mixing with uniform weights gives $1 - \lambda \le w_{i,m} \le 1 + (M-1)\lambda$ and controls redistribution strength. These are trajectory-level allocation weights, not online action advantages or signed causal rewards. Appendix D proves the identities and explains why scalar conservation does not preserve the policy gradient.

B.4 From prediction error to allocation error

Equation 11 bounds the allocation discrepancy between predicted and exact credits under a common fixed terminal advantage A_i, temperature T, stabilizer ε, and mixture strength λ. At $A_i = 0$ both allocations are uniform. The deployment-oriented metric

$$E_w = \frac{1}{N} \sum_i \|w(\mathbf{r}_i, A_i) - w(\boldsymbol{\phi}_i, A_i)\|_1 \tag{20}$$

depends on relative credit error, temperature, and mixture strength, not Step MAE alone. For fixed data and actor parameters, this also bounds perturbation of the clipped surrogate. The proof in Appendix D establishes fidelity to the chosen exact-credit update, not policy improvement. Terminal-value errors and changes to group normalization are outside this fixed-A_i comparison.

C Array Metrics and Numerical Output

For N windows of length M, step and final errors are

$$\text{MAE}_{\text{step}} = \frac{1}{NM} \sum_{i,m} |r_{i,m} - \phi_{i,m}|, \quad \text{MAE}_{\text{final}} = \frac{1}{N} \sum_i |r_{i,\text{final}} - V_i(W)|. \tag{21}$$

Within-window Spearman compares each predicted array's step ranks with the target ranks, followed by aggregation over windows. Sign accuracy compares predicted and target signs; top-absolute accuracy compares the selected coordinates with the largest absolute contributions. Parse validity and numerical fidelity evaluate distinct aspects of array output.

Two different conservation diagnostics are

$$R_{\text{self}} = \frac{1}{N}\sum_i \left|\sum_m r_{i,m} - r_{i,\text{final}}\right|, \quad R_{\text{teacher}} = \frac{1}{N}\sum_i \left|\sum_m r_{i,m} - V_i(W)\right|. \tag{22}$$

Even zero R_{self} permits an incorrect allocation and an incorrect total. Normalized attribution error, $\sum_{i,m}|r_{i,m} - \phi_{i,m}| / \sum_{i,m}|\phi_{i,m}|$, expresses prediction error relative to the total target magnitude; it is undefined when that denominator is zero. The oracle-total uniform predictor $V_i(W)/M$ uses privileged endpoint information. Constant credit arrays give uniform weights even when their attribution MAE differs. The metric E_w compares predicted and exact-credit allocations under the same fixed A_i, which depends on the entire group through Eq. 7. Dividing $\|\Delta w_i\|_1$ by M_i normalizes discrepancies across window lengths.

Serialization and failures. The answer contract is a chronological list of $M+1$ signed numbers with the final difference last. Raw event probabilities imply $V(W), \phi_m \in [-1, 1]$. Independent round-to-nearest quantization with step δ satisfies

$$\left|\sum_m Q_\delta(\phi_m) - Q_\delta(V(W))\right| \le (M+1)\delta/2, \tag{23}$$

because each coordinate's rounding error is at most $\delta/2$. This is a label-rounding bound, not a bound on generated predictions. Unconstrained parse validity is the fraction of generated arrays that satisfy the numerical format before any grammar restriction.

C.1 Latent perception and initialization diagnostics

Table 4 evaluates pairwise preference accuracy on ImageReward, HPDv2/HPDv3, and GenAI-Bench (Xu et al., 2023; Wu et al., 2023; Ma et al., 2025; TIGER-AI-Lab, 2024). Stage I leads on ImageReward and HPDv2; the transfer control leads on HPDv3 and SURE-LRM on GenAI-Bench. Its 0.375 percentage-point macro advantage over the transfer control is not a uniform dataset-level gain.

