TEMPORAL CREDIT ASSIGNMENT UNDER IMPERFECT REWARDS IN TEXT-TO-IMAGE RL

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent text-to-image reinforcement learning methods increasingly use temporally structured credit to assign different learning signals across denoising steps. Because these signals are constructed from learned reward models that are imperfect proxies for human preference, finer temporal credit can also change how reward-model error enters optimization. We study this interaction through a controlled stress test that perturbs reward-model outputs before credit construction, using reward-specific corruption scales informed by human preference calibration. Across three reward models, we compare terminal, locally differenced, and trajectory-structured credit; a descendant-backed construction serves as a complementary control. We find that robustness is not determined by temporal specificity alone. A construction that relies directly on local reward differences consistently degrades most under corruption, while alternative forms of temporally structured credit preserve substantially more of their clean performance. Controls that match clean optimization strength and vary the temporal covariance of reward errors further support this interpretation. For HPSv3, human judgments reveal a complementary picture: the calibrated perturbation captures the marginal scale and margin dependence of reward-model–human disagreement, but observed disagreements are more concentrated and temporally persistent than the iid probe predicts. Moreover, under ordinary HPSv3 training without injected corruption, the construction that is most sensitive in the controlled study also exhibits substantial divergence between optimized reward and human preference. Together, these results point away from temporal specificity alone and toward the construction of temporal credit, with adjacent reward differencing emerging as a particularly sensitive case.

1 INTRODUCTION

Reinforcement learning (RL) for text-to-image (T2I) generation increasingly goes beyond a single terminal reward. Recent methods redistribute terminal feedback across timesteps, construct local reward increments, introduce trajectory-dependent corrections, or derive step-specific credit from branching rollouts (Lee & Ye, 2025; Deng et al., 2026; Tong et al., 2026; Ding & Ye, 2026; He et al., 2026; Lyu et al., 2025). These approaches seek a more informative learning signal from the generation trajectory, but that signal is typically built from a learned reward model (RM), whose score remains an imperfect proxy for human preference (Kirstain et al., 2023; Wu et al., 2023; Ma et al., 2025; Xu et al., 2023). This raises a basic question that clean target reward alone cannot answer: *how does reward-model error change when reward evaluations are transformed into temporal credit for individual policy updates?*

A natural hypothesis is that temporally specific credit is inherently more fragile because it relies on finer-grained reward information. We argue that this framing is too coarse. Temporal specificity and credit construction are distinct design choices. DenseGRPO constructs local increments from neighboring reward evaluations, TP-GRPO combines local and trajectory-level comparisons with trajectory-dependent corrections, while TreeGRPO obtains step-specific credit by backing up rewards from completed terminal descendants. These mechanisms can expose the policy update to very different effective uncertainty even when they receive reward perturbations of the same nominal scale. The relevant object is therefore the complete transformation from evaluated outcomes to the learning signal, of which the terminal-versus-temporal choice is only one part.

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.

We study this transformation through controlled interventions at the reward-model output. Before each method constructs its credit and normalized advantages, we perturb the scalar RM evaluations and then apply the method's original learning rule. Reward-specific corruption scales are informed by human-preference calibration, while validation always uses the original unperturbed RM. Our primary sweep compares Flow-GRPO, PCPO, DenseGRPO, and TP-GRPO across PickScore, HPSv2, and HPSv3, and measures corruption both in final target reward and after method-specific credit construction and normalization. TreeGRPO, matched-clean operating points, and temporally correlated perturbations provide targeted tests of competing explanations for the observed sensitivity.

The experiments consistently separate temporal specificity from the mechanism used to construct temporal credit. DenseGRPO is the strongest clean optimizer of the target reward across the three reward models, but is also the most sensitive to corruption. TP-GRPO is substantially more robust despite using intermediate-origin, temporally structured credit, and TreeGRPO retains step-specific credit without reproducing DenseGRPO's larger degradation. Matching clean performance preserves the gap, while increasing correlation between neighboring reward errors reduces DenseGRPO's degradation in the direction predicted by local differencing. These results point away from temporal specificity alone and toward the construction of temporal credit, in particular adjacent reward differencing, as an important source of sensitivity.

We complement the controlled intervention with two human studies. Repeated judgments show that the calibrated iid probe closely captures the marginal scale and margin dependence of HPSv3 RM–human disagreement, while underestimating its concentration and temporal persistence. Separately, under ordinary HPSv3 training with no injected corruption, Flow-GRPO and TP-GRPO are strongly preferred by human annotators over DenseGRPO despite DenseGRPO achieving the highest optimized reward. Thus, the controlled sensitivity exposed by our protocol is associated with a concrete reward–preference divergence under standard training.

Our contributions are: (i) a controlled framework for tracing reward uncertainty from RM outputs through method-specific credit construction to normalized learning signals; (ii) evidence across multiple reward models and credit constructions that temporal specificity alone does not determine robustness, with targeted controls pointing to local reward differencing as a particularly vulnerable construction; and (iii) human evidence showing that the probe matches the marginal scale of HPSv3 RM–human disagreement and linking reward-noise sensitivity to reward–preference divergence. The design implication is to keep useful temporal structure while treating the transformation and aggregation of uncertain reward evidence as a first-class component of the RL algorithm.

2 Related Work

Temporal credit in text-to-image RL. Early diffusion and flow RL methods such as DDPO and Flow-GRPO optimize terminal rewards that are shared across, or assigned to, the generation trajectory (Black et al., 2023; Liu et al., 2025). Recent work seeks finer temporal credit in several ways. PCPO redistributes terminal feedback across timesteps, DenseGRPO constructs local increments from intermediate-origin reward evaluations, and TP-GRPO combines local and trajectory-level comparisons with trajectory-dependent corrections (Lee & Ye, 2025; Deng et al., 2026; Tong et al., 2026). TreeGRPO instead assigns edge-specific credit by backing up rewards from completed terminal descendants (Ding & Ye, 2026). Related approaches explore timestep weighting, branching trajectories, and stepwise credit (He et al., 2026; Lyu et al., 2025; Savani et al., 2026). These methods therefore differ in how temporally specific their credit is, and also in which outcomes they evaluate and how they transform those rewards into the learning signal. We study the second kind of difference under controlled reward uncertainty.

Imperfect rewards and reward over-optimization. Learned preference models such as PickScore, HPSv2, HPSv3, and ImageReward provide scalable objectives for text-to-image alignment, but their scores remain imperfect proxies for human preference (Kirstain et al., 2023; Wu et al., 2023; Ma et al., 2025; Xu et al., 2023). Prior work has studied calibration of preference models and the broader problem of reward over-optimization, in which improvement on a learned proxy can diverge from the intended objective (Lee et al., 2025; Gao et al., 2023; Skalse et al., 2022; Hong et al., 2026). Related work on intermediate supervision also reports difficulties in evaluating and optimizing step-level signals in diffusion models (Xue et al., 2026; Zhang et al., 2025). Offline pairwise agreement,

however, does not determine how reward error propagates once the same scores are differenced, reused, normalized, and repeatedly optimized against. We focus on this transformation from evaluator uncertainty to the effective learning signal, and pair controlled perturbations with direct human comparisons.

3 METHOD

3.1 PROBLEM SETUP AND NOTATION

Let π_θ denote a text-to-image diffusion or flow model and $c \sim \mathcal{D}$ a text prompt. For one prompt group, let

$$\tau_i = (z_{i,0}, \ldots, z_{i,T}), \qquad i = 1, \ldots, G,$$

denote sampled generation trajectories in generation order.

A learned reward model $R_r(x, c)$ assigns scores to the outcomes used by a credit-assignment method. For method a, let

$$\mathcal{E}_a(\tau_i) = (x_{a,i,e})_{e \in I_a}$$

denote its evaluated outcomes and

$$\mathbf{q}_{a,i} = (q_{a,i,e})_{e \in I_a}, \qquad q_{a,i,e} = R_r(x_{a,i,e}, c),$$

the corresponding reward evidence. We collect the full prompt-group evidence as

$$\mathbf{Q}_a = (\mathbf{q}_{a,1}, \ldots, \mathbf{q}_{a,G}).$$

We write the method-specific credit and normalization maps as

$$\mathbf{S}_a = C_a(\mathbf{Q}_a), \qquad \mathbf{A}_a = N_a(\mathbf{S}_a), \qquad F_a = N_a \circ C_a.$$

Our object of study is the sensitivity of F_a to perturbations of the reward evidence: methods can respond differently to perturbations of the same nominal RM-output scale because they evaluate, combine, and normalize rewards differently.

3.2 CREDIT CONSTRUCTION ACROSS METHODS

The compared methods differ both in the reward evidence they evaluate and in how that evidence is transformed into credit. Our primary comparison includes Flow-GRPO, PCPO, DenseGRPO, and TP-GRPO; TreeGRPO provides a complementary control with step-specific credit derived from terminal descendants.

Terminal-reward credit. Flow-GRPO evaluates only the final sampled image. Let q_i^{term} denote its terminal reward and

$$A_i^{\mathrm{term}} = N_{\mathrm{group}}\big(q_i^{\mathrm{term}}\big)$$

the corresponding prompt-group-normalized advantage. Flow-GRPO applies this terminal advantage across the sampled trajectory, so all transitions of a trajectory inherit reward evidence from the same final outcome.

PCPO uses the same terminal reward evidence and does not introduce additional RM evaluations associated with intermediate states. Instead, it changes how the terminal advantage contributes across denoising steps through a proportionate timestep reweighting of the policy objective. Thus, Flow-GRPO and PCPO differ in temporal weighting while sharing the same underlying terminal reward evidence.