Table 4: Stage I pairwise preference accuracy (%). Macro averages the four dataset columns. Same-clean StitchVM-style denotes a transfer control built from the same clean Qwen checkpoint. Colors follow Table 1.

Method	ImageReward	HPDv2	HPDv3	GenAI	Macro
DiNa-LRM	60.6	81.8	74.5	68.9	71.450
SURE-LRM	62.0	83.7	74.9	**73.5**	73.525
DRM	65.5	83.2	74.8	70.4	73.475
Same-clean StitchVM-style	66.7	84.5	**75.9**	72.1	74.800
Ours Stage I	**67.1**	**84.8**	75.8	73.0	**75.175**

Table 5: Stage I accuracy (%) averaged over the four datasets at each reported noise level.

$t = 0.10$	$t = 0.25$	$t = 0.40$	$t = 0.50$	$t = 0.75$	$t = 0.90$
76.200	75.800	75.175	73.800	66.800	55.500

Table 6: All evaluated Stage II label budgets. Budgets count windows, not coalition images. Dark/light pink marks lower/higher MAE within each row at the same budget.

Training windows	Stage II only MAE	Stage I + II MAE
1,000	0.050	**0.036**
2,000	0.043	**0.029**
5,000	0.034	**0.024**
10,000	0.028	**0.019**
20,000	0.024	**0.017**

Table 7: Output validity and policy transfer.

Diagnostic	Value
Unconstrained array parse validity (%)	99.6
Clean verdict accuracy change after Stage II (pp)	-0.3
OOD-policy Step MAE, without refresh	0.026
OOD-policy Step MAE, after refresh	0.021

D Allocation Identities and Interpretation

The softmax probabilities in Eq. 8 sum to one, hence $\sum_m w_{i,m} = M(1-\lambda) + \lambda M = M$. Multiplication by the same terminal advantage proves Eq. 9. Since each softmax probability lies in $[0,1]$, $1-\lambda \le w_m \le 1+\lambda(M-1)$. For fixed scores u_m, define $H(p) = -\sum_m p_m \log p_m$. The unique solution on the probability simplex to

$$p^* = \arg \max_{p_m \ge 0,\, \sum_m p_m = 1} \left\{ \sum_m p_m u_m + T H(p) \right\} \tag{24}$$

is $p^*_m \propto \exp(u_m/T)$: the stationary condition is $u_m - T(1+\log p_m) + \nu = 0$, and strict concavity gives uniqueness. This justifies the softmax as entropy-regularized redistribution for fixed input credits, not as an optimal policy-gradient estimator.

Credit ordering and sign. For $A_{\text{final}} \neq 0$, $\lambda > 0$, and finite $T > 0$, strict monotonicity of the exponential gives

$$\operatorname{sign}(A_m - A_n) = \operatorname{sign}(r_m - r_n). \tag{25}$$

Indeed, the logit difference has sign $\operatorname{sign}(A_{\text{final}})\operatorname{sign}(r_m - r_n)$; multiplying the weight difference by A_{final} cancels the first sign. On a below-average trajectory, smaller credits receive more negative advantages; this remains true if all credits happen to be positive. The ordering is strict for distinct credits, but all step advantages retain the terminal sign. At zero terminal advantage, that trajectory's advantage-weighted surrogate terms vanish; reference regularization may still update the actor. At $\lambda = 0$, all weights equal one; equal step credits also give uniform weights. Nonnegative weights preserve the sign of the terminal group advantage at every transition. A trajectory can improve over ODE but improve less than its group peers, so $r_{i,\text{final}} > 0$ need not imply $A_{i,\text{final}} > 0$.

Let $g_m = \nabla_\eta \log \pi_\eta(x_{m+1} \mid x_m, p)|_{\bar\eta}$. Ignoring clipping and the common averaging factor, the difference from uniform allocation is

$$\mathbb{E}\left[A_{\text{final}} \sum_m (w_m - 1) g_m \right]. \tag{26}$$

Although the coefficients sum to zero, their products with unequal score gradients need not. For instance, score terms $(1, 3)$ and weights $(1.5, 0.5)$ give $3A_{\text{final}}$ instead of $4A_{\text{final}}$ under uniform weights. This algebraic example establishes why scalar conservation is not gradient equivalence. Full-trajectory-dependent weights are also not action-independent baselines. Clipping, group normalization, and repeated updates introduce further differences from an exact return gradient.

D.1 Proof of the allocation-error bound

Write $a = \|x\|_2/\sqrt{M}$, $b = \|y\|_2/\sqrt{M}$, $e = \|x-y\|_2/\sqrt{M}$, $d_a = a+\varepsilon$, and $d_b = b+\varepsilon$. Expanding the squared norms gives

$$\frac{1}{M}\left\| \frac{x}{d_a} - \frac{y}{d_b} \right\|_2^2 = \frac{e^2}{d_a d_b} - \frac{(a-b)^2[ab + \varepsilon(a+b)]}{d_a^2 d_b^2} \le \frac{e^2}{d_a d_b}. \tag{27}$$

For $p = \operatorname{softmax}(z)$, its Jacobian $J = \operatorname{diag}(p) - pp^\top$ satisfies

$$\|Jv\|_1 = \mathbb{E}_p|v - \mathbb{E}_p v| \le \sqrt{\operatorname{Var}_p(v)} \le \tfrac{1}{2}(\max_m v_m - \min_m v_m) \le \|v\|_2/\sqrt{2}. \tag{28}$$

For the middle bound, putting $\ell = \min v$ and $h = \max v$ gives $\operatorname{Var}_p(v) \le (\mathbb{E}_p v - \ell)(h - \mathbb{E}_p v) \le (h-\ell)^2/4$. The final bound follows from $(v_j - v_k)^2 \le 2(v_j^2 + v_k^2)$. Integrating the Jacobian along the line segment between two logit vectors proves a global ℓ_2-to-ℓ_1 Lipschitz constant of $1/\sqrt{2}$. Apply this to the normalized vectors with a common sign and temperature, then multiply by λM. The distance between probability vectors is at most two, proving Eq. 11. At $A = 0$ both logit vectors vanish.

To relate this to a fixed-batch update, write $h_m(\eta, A) = \min\{\rho_m(\eta)A, \operatorname{clip}(\rho_m(\eta), 1-\epsilon_{\text{clip}}, 1+\epsilon_{\text{clip}})A\}$ and $F_x = M^{-1}\sum_m w_m(x,A)h_m(\eta,A)$. Nonnegative stop-gradient weights factor out of the clipped minimum, so the surrogate in Eq. 10 has this form. Then

$$|F_x - F_y| \le \frac{\max_m |h_m|}{M} \|w(x,A) - w(y,A)\|_1. \tag{29}$$

At differentiable parameters the analogous gradient bound replaces $\max_m |h_m|$ by $\max_m \|\nabla_\eta h_m\|$ in any fixed norm. These are perturbation bounds with fixed data, actor parameters, and terminal advantage, not guarantees on subsequent returns.

D.2 Output permutation as an alignment control

Applying a permutation matrix P only to the M predicted step credits leaves the trajectory inputs, final output, and A fixed. RMS invariance and softmax equivariance give

$$w(P\mathbf{r}, A) = Pw(\mathbf{r}, A). \tag{30}$$

Thus the exact credit and weight multisets, their sums and RMS values, and the entropy of normalized weights w/M are preserved; only assignment to transitions changes. For a uniformly random permutation over all $M!$ possibilities, including the identity, $\mathbb{E}_P[w(P\mathbf{r}, A)_m] = M^{-1}\sum_k w_k = 1$. Since $w_m \geq 0$, it factors out of the clipped minimum for either sign of A. Consequently, for a fixed rollout, prediction, A, and actor parameter η,