Local intermediate-origin credit. DenseGRPO evaluates outcomes obtained from retained intermediate trajectory states using deterministic ODE continuations. If $q_{i,t}$ denotes the reward associated with consecutive evaluated states, its basic process reward is the adjacent gain

$$d_{i,t} = q_{i,t+1} - q_{i,t}.$$

These gains replace the shared terminal reward when constructing group-relative advantages at each timestep. Consequently, every local credit signal depends jointly on two nearby RM evaluations. The exact continuation procedure and evaluated outcomes used in our implementation are given in Appendix A.

Trajectory-structured intermediate-origin credit. TP-GRPO also evaluates intermediate-origin outcomes, but does not use their adjacent differences as the complete credit rule. Let $q_{i,0}$ denote the pure-ODE reference reward, $q_{i,1}, \ldots, q_{i,K}$ the ordered completion rewards, with $q_{i,K}$ equal to the final sampled outcome, and define

$$d_{i,t} = q_{i,t} - q_{i,t-1}, \qquad \Delta_i = q_{i,K} - q_{i,0}.$$

The local increment $d_{i,t}$ is the default process signal. TP-GRPO additionally identifies trajectory-consistent turning points from the sign structure of the local increments and the global change Δ_i. For a selected turning point, the local increment is replaced by a trajectory-level bonus,

$$B_{i,t} = \eta J_{i,t}(q_{i,K} - q_{i,t-1}), \qquad R^{\mathrm{TP}}_{i,t} = \begin{cases} B_{i,t}, & B_{i,t} \neq 0, \\ d_{i,t}, & B_{i,t} = 0. \end{cases}$$

where $J_{i,t}$ denotes the method's turning-point and trajectory-consistency selection rule and $\eta = 1$ in our configuration. The resulting process signals are normalized independently at each timestep within the prompt group. Full indicator and balancing rules are given in Appendix C.

DenseGRPO and TP-GRPO therefore use closely related sources of intermediate-origin reward evidence, but transform them differently: DenseGRPO relies directly on local adjacent gains, whereas TP-GRPO can replace those gains with a longer-horizon trajectory signal.

Step-specific credit from terminal descendants. TreeGRPO obtains temporal credit through a different source of reward evidence. It scores only completed terminal leaves of a branching rollout, normalizes their rewards within the prompt tree, and recursively backs them up to individual edges. For an edge e with descendant-leaf set $\mathcal{L}(e)$, its credit has the form

$$S^{\mathrm{Tree}}_e = \frac{1}{|\mathcal{L}(e)|} \sum_{\ell \in \mathcal{L}(e)} \bar{q}_\ell,$$

where $\bar{q}_\ell$ is the normalized reward of terminal leaf ℓ. TreeGRPO therefore assigns distinct credit to individual transitions without forming adjacent differences between reward evaluations associated with neighboring trajectory states.

These constructions separate three properties that are otherwise easy to conflate: where reward evidence is obtained, how temporally specific the resulting credit is, and how the evidence is transformed before entering the policy update.

3.3 Controlled Reward Corruption

We perturb reward-model outputs before method-specific credit construction. For every evaluated outcome,

$$\widetilde{q}_{a,i,e} = q_{a,i,e} + \alpha \sigma_{0,r} \epsilon_{a,i,e}, \qquad \epsilon_{a,i,e} \sim \mathcal{N}(0,1), \tag{1}$$

where α controls corruption magnitude and $\sigma_{0,r}$ is the reward-specific reference scale in Section 3.4. Distinct evaluated outcomes receive independent errors in the primary intervention; reused evaluations reuse the same corrupted value. All downstream credit construction and normalization are recomputed from $\widetilde{\mathbf{Q}}_a$. Validation always uses the original unperturbed RM.

Local differencing. For DenseGRPO,

$$d_t = q_{t+1} - q_t$$

becomes

$$\widetilde{d}_t = d_t + \sigma(\epsilon_{t+1} - \epsilon_t), \qquad \sigma = \alpha \sigma_{0,r}. \tag{2}$$

If adjacent errors have unit marginal variance and correlation ρ,

$$\mathrm{Var}(\widetilde{d}_t - d_t) = 2\sigma^2(1-\rho). \tag{3}$$

For a fixed $d_t > 0$ under jointly Gaussian errors,

$$\Pr(\widetilde{d}_t < 0 \mid d_t) = \Phi\left(-\frac{d_t}{\sigma\sqrt{2(1-\rho)}} \right). \tag{4}$$

Thus, the reliability of local credit depends on the reward margin relative to uncertainty in a difference of evaluations.

The full mapping F_a can further reshape this perturbation through trajectory-level corrections, descendant aggregation, and group normalization. We therefore measure corruption both before and after normalization. This is a local mechanism rather than a complete model of downstream optimization; Appendix C analyzes the induced gradient-noise structure.

Temporally correlated corruption. As a targeted HPSv2 DenseGRPO control, we replace iid errors by stationary AR(1) perturbations of fixed marginal variance and vary adjacent correlation ρ. Equation 3 predicts that positive correlation should reduce corruption from adjacent differencing. Implementation details are in Appendix C.

3.4 Reward-Specific Corruption Scale

Because PickScore, HPSv2, and HPSv3 operate on different score scales, a common absolute perturbation magnitude is not meaningful. For each reward model r, we therefore score 3,300 adjacent-rank HPDv2 preference pairs and fit a one-parameter Bradley–Terry model (Bradley & Terry, 1952),

$$p(y_{ij} = 1 \mid \Delta s) = \mathrm{sigmoid}\left(\frac{\Delta s}{\tau_r}\right), \qquad \tau_r > 0. \tag{5}$$

The fitted preference scale is converted to the fixed train-time references

$$\sigma_{0,\mathrm{PickScore}} \simeq 0.0634, \qquad \sigma_{0,\mathrm{HPSv2}} \simeq 0.0417, \qquad \sigma_{0,\mathrm{HPSv3}} \simeq 2.6369.$$

The conversion is reward-model specific and is detailed in Appendix B.

We evaluate

$$\alpha \in \{0, 0.5, 1, 2\}.$$

The Bradley–Terry temperature characterizes preference margins, whereas $\sigma_{0,r}$ defines the intervention used in training; we do not interpret the calibration model as a generative model of RM error. Appendix B reports fitting uncertainty, score-standardization comparisons, and scale conversion.

3.5 Experimental Protocol and Measurements

The primary sweep evaluates four credit constructions under three reward models at four corruption levels and three seeds, for

$$3 \times 4 \times 4 \times 3 = 144$$

training runs. All runs use SD3.5-Medium (Esser et al., 2024) with LoRA adapters (Hu et al., 2022) at 512×512 resolution. Shared and method-specific hyperparameters are reported in Appendix A.

Let $M_{a,r,s}(\alpha)$ denote the best validation reward measured by the original RM after training at corruption level α. We report paired relative degradation

$$D_{a,r,s}(\alpha) = 100\frac{M_{a,r,s}(0) - M_{a,r,s}(\alpha)}{M_{a,r,s}(0)}. \tag{6}$$

Drops are interpreted only within each reward model and fixed score convention, and summaries are computed from paired per-seed drops.

We additionally measure (i) process-signal correlation before normalization, (ii) within-prompt rank flips, and (iii) correlation between final normalized advantages computed from clean and corrupted evidence. These diagnostics trace uncertainty through C_a and F_a; full definitions are in Appendix C.

TreeGRPO, matched-clean-performance, and temporally correlated corruption are targeted HPSv2 controls rather than part of the primary sweep. The two human studies separately test whether the calibrated probe reproduces observed RM–human disagreement and whether the robustness distinction is associated with reward–preference divergence under ordinary training.

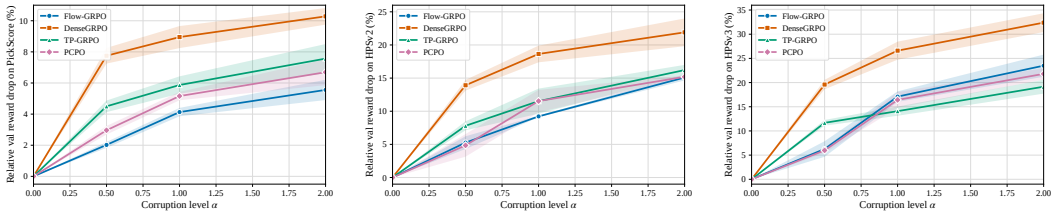
Table 1: Main three-seed comparison at the calibrated reference level $\alpha = 1$. Both reward columns are evaluated with the original, unperturbed reward model; α denotes the corruption used during training. Bold values mark the highest clean target reward and the smallest relative drop within each reward model.

RM	Method	$\alpha = 0$	$\alpha = 1$	Drop (%)
PickScore	Flow-GRPO	23.21	22.25	**4.13**
	PCPO	23.45	22.24	5.16
	DenseGRPO	**24.32**	22.15	8.95
	TP-GRPO	24.02	22.61	5.87
HPSv2	Flow-GRPO	0.371	0.337	**9.22**
	PCPO	0.377	0.334	11.53
	DenseGRPO	**0.405**	0.329	18.64
	TP-GRPO	0.398	0.352	11.52
HPSv3	Flow-GRPO	12.73	10.56	17.10
	PCPO	12.82	10.72	16.42
	DenseGRPO	**14.95**	10.97	26.60
	TP-GRPO	14.33	12.31	**14.09**

4 Results

4.1 Sensitivity Across Credit Constructions

Figure 1 shows the main robustness result. Across all three reward models, DenseGRPO achieves the highest clean target reward but degrades most strongly as corruption increases. TP-GRPO shows a markedly different profile despite also using intermediate-origin evaluations and temporally structured credit.



(a) PickScore (b) HPSv2 (c) HPSv3

Figure 1: Relative target-reward degradation as training-reward corruption increases. Each curve reports the mean paired relative drop across three training seeds for corruption level α, relative to the matched clean run for the same method and reward model; shaded regions denote one standard deviation across seeds. DenseGRPO shows the largest degradation across all three reward models, whereas TP-GRPO preserves substantially more target reward despite also using temporally structured credit. Percentages are interpreted within each reward model and are not directly comparable across reward scales.

At the calibrated reference level $\alpha = 1$ (Table 1), DenseGRPO loses 8.95%, 18.64%, and 26.60% of its clean target reward on PickScore, HPSv2, and HPSv3, respectively, compared with 5.87%, 11.52%, and 14.09% for TP-GRPO. Thus, DenseGRPO loses roughly 1.5–$1.9\times$ as much relative target reward as TP-GRPO at the same reward-specific corruption level.

Flow-GRPO has the smallest degradation on PickScore and HPSv2, whereas TP-GRPO is the most robust on HPSv3 despite constructing trajectory-dependent temporal credit. The ordering therefore cannot be reduced to terminal versus temporally specific credit: different temporal constructions respond differently to the same nominal reward uncertainty.

The learning-signal diagnostics confirm that output-level corruption is transformed differently before reaching the policy update. On HPSv2, DenseGRPO's process-signal and advantage correlations

are 0.097 and 0.072, compared with 0.650 and 0.275 for Flow-GRPO. No individual diagnostic is a scalar predictor of training robustness: on HPSv3, TP-GRPO has lower advantage correlation than DenseGRPO (0.245 vs. 0.299) despite substantially smaller target-reward degradation. Full diagnostics are reported in Appendix C.

4.2 Mechanistic Controls on the Robustness Gap

We examine three alternative explanations for DenseGRPO's larger degradation using targeted HPSv2 controls: temporal specificity itself, stronger clean optimization, and the covariance structure of local reward errors.

Step-specific credit without local differencing. TreeGRPO assigns edge-specific credit through terminal-descendant backups without forming adjacent differences between intermediate-origin evaluations. At $\alpha = 1$, its HPSv2 reward decreases from 0.3601 to 0.3234 at the common final step, corresponding to a 10.2% drop; best-checkpoint selection gives an almost identical 10.3% drop. Under the primary protocol and the same seed, DenseGRPO drops by 17.4%. A $\beta = 0.01$ TreeGRPO control gives a similar 9.6% drop. Thus, step-specific credit does not by itself reproduce DenseGRPO's sensitivity, although TreeGRPO also differs in rollout structure and aggregation.

Matching clean performance. We select checkpoints using clean HPSv2 runs only: Flow-GRPO reaches 0.3699 at step 3,800 and the closest DenseGRPO checkpoint reaches 0.3682 at step 400. The selected steps are then fixed across corruption levels.

Relative drops at $\alpha = 0.5, 1, 2$ are 5.5%, 11.4%, and 16.6% for Flow-GRPO and 10.8%, 13.8%, and 20.3% for DenseGRPO. The gap therefore persists at nearly matched clean reward and under the alternative clean-selected checkpoint rules described in Appendix A.

Temporal correlation. Equation 3 predicts that positive correlation between neighboring reward errors should reduce corruption introduced by adjacent differencing. We test DenseGRPO at $\rho = 0.5$ and $\rho = 0.9$ while holding marginal error variance fixed at $\alpha = 1$.

DenseGRPO's drop decreases from 17.4% under iid corruption to 15.8% at $\rho = 0.5$ and 10.3% at $\rho = 0.9$, matching the predicted direction. Together, these controls argue against temporal specificity or clean optimization strength alone as explanations of the main gap, while its dependence on error covariance is consistent with local differencing as a particularly vulnerable credit construction.

4.3 Calibration Against Human Disagreement

We next test whether the corruption scale corresponds to uncertainty exhibited by a real reward model. Using 200 DrawBench prompts (Saharia et al., 2022), we collect 4,794 human judgments over 1,598 pairwise comparisons of completed descendants from nearby trajectory states. The corruption reference is fixed before this study and is not refit to the human labels; full protocol details are in Appendix D.

For HPSv3, the calibrated iid probe predicts 40.3% aggregate RM–human rank disagreement, compared with 40.0% observed (95% predictive interval $[37.6\%, 43.0\%]$). It also reproduces the qualitative dependence of disagreement on reward margin.

The iid probe does not reproduce the dependence structure of observed RM–human disagreement. As Table 4 shows, observed disagreement is more concentrated across prompts and trajectory pairs and more persistent across neighboring states. Adjacent error correlation is 0.187 versus an iid-model mean of 0.048, and disagreement persists after a preceding disagreement in 53.4% of cases versus 43.2% predicted.

The calibrated probe therefore captures the marginal scale of HPSv3 RM–human disagreement in this study, but not its temporal or group-level dependence structure. The correlated-noise control in Section 4.2 tests one such departure from independence without identifying observed binary disagreement correlation with the latent reward-error correlation.

Table 2: HPSv2 clean-selected operating points, held fixed across corruption levels. The methods begin from nearly identical clean target reward, yet DenseGRPO loses more reward at every tested corruption level.

Method / selected step	$\alpha = 0$	$\alpha = 0.5$	$\alpha = 1$	$\alpha = 2$
Flow-GRPO / 3,800	0.3699	0.3494	0.3278	0.3085
DenseGRPO / 400	0.3682	0.3286	0.3172	0.2933

Table 3: HPSv2 correlation control at $\alpha = 1$. Increasing positive correlation reduces DenseGRPO's degradation while preserving the marginal reward-error variance.

Method	Error structure	Clean	Corrupted	Drop (%)
Flow-GRPO	terminal	0.3699	0.3355	9.3
DenseGRPO	iid	0.4037	0.3334	17.4
DenseGRPO	$\rho = 0.5$	0.4037	0.3400	15.8
DenseGRPO	$\rho = 0.9$	0.4037	0.3620	10.3

4.4 Reward–Human Divergence under Clean Training

Finally, we evaluate checkpoints trained normally on HPSv3 with no injected corruption. On 200 held-out DrawBench prompts, we compare Flow-GRPO and TP-GRPO separately against DenseGRPO using matched generation seeds and three blind human judgments per pair.

DenseGRPO attains the highest mean HPSv3 score on these images (14.45, versus 13.86 for Flow-GRPO and 14.20 for TP-GRPO), yet humans prefer Flow-GRPO and TP-GRPO over DenseGRPO in 81.0% and 79.8% of tie-adjusted comparisons. Among decisive comparisons in which HPSv3 itself favors DenseGRPO, humans instead select Flow-GRPO in $109/121$ cases (90.1%) and TP-GRPO in $91/105$ cases (86.7%).

Across the controlled and clean-training evaluations, the local-increment construction that is most sensitive to controlled reward corruption is also associated with substantial reward–human divergence under ordinary HPSv3 optimization. We do not interpret corruption sensitivity as a general predictor of human preference, and Flow-GRPO and TP-GRPO are not directly compared with each other.

External evaluator checks. On a separate DrawBench evaluation, DenseGRPO again has the highest optimized HPSv3 score but the lowest PickScore, HPSv2, and UnifiedReward scores among the primary methods; DeQA gives a different ordering. We therefore treat learned evaluators as complementary evidence and use the direct human comparison above as the primary result. Full results are in Appendix E.

5 Discussion and Limitations

Our results shift the robustness question from whether credit is temporally specific to how uncertain reward evidence is transformed before reaching the policy update. The distinction matters because temporal information can provide finer-grained credit without local differencing of uncertain reward evaluations.

Design implications. The experiments point toward preserving temporal structure while making the construction of credit itself uncertainty-aware. Local differences between nearby reward evaluations are particularly exposed when the underlying reward margins are small, whereas trajectory-level aggregation and descendant-based backups provide examples of temporally specific credit with different robustness profiles. More broadly, reward reliability should be assessed after the transformations that produce the learning signal (including aggregation, ranking, and normalization), not only at the raw evaluator output. These results do not identify a universally optimal credit construction, but they suggest that robustness should be treated as a property of the full reward-to-credit mapping rather than of temporal resolution alone.

Table 4: HPSv3 RM–human disagreement structure in the repeated-judgment study. Intervals are predictive intervals under the fixed calibrated iid probe conditioned on the observed reward margins.

Statistic	Observed	Iid mean [95% PI]
Prompt-level error variance	0.065	0.049 [0.041, 0.058]
Trajectory-pair error variance	0.120	0.099 [0.089, 0.109]
Adjacent error correlation	0.187	0.048 [−0.023, 0.119]
$\Pr(e_{t+1} = 1 \mid e_t = 1)$	53.4%	43.2% [37.7%, 48.8%]

Table 5: Blind human evaluation of models trained on the original HPSv3 reward with no injected corruption. Preferences favor the first method in each comparison. Ties receive half weight in the reported preference; 95% confidence intervals use pair bootstrap. Decisive-majority wins use only pairs with a decisive human majority.

Comparison	Preference [95% CI]	Decisive wins
Flow-GRPO vs DenseGRPO	81.0% [77.8%, 84.2%]	91.1%
TP-GRPO vs DenseGRPO	79.8% [76.2%, 83.3%]	90.8%

Relation to human preference. The human studies also clarify what the controlled corruption experiment does and does not measure. Its calibrated scale captures the marginal magnitude and margin dependence of HPSv3 disagreement reasonably well, but observed RM–human disagreements are more concentrated and temporally persistent than predicted by the iid probe. At the same time, the clean-training comparison shows that stronger optimization of the learned reward can coexist with both greater sensitivity to perturbations of that reward and substantial divergence from human preference. We therefore view target reward, robustness to evaluator uncertainty, and direct preference evaluation as complementary signals rather than interchangeable measures of progress.

Limitations. Our primary intervention uses calibrated zero-mean iid output corruption, with a targeted temporal-correlation control. Real reward-model errors may additionally be biased, content- or prompt-dependent, heteroscedastic, or strategically exploitable by the policy. The repeated human judgments directly show that iid corruption understates the concentration and temporal persistence of real disagreement.