$$\mathbb{E}_P[F_{P\mathbf{r}}(\eta)] = F_{\text{uniform}}(\eta). \tag{31}$$

The finite permutation average also preserves the gradient at differentiable parameters (or under a common branch-derivative rule at clipping boundaries). A common reference regularizer leaves this identity unchanged. It does not equate realized gradients, their variance, nonlinear optimizer steps, or training trajectories. If the same permutation is reused across updates, later actor parameters depend on it; the fixed-parameter expectation cannot then be applied naively. Excluding the identity or using only derangements generally invalidates the coordinate mean-one claim.

E Amortization Cost

Let $B = U_{\text{uniform}} c_{\text{uniform}}$ be the update-dependent cost of reaching a chosen evaluation target, excluding common fixed overheads from both budgets. Here U counts updates and c is cost per update. Define $a = C_{\text{extra,offline}}/B$, $q = U_{\text{ours}}/U_{\text{uniform}}$, and $b = c_{\text{ours}}/c_{\text{uniform}}$. Then

$$\frac{C_{\text{ours}}}{B} = a + qb, \qquad C_{\text{ours}} < B \iff a + qb < 1. \tag{32}$$

This break-even relation separates offline acquisition and fitting from online collection and updates. Offline cost includes Stage I, coalition generation, decoding and judging, and Stage II. Online cost includes the ODE reference, reward-side feature extraction when actor activations are incompatible, trajectory input and autoregressive output processing, actor collection, and policy updates. A refresh that acquires new-policy coalition labels and refits the evaluator contributes both acquisition and fitting costs. The autoregressive interface is one model call with multiple decoding steps. When the clean evaluator is a shared asset, $C_{\text{extra,offline}}$ measures marginal adaptation cost; including its initial training instead gives the full-pipeline accounting.

F Additional Qualitative Results

Figure 7 extends the main-text comparison in Figure 4 using the same five method columns. The first three rows test text rendering in a smartwatch display, burger advertisement, and classroom periodic-table poster. The remaining rows examine a dog assembled from fruit, three giraffes, a pink handbag with black scissors, and a keyboard combining water and light.