The study is restricted to SD3.5-Medium and three learned reward models. The TreeGRPO, matched-clean, and correlation controls use a single HPSv2 seed and are intended to test specific explanations rather than estimate fully tuned performance frontiers. The compared methods also differ in evaluated outcomes, rollout structure, normalization, regularization, and computational cost, so our experiments identify credit construction as an important source of sensitivity without isolating local differencing as the sole causal variable. Finally, the direct human comparison is limited to HPSv3-trained models and the evaluated DrawBench prompts, and therefore establishes a concrete reward–preference divergence in this setting rather than a universal ordering of credit-assignment methods.

Conclusion. Temporal credit under learned rewards is not inherently fragile. The main design issue is how uncertain reward evidence is converted into the signals that drive optimization. A useful direction for future T2I RL is to retain informative temporal structure while designing credit constructions that remain reliable when the reward model is imperfect.

AI USE STATEMENT

We used AI assistants in two ways. For research execution, AI coding assistants helped write training and evaluation code and make figures; the authors set the research direction, reviewed and tested all code, and recomputed every reported number from the stored run outputs. For writing, AI assistants helped edit and polish author-written text. AI tools were not used to generate data, and the method and its derivations are the authors' own. The authors take full responsibility for the content of this paper.

ETHICS STATEMENT

This work studies the sensitivity of text-to-image reinforcement learning to imperfect learned rewards. Stronger reward optimization can amplify reward-specific artifacts or other undesirable behavior, so the reported target-reward gains should not be interpreted as evidence of safe deployment. Human evaluations involved paid annotators, informed consent, and ethics review. We report uncertainty and non-decisive judgments where available and restrict the interpretation to the evaluated settings. We use existing model and data assets and do not release a new image generator or dataset.

REPRODUCIBILITY STATEMENT

Appendix A reports the shared experimental settings, method-specific KL coefficients, seed scope, and checkpoint-selection rule. Appendix B specifies the calibration data, fitting comparison, and RM-specific conversion to train-time corruption scales. Appendix C defines noise injection, correlation interventions, and diagnostic calculations. Appendices D and E describe the human comparisons and give the complete external-metric table. These details distinguish the primary sweep from additional evidence and make explicit the assumptions behind the analytical illustration.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023.

Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: The method of paired comparisons. *Biometrika*, 39(3-4):324–345, 12 1952. ISSN 0006-3444. doi: 10.1093/biomet/39.3-4.324. URL https://doi.org/10.1093/biomet/39.3-4.324.

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 (ICLR)*, 2026.

Zheng Ding and Weirui Ye. TreeGRPO: Tree-advantage GRPO for online RL post-training of diffusion models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=3rZdp4TmUb.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12606–12633, 2024.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pp. 10835–10866. PMLR, 2023.

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 (ICLR)*, 2026.

Yunqi Hong, Kuei-Chun Kao, Hengguang Zhou, and Cho-Jui Hsieh. Understanding reward hacking in text-to-image reinforcement learning. In *Findings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR Findings)*, 2026.

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

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.

Jeongjae Lee and Jong Chul Ye. PCPO: Proportionate credit policy optimization for aligning image generation models. *arXiv preprint arXiv:2509.25774*, 2025. URL https://arxiv.org/abs/2509.25774.

Kyungmin Lee, Xiahong Li, Qifei Wang, Junfeng He, Junjie Ke, Ming-Hsuan Yang, Irfan Essa, Jinwoo Shin, Feng Yang, and Yinxiao Li. Calibrated multi-preference optimization for aligning diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 18465–18475, 2025.

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 (NeurIPS)*, volume 38, pp. 45650–45685, 2025. doi: 10.52202/085713-1362.

Qiang Lyu, Zicong Chen, Chongxiao Wang, Haolin Shi, Shibo Gao, Ran Piao, Youwei Zeng, Jianlou Si, Fei Ding, Jing Li, Chun Pong Lau, and Weiqiang Wang. Multi-GRPO: Multi-group advantage estimation for text-to-image generation with tree-based trajectories and multiple rewards. *arXiv preprint arXiv:2512.00743*, 2025. URL https://arxiv.org/abs/2512.00743.

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.

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. In *Advances in Neural Information Processing Systems*, volume 35, pp. 36479–36494, 2022.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. *Advances in Neural Information Processing Systems*, 35:9460–9471, 2022.

Yunze Tong, Mushui Liu, Canyu Zhao, Wanggui He, Shiyi Zhang, Peng Zhang, Hongwei Zhang, Jinlong Liu, and Hao Jiang. Alleviating sparse rewards by modeling step-wise and long-term sampling effects in flow-based GRPO. In *International Conference on Machine Learning (ICML)*, 2026.

Yibin Wang, Yuhang Zang, Hao Li, Cheng Jin, and Jiaqi Wang. Unified reward model for multimodal understanding and generation. *arXiv preprint arXiv:2503.05236*, 2025.

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.

Shuchen Xue, Chongjian Ge, Shilong Zhang, Yichen Li, and Zhi-Ming Ma. Advantage weighted matching: Aligning RL with pretraining in diffusion models. In *International Conference on Machine Learning (ICML)*, 2026.

Zhiyuan You, Xin Cai, Jinjin Gu, Tianfan Xue, and Chao Dong. Teaching large language models to regress accurate image quality scores using score distribution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14483–14494, 2025.

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 (NeurIPS)*, volume 38, pp. 117621–117654, 2025. doi: 10.52202/085713-3545.

A Full Experimental Setup

This section gives the details of the experimental protocol in Section 3.5: the shared training setup, the exact evaluated outcomes used by intermediate-credit methods, and the configurations of the targeted controls reported in Section 4.2.

Primary comparison protocol. The primary sweep compares Flow-GRPO, PCPO, DenseGRPO, and TP-GRPO under the controlled reward-corruption intervention defined in Section 3.3. The implementations share the backbone, LoRA parameterization, prompt splits, image resolution, denoising-step counts, optimizer family, learning rate, and checkpoint-selection criterion. Reward aggregation, advantage construction, timestep weighting, clipping objectives, KL coefficients, and method-specific sampling remain those of the corresponding configurations. Most methods also share prompt-group and batch sizes; DenseGRPO with PickScore uses the larger sampling configuration listed below. The comparison therefore measures sensitivity to reward corruption within the specified method implementations; it does not equalize reward-model evaluation counts or computational cost.

Backbone, data, and sampling. All primary experiments use SD3.5-Medium (Esser et al., 2024) with LoRA (Hu et al., 2022) rank 32 and scaling factor 64 at 512×512 resolution. The prompt split distributed with the Flow-GRPO codebase contains 25,432 training prompts and 2,048 validation prompts. Training rollouts use 10 denoising steps and classifier-free guidance scale 4.5. Prompt groups normally contain 16 images generated from different initial latents; DenseGRPO with PickScore uses 24. The primary implementations optimize nine transitions per trajectory.

Validation uses 40 denoising steps, guidance scale 4.5, and one image per validation prompt per evaluation. The validation criterion is mean valid target reward under the original, unperturbed reward model. DrawBench (Saharia et al., 2022) evaluation uses 200 prompts and five generated images per prompt.

Optimization and checkpoint selection. Table 6 summarizes the primary configurations. Validation is triggered every 50 sampling/training epochs, including initialization. An epoch consists of the configured sampling batches followed by one inner optimization pass; the configurations are designed for two optimizer updates per epoch. Validation uses EMA parameters, and the checkpoint with the highest clean validation target reward is selected. Here, "clean" means that no synthetic reward corruption is applied; the learned reward model remains an imperfect evaluator. External evaluators and human comparisons are applied only after checkpoint selection and are never selection criteria.

The three-seed reporting and the 144-run count below apply to the primary four-method sweep. The TreeGRPO, matched-clean-performance, correlated-corruption, and human studies use the separate configurations described below or in their respective appendix sections.

Method-specific regularization. The KL coefficient is $\beta = 0.01$ for Flow-GRPO, PCPO, and DenseGRPO and $\beta = 10^{-4}$ for TP-GRPO. The primary TreeGRPO control uses no explicit KL regularization ($\beta = 0$); an additional $\beta = 0.01$ configuration matches the coefficient used by the primary Flow-GRPO, PCPO, and DenseGRPO runs. We therefore do not treat the experiments as sharing a common KL penalty across methods.

Evaluated outcomes for intermediate-origin credit. Section 3.2 summarizes the evaluated-outcome collections $\mathcal{E}_a(\tau)$; we specify them here.

Following the source implementation, DenseGRPO retains the first ten latent states of the ten-step sampled trajectory together with the first nine scheduler entries used for intermediate reward estimation. For each of the first nine retained states, the implementation applies a deterministic ODE continuation over the corresponding suffix of the retained scheduler grid and decodes the resulting endpoint for reward evaluation; the tenth retained state is decoded directly. These ten scalar evaluations yield nine successive reward differences, while the separately generated terminal image is also scored.

Table 6: Primary-sweep training configuration. Values in parentheses denote the DenseGRPO/PickScore exception. Gradient accumulation is expressed in trajectory batches; the implementation also accumulates over the nine optimized transitions.

Component	Value
Backbone / trainable parameters	SD3.5-Medium / LoRA adapters
LoRA rank / scaling	32 / 64
Resolution / hardware	512×512 / $8\times$H100
Optimization horizon	4,500 steps per run
Train / validation denoising steps	10 / 40
Optimized transitions per trajectory	9
Guidance scale	4.5
Prompt group size	16 images (24)
Per-device sampling / training batch	8 / 8 (9 / 9)
Sampling batches per epoch	4 (16)
Gradient accumulation	2 (8) trajectory batches
Train / validation prompts	25,432 / 2,048
Validation samples / aggregation	1 per prompt / mean valid target reward
DrawBench samples	200 prompts, 5 images per prompt
Optimizer / learning rate	AdamW / 3×10^{-4}
Adam betas / epsilon	$(0.9, 0.999)$ / 10^{-8}
Weight decay / max gradient norm	10^{-4} / 1.0
Advantage clipping	5
Precision / TF32	fp16 / enabled
Text embeddings	Maximum sequence length 128; optional precomputation
Primary-sweep seeds	42, 43, 44
Checkpoint evaluation	Every 50 sampling/training epochs
Checkpoint selection	Highest clean validation target reward
Corruption multipliers	$\alpha \in \{0, 0.5, 1, 2\}$

TP-GRPO evaluates the final mixed SDE/ODE sample, a pure-ODE reference, and intermediate ODE completions. The final sampled image is also reused as the last entry of the ordered reward sequence. The complete transformation of these evaluations into TP-GRPO credit is given in Appendix C.

TreeGRPO control. The TreeGRPO experiment in Section 4.2 uses HPSv2 and seed 42. The tree has binary branching at four positions, with one tree per prompt and two prompts per device, producing 16 terminal leaves and 15 branching nodes per prompt. The four branching positions shift every 750 optimizer steps.

Terminal-leaf rewards are normalized within each prompt's tree using the population standard deviation plus 10^{-8}. Recursive child-mean backups then assign each edge the mean normalized reward of its descendant leaves. A corrupted leaf score is sampled once and reused everywhere that leaf enters the recursive backup.

The primary TreeGRPO configuration uses $\beta = 0$, learning rate 10^{-5}, one branching node per training batch, and gradient accumulation over the 15 branching nodes. Validation is triggered every 50 optimizer steps. The clean and corrupted runs reported in the main text are compared at the same final step, 4,500. Selecting each run's best checkpoint instead gives the same qualitative result. We also run a $\beta = 0.01$ control to verify that the observed robustness profile is not specific to the absence of explicit KL regularization.

TreeGRPO uses different sampling and update units from the primary trajectory methods, so matching terminal-image counts does not imply matched computational cost.

Matched-clean-performance control. The matched-operating-point experiment in Section 4.2 uses Flow-GRPO and DenseGRPO with HPSv2 and seed 42. Checkpoint selection is performed using the clean runs only. Flow-GRPO reaches its clean peak of 0.3699 at step 3,800. We then choose

the DenseGRPO checkpoint with clean validation reward closest to that value, which occurs at step 400 with reward 0.3682.

These two selected steps are fixed before examining the corrupted runs and are held constant across $\alpha \in \{0, 0.5, 1, 2\}$. Thus, corrupted validation performance does not influence the operating-point selection. We also verified the same qualitative ordering using best-checkpoint selection within each fixed clean-selected budget and using the earliest DenseGRPO checkpoint that crosses Flow-GRPO's clean target.

Temporally correlated corruption control. The covariance control in Section 4.2 uses DenseGRPO with HPSv2 and seed 42 at the reference corruption level $\alpha = 1$. In addition to the iid condition, we train with stationary AR(1) reward errors at $\rho = 0.5$ and $\rho = 0.9$ while holding the marginal error variance fixed. The seed-42 Flow-GRPO run at $\alpha = 1$ is reported as a terminal-reward reference. The exact AR(1) construction and its relationship to the local-difference variance are given in Appendix C.

Compute scope. The primary sweep comprises three reward models, four credit constructions, four corruption levels, and three seeds, for

$$3 \times 4 \times 4 \times 3 = 144$$

training runs. Each run uses $8\times$H100 GPUs for 4,500 optimization steps and takes approximately 1–3 wall-clock days, corresponding to 8–24 H100 GPU-days. The resulting estimate for the primary sweep is 1,152–3,456 H100 GPU-days, including periodic validation.

This estimate does not include the additional TreeGRPO, temporally correlated corruption, matched-clean-performance, human-evaluation, or external-evaluation experiments.

Existing assets. We use public model and data assets and cite their original sources: SD3.5-Medium, HPDv2, DrawBench, PickScore, HPSv2, HPSv3, DeQA, UnifiedReward-2.0, and Qwen3-VL-8B-Instruct. The SD3.5-Medium checkpoint is used under the Stability AI Community License. We do not redistribute external model weights or datasets. Asset-specific preprocessing and reward-score normalization remain part of the corresponding evaluation pipelines.

B Reward-Specific Corruption Scale and Calibration

This section details the reward-specific scale construction introduced in Section 3.4: the preference data and Bradley–Terry fit, uncertainty in the fitted scale, a comparison of preference calibration with simple score standardization, and the conversion from calibration units to the fixed corruption scales used during training.

Preference pairs and Bradley–Terry fit. For each reward model r, we evaluate 3,300 adjacent-rank preference pairs from HPDv2 (Wu et al., 2023). We use adjacent-rank pairs (Δ-rank 1) to emphasize relatively small preference margins, where perturbations are most likely to alter sample rankings and normalized advantages.

For reward margin

$$\Delta s = s_1 - s_2$$

and binary label $y = 1$ when image 1 is preferred, we fit

$$p(y = 1 \mid \Delta s) = \mathrm{sigmoid}\left(\frac{\Delta s}{\tau_r}\right), \qquad \tau_r > 0,$$

by minimizing binary negative log-likelihood. We perform the fit in both raw reward units and globally standardized reward units.

For N retained comparisons, standardization pools the $2N$ reward-score endpoints, counting an image again if it appears in multiple comparisons. Let s_{fit} denote the resulting sample standard deviation, using denominator $2N - 1$. The standardized margin is then

$$d_i = \frac{\Delta s_i}{s_{\mathrm{fit}}},$$

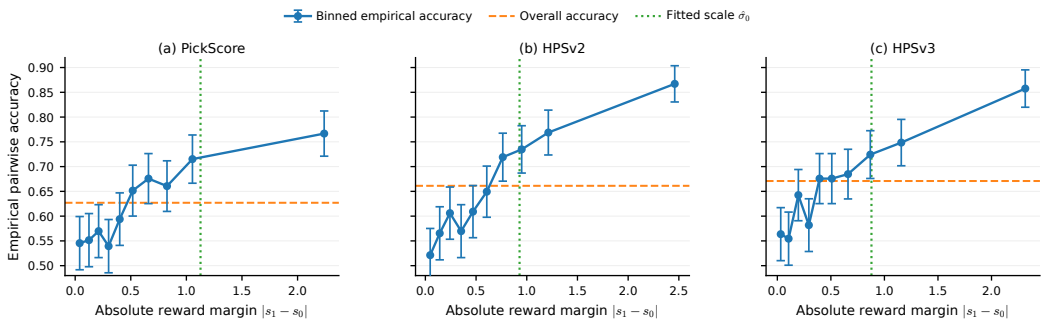


Figure 2: Preference agreement versus absolute standardized reward margin on HPDv2 adjacent-rank pairs. Error bars are 95% normal-approximation binomial intervals and horizontal dashed lines show aggregate preference accuracy. The vertical reference is the fitted standardized Bradley–Terry temperature $\hat{\tau}_r^{\text{norm}}$.

since the common score mean cancels from pairwise differences. The original one-dimensional fit searches 2,000 logarithmically spaced temperatures in $[10^{-4}, 100]$.

The fitted temperature is distribution-dependent: it characterizes the scale over which preference probability changes with reward margin on this calibration distribution. It is not an intrinsic uncertainty parameter of the reward model, and the relevance of the HPDv2 adjacent-rank margin distribution to a particular RL rollout distribution remains an empirical question.

Bradley–Terry temperature and corruption scale are distinct. The fitted temperature τ_r and the Gaussian corruption standard deviation $\sigma_{0,r}$ play different roles. The former parameterizes a preference model; the latter defines the controlled intervention used in training. We use the calibration only to establish an operational, reward-specific reference scale and do not identify the Bradley–Terry likelihood with a generative model of reward-model error.

At corruption level α, training uses

$$\sigma_r(\alpha) = \alpha\sigma_{0,r}.$$

The mapping from the calibration fit to $\sigma_{0,r}$ is reward-model specific and is reported in Table 9. This convention defines the stress-test axis used in the main experiments; it does not assume that real reward-model errors are independent, Gaussian, unbiased, homoscedastic, or constant across prompts and trajectories.

Margin-dependent preference agreement. Figure 2 visualizes empirical preference agreement as a function of absolute standardized reward margin. For each reward model, adjacent-rank pairs are partitioned into ten empirical quantile bins of $|d_i|$, with duplicate bin edges merged. Within each bin, we report empirical preference agreement with the normal-approximation interval

$$\hat{p} \pm 1.96\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}.$$

Agreement generally increases with reward-margin magnitude, so a single aggregate pairwise accuracy does not determine the reliability of small reward differences. The vertical reference in Figure 2 is the fitted standardized Bradley–Terry temperature $\hat{\tau}_r^{\text{norm}}$. It is a characteristic margin scale, not a hard reliability threshold.

Bootstrap uncertainty. We quantify uncertainty in the fitted temperature using 200 bootstrap resamples of the calibration pairs. For each resample, score normalization and the Bradley–Terry fit are recomputed from scratch. Table 7 reports the 5th, 50th, and 95th percentiles of the resulting raw- and standardized-temperature distributions. These outer quantiles form a central 90% bootstrap range; the bootstrap median need not equal the point estimate fitted on the complete dataset.

This pair-level bootstrap does not account for dependence among comparisons sharing a prompt. The prompt-group bootstrap used in the standardization comparison below preserves those groups.

Table 7: Bradley–Terry temperature quantiles from 200 bootstrap resamples of the calibration pairs.

RM	$\hat{\tau}_r^{\mathrm{norm}}$			$\hat{\tau}_r^{\mathrm{raw}}$		
	5%	50%	95%	5%	50%	95%
PickScore	1.0163	1.1273	1.2504	1.4965	1.6486	1.8287
HPSv2	0.8606	0.9289	1.0163	0.0384	0.0417	0.0453
HPSv3	0.8034	0.8759	0.9616	5.8376	6.3227	6.9410

Table 8: Five-fold prompt-group comparison of score standardization and Bradley–Terry calibration. Top: held-out NLL and ECE (lower is better). Bottom: full-data normalized temperature with a 95% prompt-group-bootstrap interval and expected clean-order flips at $\alpha = 1$.