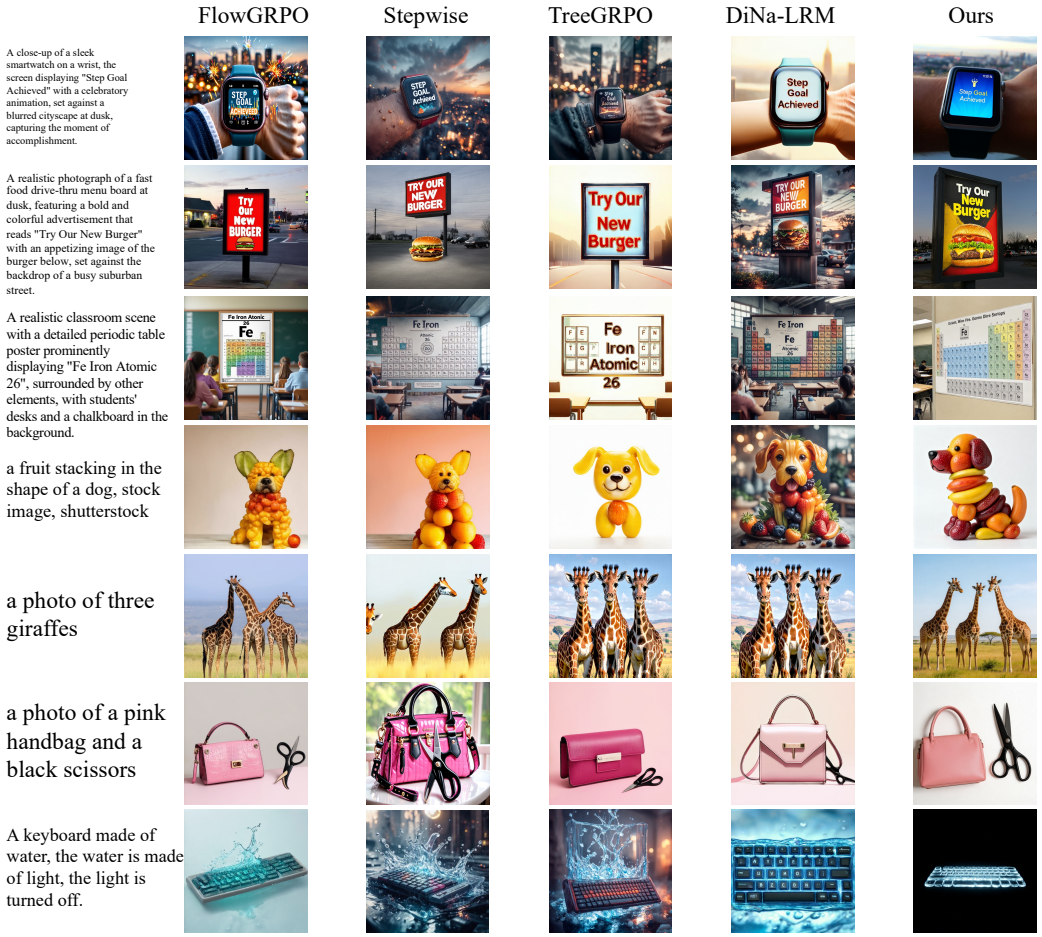


Figure 7: Additional qualitative generation results. Columns show FlowGRPO, Stepwise, TreeGRPO, DiNa-LRM, and Ours. Prompts at left specify text content, object composition and count, color attributes, or material appearance. The seven examples complement the four main-text cases in Figure 4.

G Extended Related Work

G.1 Preference evaluation from images and noisy latents

ImageReward, HPSv2, and HPSv3 develop learned visual preference scores and benchmarks (Xu et al., 2023; Wu et al., 2023; Ma et al., 2025). RewardDance uses instruction-conditioned generative judgments (Wu et al., 2025); our evaluator adopts this interface with a Qwen3.5-9B backbone (Qwen Team, 2026). The clean judge supplies the common preference statistic for our coalition game, while the learned trajectory evaluator receives latent evidence. LRM, DiNa-LRM, DRM, and SURE-LRM extend preference evaluation to noisy states (Zhang et al., 2025; Liu et al., 2026; Zhang et al., 2026; Li et al., 2026b). These evaluators can support process optimization: DRM normalizes scores of stochastic successors sharing one parent to obtain local transition advantages, and Latent Reward Registers use reward gradients for stepwise distillation (Guan et al., 2026). Thus latent scoring and transition feedback are compatible. SLRM augments a diffusion-based preference discriminator with the denoising score, while TAPO evaluates several timesteps to handle inconsistent preference rankings across sampled trajectories (Xian et al., 2026). Its focus on reliable trajectory comparison is complementary to allocating a specified terminal comparison across replayed interventions. StitchVM learns an interface between diffusion features and a clean evaluator, regressing clean scores from noisy states (Go et al., 2026). It is a close precedent for transferring preference perception into the denoising process. Our Stage I follows that broad motivation; Stage II changes the supervised object to a coalition allocation of the completed trajectory's preference change. The same-checkpoint StitchVM-style variants in our experiments are adapted controls, distinguished from the original method's training recipe and architecture.