RM	NLL std.	NLL BT	ECE std.	ECE BT
PickScore	0.6486	0.6481	0.0182	0.0200
HPSv2	0.6156	0.6153	0.0166	0.0174
HPSv3	0.6191	0.6184	0.0397	0.0338

RM	$\hat{\tau}_r^{\mathrm{norm}}$ [95% interval]	Flip std.	Flip BT
PickScore	1.126 [1.047, 1.209]	0.353	0.368
HPSv2	0.930 [0.871, 0.991]	0.334	0.325
HPSv3	0.881 [0.814, 0.958]	0.350	0.334

Comparison with score standardization. Human labels are useful for assigning an interpretable preference scale, but a simpler alternative is to standardize reward scores and use unit temperature. We compare these choices using five-fold prompt-group cross-validation on the same 3,300 preference pairs. All comparisons from the same prompt/ranking group remain in one fold.

Within each fold, the endpoint standard deviation and Bradley–Terry temperature are estimated on the training groups only. The same training-set standard deviation is then applied to the held-out margins. The fitted model optimizes log temperature over $[10^{-4}, 100]$. Metrics are pooled over the resulting out-of-fold predictions. Expected calibration error (ECE) uses ten equal-width bins of predicted preference probability in $[0, 1]$.

Table 8 shows that the fitted Bradley–Terry temperature changes held-out predictive performance only slightly relative to unit-temperature standardization. The NLL improvements are 0.0005, 0.0003, and 0.0007 for PickScore, HPSv2, and HPSv3, respectively. ECE improves for HPSv3 and is slightly worse for PickScore and HPSv2. Simple standardization is therefore a strong label-free approximation to the relative reward scale, while Bradley–Terry calibration supplies the preference-grounded unit used to define our main operating reference.

For uncertainty on $\hat{\tau}_r^{\mathrm{norm}}$, we also perform 200 prompt-group bootstrap resamples, retaining all comparisons belonging to each sampled group and refitting normalization and temperature. The reported interval is the 2.5th–97.5th percentile range.

Expected order flips under the calibration models. The final two columns of Table 8 give a further scale comparison, not an observed training metric. For standardized calibration margins $d_i = \Delta s_i / s_{\mathrm{fit}}$, independent Gaussian perturbations of the two endpoint scores imply the expected clean-order flip rate

$$\frac{1}{N} \sum_{i=1}^{N} \Phi\left(-\frac{|d_i|}{\sqrt{2}\,\alpha\tau} \right),$$

where $\tau = 1$ for simple standardization and $\tau = \hat{\tau}_r^{\mathrm{norm}}$ for the Bradley–Terry model. Table 8 reports this quantity at $\alpha = 1$.

These values are model-based expected reversals of the clean reward ordering. They are distinct both from observed RM–human disagreement in Section 4.3 and from the downstream degradation measured after RL training.

Table 9: Reward-model-specific mapping to the fixed train-time corruption reference $\sigma_{0,r}$ used at $\alpha = 1$.

RM	$\sigma_{0,r}$	Mapping
PickScore	$1.648574/26 \simeq 0.0634$	Raw BT scale divided by reward normalization factor
HPSv2	0.0417167	Raw BT scale used directly
HPSv3	2.6368973	Standardized BT scale mapped to clean pre-RL rollout reward units

Train-time scale mapping. The primary experiments use the fixed train-time reference scales in Table 9. The conversion is not identical across reward models.

For PickScore, the fitted raw Bradley–Terry scale is divided by the reward normalization factor 26 used by the training pipeline. HPSv2 uses the fitted raw scale directly. For HPSv3, we map the fitted standardized Bradley–Terry temperature back to the reward units observed on clean pre-RL rollouts,

$$\sigma_{0,\mathrm{HPSv3}} = \hat{\tau}^{\mathrm{norm}}_{\mathrm{HPSv3}} s_{\mathrm{roll}},$$

where s_{roll} is the standard deviation of HPSv3 scores measured once on clean rollouts before RL training and then held fixed. This gives the train-time reference $\sigma_{0,\mathrm{HPSv3}} = 2.6368973$. We therefore do not apply a universal rule that multiplies the standardized Bradley–Terry temperature by a rollout reward standard deviation.

Once chosen, each $\sigma_{0,r}$ is fixed before corrupted training and the levels $\alpha \in \{0, 0.5, 1, 2\}$ are obtained by simple multiplication.

C Corruption and Diagnostic Details

This section details the corruption intervention and learning-signal diagnostics introduced in Sections 3.3 and 3.5. We use the notation of Section 3.1: a indexes a credit-assignment method, r a reward model, C_a maps prompt-group reward evidence to the pre-normalization credit signal, and $F_a = N_a \circ C_a$ maps reward evidence to the final normalized advantages.

Evaluation-level corruption. For trajectory i and evaluated outcome $e \in I_a$, let

$$q_{a,i,e} = R_r(x_{a,i,e}, c)$$

denote the clean reward-model output. At corruption level α, we construct

$$\widetilde{q}_{a,i,e} = q_{a,i,e} + \sigma_r \epsilon_{a,i,e}, \qquad \sigma_r = \alpha \sigma_{0,r}, \qquad \epsilon_{a,i,e} \sim \mathcal{N}(0,1).$$

The primary intervention samples independent errors for distinct evaluated outcomes. If the same scalar evaluation appears multiple times in a method's credit construction, its corrupted value is reused rather than independently resampled. For example, TP-GRPO reuses the reward of its final sampled outcome as the last entry of its ordered reward sequence.

All downstream quantities are recomputed separately from the clean and corrupted reward evidence. Thus differences, comparison signs, indicators, trajectory corrections, backups, and normalization in a corrupted run are functions of $\widetilde{\mathbf{Q}}_a$ rather than mixtures of clean and corrupted scores. Only the corrupted advantages enter the policy update. Validation always uses the original unperturbed reward model.

Temporally correlated corruption. The targeted covariance control in Section 4.2 replaces the iid errors along an ordered sequence of reward evaluations with a stationary AR(1) process,

$$\epsilon_0 = u_0, \qquad \epsilon_t = \rho \epsilon_{t-1} + \sqrt{1-\rho^2}\, u_t, \qquad u_t \overset{\mathrm{iid}}{\sim} \mathcal{N}(0,1).$$

The reward perturbation remains $\sigma_r \epsilon_t$, so the marginal error variance is fixed at σ_r^2 while the temporal dependence changes. Innovations are deterministically seeded using the configured training seed and sample/channel identifiers.

The correlation axis applies to ordered reward evaluations, not to repeated algebraic uses of the same already-scored outcome. For adjacent reward differences, the resulting error variance is

$$\mathrm{Var}\left[\sigma_r(\epsilon_{t+1} - \epsilon_t)\right] = 2\sigma_r^2(1-\rho),$$

which yields the directional prediction tested in Section 4.2.

Signals before and after normalization. For a prompt group, let

$$\mathbf{S}_a^{\mathrm{clean}} = C_a(\mathbf{Q}_a), \qquad \mathbf{S}_a^{\mathrm{corr}} = C_a(\widetilde{\mathbf{Q}}_a)$$

denote the clean and corrupted method-specific credit signals before the final normalization step. The corresponding advantages are

$$\mathbf{A}_a^{\mathrm{clean}} = F_a(\mathbf{Q}_a), \qquad \mathbf{A}_a^{\mathrm{corr}} = F_a(\widetilde{\mathbf{Q}}_a).$$

For terminal-reward methods, the underlying signal is derived from the final reward. For DenseGRPO, suppressing method and group indices,

$$d_t = q_{t+1} - q_t, \qquad \widetilde{d}_t = d_t + \sigma_r(\epsilon_{t+1} - \epsilon_t).$$

The reported process-signal diagnostic is computed on the method's resulting credit signal, not on the raw reward-model outputs. TP-GRPO uses the more structured construction specified below.

TP-GRPO credit construction. Let q_0 denote the pure-ODE reward and $q_1, \ldots, q_K$ the ordered completion rewards, with q_K equal to the reward of the final sampled outcome reused as the last trajectory entry. Define

$$d_t = q_t - q_{t-1}, \qquad \Delta = q_K - q_0.$$

The SD3 configuration used in our experiments takes local increments as the default process signal, enables trajectory-dependent bonuses from the first update with $\eta = 1$, and applies trajectory-consistency selection at both the first and subsequent steps.

The candidate indicators are

$$I_1 = \mathbf{1}\left\{d_1(q_K - q_1) > 0\right\},$$

and, for $t \geq 2$,

$$I_t = \mathbf{1}\{d_{t-1}\Delta < 0\}\mathbf{1}\{d_t\Delta > 0\}\mathbf{1}\{d_t(q_K - q_t) > 0\}.$$

Candidate entries are balanced by the sign of Δ within each local sampling batch. If both signs are present, the same number of positive- and negative-Δ entries is retained, keeping the largest $|\Delta|$ values when one sign is overrepresented. If only one sign occurs, the selection is left unchanged. Let J_t denote the resulting selected indicator.

The trajectory-dependent bonus and process signal are

$$B_t = \eta J_t(q_K - q_{t-1}),$$

$$R_t^{\mathrm{TP}} = \begin{cases} B_t, & B_t \neq 0, \\ d_t, & B_t = 0. \end{cases}$$

An active nonzero bonus therefore replaces, rather than adds to, the local increment. Bonuses are not propagated to later steps, and selection is not restricted to trajectories with positive global improvement.

The resulting signals are normalized independently at each timestep within each prompt group using the population standard deviation plus 10^{-4}. The full construction (differences, indicators, balancing, and bonus selection) is recomputed independently for $\mathbf{Q}_a$ and $\widetilde{\mathbf{Q}}_a$ when producing the clean and corrupted diagnostics.

Process-signal and advantage correlation. Let k index the relevant trajectory or trajectory–timestep units of a method-specific process signal. We report

$$\mathrm{SignalCorr}_a = \mathrm{corr}_k \left(S_{a,k}^{\mathrm{clean}}, S_{a,k}^{\mathrm{corr}} \right).$$

The corresponding post-normalization diagnostic is

$$\mathrm{AdvCorr}_a = \mathrm{corr}_{i,t} \left(A_{a,i,t}^{\mathrm{clean}}, A_{a,i,t}^{\mathrm{corr}} \right).$$

Both correlations compare counterfactual clean and corrupted signals on the same sampled trajectories. They are therefore distinct from comparisons between checkpoints reached by separate clean and corrupted training runs. In the DenseGRPO and TP-GRPO implementations used here, arrays are flattened before computing Pearson correlation. If either input standard deviation is below 10^{-12}, the implementation returns zero. The clean diagnostic path is separate from the corrupted advantages used by the policy update.

Within-prompt rank flips. For two eligible samples i and j from the same prompt group and relevant timestep, let $h_{a,i,t}^{\mathrm{clean}}$ and $h_{a,i,t}^{\mathrm{corr}}$ denote the method-specific scalar signals used for within-prompt ordering. Define

$$\Delta_{ij,t}^{\mathrm{clean}} = h_{a,i,t}^{\mathrm{clean}} - h_{a,j,t}^{\mathrm{clean}}, \qquad \Delta_{ij,t}^{\mathrm{corr}} = h_{a,i,t}^{\mathrm{corr}} - h_{a,j,t}^{\mathrm{corr}}.$$

A reversal occurs when

$$\Delta_{ij,t}^{\mathrm{clean}} \Delta_{ij,t}^{\mathrm{corr}} < 0.$$

Clean ties satisfying $|\Delta_{ij,t}^{\mathrm{clean}}| < 10^{-8}$ are excluded. Pooling all eligible within-prompt comparisons in an evaluated batch gives

$$\mathrm{RankFlip}_a = \frac{\sum_{(i,j,t)\in\mathcal{P}} \mathbf{1}\left\{\Delta_{ij,t}^{\mathrm{clean}} \Delta_{ij,t}^{\mathrm{corr}} < 0\right\}}{|\mathcal{P}|}.$$

A corrupted tie is not counted as a reversal, and the implementation returns zero if no eligible comparisons are present.

Complete diagnostics for the primary sweep. Table 10 reports the three diagnostics at $\alpha = 1$ for all four primary methods and three reward models. Signal correlation measures corruption after method-specific credit construction but before final normalization, rank flips measure changes in within-prompt ordering, and advantage correlation includes the complete method-specific normalization. These quantities characterize different stages of $\mathbf{Q}_a \mapsto F_a(\mathbf{Q}_a)$ and need not induce the same ordering across methods.

The diagnostics confirm that the same nominal reward-output corruption can be transformed very differently by different credit constructions. They are not, however, scalar predictors of optimization performance. On HPSv3, TP-GRPO has lower signal and advantage correlations than DenseGRPO despite substantially smaller target-reward degradation, while on HPSv2 its advantage correlation is higher. The main analysis therefore treats these quantities as diagnostics of effective learning-signal corruption, not as standalone robustness metrics.

Local differencing illustration. The main text uses Eq. 3 as a directional mechanistic prediction. Here we give a simple signal-to-noise illustration.

Consider an idealized progress sequence $V_0, \ldots, V_T$ observed with jointly Gaussian error,

$$\widehat{V}_t = V_t + \epsilon_t,$$

where the errors have marginal variance σ^2 and adjacent correlation $\rho < 1$. A local increment is

$$\widehat{r}_t = \widehat{V}_{t+1} - \widehat{V}_t = \delta_t + \eta_t, \qquad \delta_t = V_{t+1} - V_t, \qquad \eta_t = \epsilon_{t+1} - \epsilon_t.$$

Hence

$$\mathrm{Var}(\eta_t) = 2\sigma^2(1-\rho),$$

Table 10: Complete effective-corruption diagnostics at $\alpha = 1$. Rank flips are fractions, not percentages. All quantities compare clean and corrupted signals on matched trajectories rather than outputs of separately trained checkpoints.

RM	Method	Signal corr.	Rank flip	Adv. corr.
PickScore	Flow-GRPO	0.676	0.417	0.275
	PCPO	0.666	0.420	0.262
	DenseGRPO	0.164	0.459	0.127
	TP-GRPO	0.168	0.457	0.146
HPSv2	Flow-GRPO	0.650	0.418	0.275
	PCPO	0.632	0.416	0.273
	DenseGRPO	0.097	0.477	0.072
	TP-GRPO	0.159	0.456	0.156
HPSv3	Flow-GRPO	0.722	0.394	0.382
	PCPO	0.724	0.388	0.377
	DenseGRPO	0.275	0.430	0.299
	TP-GRPO	0.238	0.433	0.245

and, for $\delta_t > 0$,

$$\Pr(\widehat{r}_t < 0) = \Phi\left(-\frac{\delta_t}{\sigma\sqrt{2(1-\rho)}}\right).$$

Suppose, only for this illustration, that fixed total progress

$$\Delta = V_T - V_0 > 0$$

is distributed approximately uniformly over T evaluation intervals, so $\delta_t \simeq \Delta/T$. Define

$$\kappa = \frac{\sigma}{\Delta},$$

which is a toy relative-noise ratio and is distinct from the experimental corruption multiplier $\alpha = \sigma/\sigma_{0,r}$. Then

$$\Pr(\widehat{r}_t < 0) \simeq \Phi\left(-\frac{1}{T\kappa\sqrt{2(1-\rho)}}\right).$$

At fixed Δ and σ, finer partitioning decreases the local signal scale, whereas positive adjacent correlation suppresses differenced noise. These assumptions are illustrative: they do not require actual denoising progress to be monotone or uniform, nor do they imply that increasing the number of denoising steps leaves total progress fixed in a real model.

For comparison, an idealized terminal-improvement signal $\widehat{V}_T - \widehat{V}_0$ has error variance

$$2\sigma^2(1-\rho_T),$$

where ρ_T is the endpoint-error correlation. Its signal-to-noise ratio scales as

$$\frac{\Delta}{\sigma\sqrt{2(1-\rho_T)}},$$

whereas the local ratio scales as

$$\frac{\Delta/T}{\sigma\sqrt{2(1-\rho)}}.$$

This comparison illustrates the local differencing mechanism only. It is not a claim that every terminal-credit method explicitly subtracts an initial reward or that terminal credit universally dominates temporally specific credit.

Gradient-noise view. We can also ask how reward uncertainty enters the policy-gradient estimator. Condition on fixed score-function vectors

$$z_t = \nabla_\theta \log \pi_\theta(a_t \mid x_t, c)$$

and fixed baselines, and ignore normalization and clipping for this local calculation.

A terminal update receives a common reward-error component

$$g^{\text{noise}}_{\text{term}} = \epsilon \sum_t z_t,$$

whereas a temporally varying update receives

$$g^{\text{noise}}_{\text{temp}} = \sum_t z_t \xi_t.$$

The corresponding covariance traces are

$$\operatorname{tr} \operatorname{Cov}\left(g^{\text{noise}}_{\text{term}}\right) = \sigma_R^2 \left\| \sum_t z_t \right\|^2,$$

and

$$\operatorname{tr} \operatorname{Cov}\left(g^{\text{noise}}_{\text{temp}}\right) = \sum_t \sum_s z_t^\top z_s \operatorname{Cov}(\xi_t, \xi_s).$$

There is no universal ordering between these expressions: the result depends on both score-vector geometry and the covariance structure of the reward-derived errors. For adjacent local differences,

$$\xi_t = \epsilon_{t+1} - \epsilon_t,$$

so even independent evaluation errors induce dependence between neighboring credit errors. The calculation motivates measuring corruption after credit construction and normalization rather than treating a common RM-output perturbation scale as a common effective learning-signal perturbation. It does not model clipping, policy-induced distribution shift, prompt-group normalization, or nonlinear TP-GRPO bonus decisions.

D Human-Evaluation Protocols

This section gives the complete data-collection and statistical procedures for the repeated-judgment study in Section 4.3 and the blind clean-training comparison in Section 4.4. The two studies answer different questions. The first evaluates how well the calibrated corruption probe reflects observed RM–human disagreement on trajectory-related comparisons. The second tests whether the reward–human divergence discussed in the main text appears under ordinary training without injected corruption.

Repeated judgments on intermediate-state completions. The first study uses all 200 DrawBench prompts (Saharia et al., 2022). For each prompt, we sample two independent trajectory pairs, giving 400 trajectory pairs in total. For each pair, human annotators compare completed descendants from four neighboring retained states. The final dataset contains 1,598 pairwise tasks and 4,794 individual judgments, with three independent ratings per task.

Annotators may choose the left image, the right image, or a tie. We discard invalid individual ratings and define the human outcome as the unique most frequent valid response among these three options. With three valid ratings, a decisive left/right outcome therefore requires at least two matching votes. Overall, 68% of tasks have a decisive human outcome.

For binary RM–human disagreement analyses, we retain only tasks with a decisive left/right human outcome and a nonzero reward-model margin. For retained task i, define

$$e_i = \begin{cases} 1, & \text{if the RM and human majority prefer different images,} \\ 0, & \text{otherwise.} \end{cases}$$

Overall disagreement is the mean of e_i over the retained tasks.