G.2 Temporal supervision and diffusion policy updates

DDPO and Flow-GRPO optimize generation with terminal rewards (Black et al., 2024; Liu et al., 2025). MixGRPO places a stochastic window within deterministic sampling (Li et al., 2025). DenseGRPO and Stepwise-Flow-GRPO obtain temporal feedback from intermediate clean completions (Deng et al., 2026; Savani et al., 2026); the latter forms successive reward improvements. TempFlow evaluates exploratory transitions through deterministic suffixes (He et al., 2026). TreeGRPO and BranchGRPO instead organize sampled continuations into branching structures (Ding & Ye, 2026; Li et al., 2026c). Our counterfactual construction also uses completed outcomes, but averages a transition's marginal over subsets of retained interventions, with an indexed noise tape shared across replays. This target differs from adjacent reward gains or backups over sampled descendants. STAR weights terminal advantages using timestep-specific attention (Shen et al., 2026). Our weights use learned intervention credits and explicitly preserve each trajectory's mean scalar advantage; STAR's spatial coefficients do not impose this same mean-one constraint. CoCA redistributes terminal rewards using an outcome-aligned latent-progress proxy: changes in similarity between intermediate latents and the realized terminal latent (Liao et al., 2026). This preprint provides trajectory-dependent credit allocation without training an auxiliary per-step preference model. Our target instead measures preference changes under explicit transition replacements and downstream replay, at the cost of offline coalition labeling. PCPO addresses a different source of uneven credit: timestep scaling induced by the sampler. It reformulates the policy objective for numerical stability and reweights timesteps to enforce proportional feedback (Lee & Ye, 2026). This correction concerns how the sampler scales updates; our weights depend on the realized interventions' contributions to the chosen terminal preference statistic. Pave-GRPO reuses coarse rollout endpoints for finer velocity supervision (Ling et al., 2026), avoiding additional generation and reward calls while requiring extra update-time computation. This distinction matters when comparing its supervision cost with offline coalition labeling. Path-space analyses organize diffusion policy-gradient estimators and samplers (Xu et al., 2026). Our future-dependent reweighting is a specified update rule; scalar-budget conservation alone does not establish gradient equivalence or policy invariance.

G.3 Retrospective and counterfactual credit assignment

RUDDER predicts sequence returns and redistributes them over transitions (Arjona-Medina et al., 2019). Model-free counterfactual credit assignment uses future-conditioned information under action-independence conditions (Mesnard et al., 2021), while COCOA estimates how actions change the likelihood of rewarding outcomes (Meulemans et al., 2023). These works establish that retrospective

evidence can inform temporal credit, with the intervention definition and policy use determining its interpretation. Shapley values average marginal effects across coalition contexts (Shapley, 1953). SCAR applies this principle to preference-model rewards over text units, replacing omitted units with spaces (Cao et al., 2025). Counterfactual Shapley Credit Assignment defines action interventions relative to a baseline policy in a structural causal model and reuses coalition evaluations across action credits (Li et al., 2026a). Our diffusion game specifies a different intervention: switching an indexed transition between coupled SDE and ODE maps, then recomputing all affected downstream states. Its credits allocate a fixed teacher statistic relative to the shared all-ODE image. They characterize this sampler-specific game rather than a universal contribution of injected noise or an online action advantage.

G.4 Learning to amortize coalition attribution

FastSHAP trains an explainer from coalition values without requiring precomputed Shapley coordinates (Jethani et al., 2022). Its weighted coalition-regression formulation admits general value functions, including a rollout-defined oracle. Learned amortization therefore has a direct precedent beyond static feature masking. Our Stage II instead fits quantized, serialized Shapley targets using token negative log likelihood. Its inputs are completed mixed and all-ODE histories, both final images, and the prompt and rubric; training labels incorporate additional coalition outcomes. This separates the expensive target construction from deployment, where one autoregressive call predicts the array from the observed pair. One call still includes multiple decoding steps and does not remove paired collection or feature-extraction costs. The relevant design choices are the diffusion replay target, latent initialization, trajectory representation, and learning objective.

H Limitations

Our step-level attribution method primarily strengthens reinforcement learning guided by image reward models. Its effectiveness remains constrained by the underlying reward model's robustness and alignment with human preferences. Developing larger-scale, robust image reward models with stronger human-preference alignment is therefore an important direction for future work.