Prompt- and trajectory-level concentration. To measure whether disagreement is concentrated in particular prompts or trajectory pairs, let $g = 1, \ldots, G$ index either grouping and let $\bar{e}_g$ denote the mean disagreement rate within group g. We report the population variance

$$V_G = \frac{1}{G}\sum_{g=1}^{G}\left(\bar{e}_g - \bar{e}_{\text{groups}}\right)^2, \qquad \bar{e}_{\text{groups}} = \frac{1}{G}\sum_{g=1}^{G}\bar{e}_g.$$

Each group receives equal weight in this summary.

For the confidence interval on overall disagreement, we resample prompts with replacement and retain every eligible task belonging to each sampled prompt. For every bootstrap sample, we recompute the total number of disagreements divided by the total number of retained tasks. The reported 95% confidence interval is the 2.5th–97.5th percentile range over these prompt-cluster bootstrap estimates.

Temporal persistence of disagreement. Within each trajectory pair, retained decisive comparisons are ordered by their original trajectory-state order. Adjacency refers to consecutive entries in this retained sequence; if an intermediate task is excluded because its human outcome is non-decisive, two retained neighboring entries may therefore span that excluded state.

We pool these adjacent retained comparisons to compute the Pearson correlation between successive disagreement indicators. We additionally report the continuation probability

$$\widehat{\Pr}\left(e_{\text{next}} = 1 \mid e_{\text{prev}} = 1\right) = \frac{\sum_{(i,j)\in\mathcal{A}} e_i e_j}{\sum_{(i,j)\in\mathcal{A}} e_i},$$

where $\mathcal{A}$ is the set of adjacent retained entries within trajectory pairs. These quantities characterize the persistence of observed RM–human disagreement and should not be interpreted as direct estimates of the latent AR(1) error parameter ρ used in the controlled corruption experiment.

Predictive comparison with the calibrated iid probe. The corruption reference is fixed before this human study and is not refit to the newly collected labels. For HPSv3, the reference used in training is

$$\sigma_{0,\text{HPSv3}} = 2.6368973$$

in raw HPSv3 reward units.

For an observed HPSv3 margin Δr_i, independently perturbing both endpoint scores with zero-mean Gaussian noise of standard deviation $\sigma_{0,\text{HPSv3}}$ gives the conditional clean-order flip probability

$$p_i^{\text{flip}} = \Phi\left(-\frac{|\Delta r_i|}{\sqrt{2}\sigma_{0,\text{HPSv3}}}\right).$$

We keep the observed reward margins, eligible task set, prompt grouping, trajectory-pair grouping, and state ordering fixed and simulate independent Bernoulli disagreement indicators with probabilities p_i^{flip}. Every statistic reported for the human study is recomputed on this fixed layout.

The predictive mean is the mean of the simulated statistic, and the reported 95% predictive interval is its 2.5th–97.5th percentile range. These intervals characterize the fixed calibrated iid probe conditional on the observed margins and layout. They do not include uncertainty in the fitted calibration parameters.

For HPSv3, this procedure predicts an overall disagreement rate of 40.3% with 95% predictive interval $[37.6\%, 43.0\%]$, compared with 40.0% observed disagreement. The main text reports the corresponding concentration and temporal-persistence comparisons in Section 4.3.

Blind preference comparison of clean-trained checkpoints. The second human study evaluates checkpoints trained with the original HPSv3 reward and no injected corruption ($\alpha = 0$). We separately compare Flow-GRPO against DenseGRPO and TP-GRPO against DenseGRPO on 200 held-out DrawBench prompts. Generation seeds are matched within each method pair, and every image pair receives three independent human judgments. Method identities are hidden from annotators.

For pair i, let c_i, b_i, and t_i denote the numbers of valid votes for the challenger method, DenseGRPO, and tie, respectively. The pair-level tie-adjusted challenger score is

$$u_i = \frac{c_i + \frac{1}{2}t_i}{c_i + b_i + t_i},$$

and the reported preference is

$$\widehat{P} = \frac{1}{N}\sum_{i=1}^{N} u_i.$$

Thus, every image pair receives equal weight regardless of the number or composition of valid votes.

The 95% confidence interval is obtained by resampling image pairs with replacement within each method comparison and recomputing $\widehat{P}$. For the decisive-majority statistic, we retain only pairs for which at least two annotators prefer one of the two methods. Majority ties, ambiguous outcomes, and invalid comparisons are excluded from that denominator.

The main-text preferences are 81.0% [77.8%, 84.2%] for Flow-GRPO over DenseGRPO and 79.8% [76.2%, 83.3%] for TP-GRPO over DenseGRPO. The corresponding decisive-majority win rates are 91.1% and 90.8%.

Reward–human reversal conditional on HPSv3 preference. To distinguish the overall human preference result from cases in which the optimized reward itself favors DenseGRPO, we also condition on human-decisive pairs with a nonzero HPSv3 margin favoring DenseGRPO.

Within this subset, humans instead prefer Flow-GRPO in $109/121$ comparisons (90.1%) and TP-GRPO in $91/105$ comparisons (86.7%). These are conditional reversal rates and should not be interpreted as overall human preference frequencies. Flow-GRPO and TP-GRPO are not directly compared with one another in this study.

Study conduct and interpretation. Annotators were paid, provided informed consent, and participated under ethics review. Human preference can be heterogeneous, and ties and non-decisive outcomes are retained as part of the uncertainty of the evaluation rather than forced into binary labels.

The repeated-judgment study tests the correspondence between a fixed corruption probe and observed RM–human disagreement on the evaluated trajectory-derived comparisons. The clean-training study tests reward–human divergence for HPSv3-trained models on the evaluated DrawBench prompts. Neither study establishes a universal ranking of credit-assignment methods across reward models, prompt distributions, or training regimes.

Confidence intervals in the clean-training study quantify sampling uncertainty of an observed preference estimator. Predictive intervals in the repeated-judgment study instead characterize statistics generated under the specified corruption model conditional on the observed comparison layout.

E Full External-Evaluator Results

This section gives the full results of the external-evaluator check summarized in Section 4.4. The direct human comparison is our primary evidence for reward–preference divergence; the learned evaluators below additionally measure how strongly optimization of one target transfers to other reward models.

We evaluate clean checkpoints on DrawBench (Saharia et al., 2022) using PickScore, HPSv2, HPSv3, DeQA (You et al., 2025), and UnifiedReward-2.0 built on Qwen3-VL-8B-Instruct (Wang et al., 2025; Bai et al., 2025). Evaluation uses 200 prompts and five generated images per prompt. Within each training-reward block in Table 11, the column corresponding to the training reward is the optimized target; all other columns are external learned evaluators.

The transfer patterns depend strongly on the optimized reward. For PickScore training, TP-GRPO attains the highest optimized PickScore, whereas PCPO is highest under several external evaluators. For HPSv2 training, TP-GRPO obtains both the highest optimized HPSv2 score and the highest HPSv3 score, while Flow-GRPO is favored by the remaining external metrics.

Table 11: Full DrawBench evaluation of clean checkpoints. Blocks identify the reward optimized during training; bold indicates the highest value within each training-reward block and evaluator. External learned rewards complement, but do not replace, the direct human comparison in Section 4.4.

Method	PickScore	HPSv2	HPSv3	DeQA	UnifiedReward
Training reward: PickScore					
Flow-GRPO	0.9007	0.2941	10.116	4.088	3.112
PCPO	0.9034	**0.3050**	**10.460**	**4.234**	**3.121**
DenseGRPO	0.9085	0.2783	9.317	3.841	3.086
TP-GRPO	**0.9109**	0.2974	10.306	3.993	3.120
Training reward: HPSv2					
Flow-GRPO	**0.8875**	0.3271	11.342	**4.377**	**3.103**
PCPO	0.8824	0.3296	11.092	4.349	3.086
DenseGRPO	0.8712	0.3353	10.217	3.661	3.049
TP-GRPO	0.8810	**0.3428**	**11.604**	3.858	3.076
Training reward: HPSv3					
Flow-GRPO	**0.8798**	**0.3045**	11.608	4.349	**3.101**
PCPO	0.8530	0.2925	12.020	4.365	3.035
DenseGRPO	0.8056	0.2596	**13.557**	**4.440**	2.815
TP-GRPO	0.8649	0.2998	12.512	4.177	3.051

The strongest target-versus-external divergence occurs under HPSv3 training. DenseGRPO obtains the highest optimized HPSv3 score but the lowest PickScore, HPSv2, and UnifiedReward scores among the four primary methods. DeQA produces a different ordering and assigns DenseGRPO the highest score. This disagreement between evaluators is itself a reason not to treat learned reward models as interchangeable human-preference oracles, and the main text therefore relies on the blind human study instead of using external-evaluator consensus as a substitute for human preference.

F Qualitative Examples

This section gives qualitative context for the external-evaluator differences in Appendix E and the direct human comparison in Section 4.4. All images are generated from clean checkpoints selected using the original validation target reward. The examples are illustrative, not a population-level preference estimate, and are not used for model selection or quantitative conclusions.

PickScore-trained checkpoints. Figure 3 shows representative samples from models trained to optimize PickScore. The methods differ in texture, contrast, composition, and visual style even when the prompt semantics are broadly preserved. These examples illustrate why improvement under one learned reward need not transfer uniformly to other evaluators.

HPSv2-trained checkpoints. Figure 4 provides the corresponding comparison for HPSv2-trained models. Again, optimization changes visual characteristics in method-dependent ways, while the external metrics in Appendix E disagree on which changes are preferable.

HPSv3-trained checkpoints. The HPSv3 setting shows the clearest quantitative divergence between the optimized target and other evaluators. Figure 5 provides qualitative examples from these checkpoints. DenseGRPO samples in this selected display include strongly stylized outputs and occasional semantic or compositional shifts. These examples are consistent with the reward-specific divergence measured quantitatively, but the direct blind human comparison in Section 4.4 provides the stronger evidence for the reward–preference reversal.

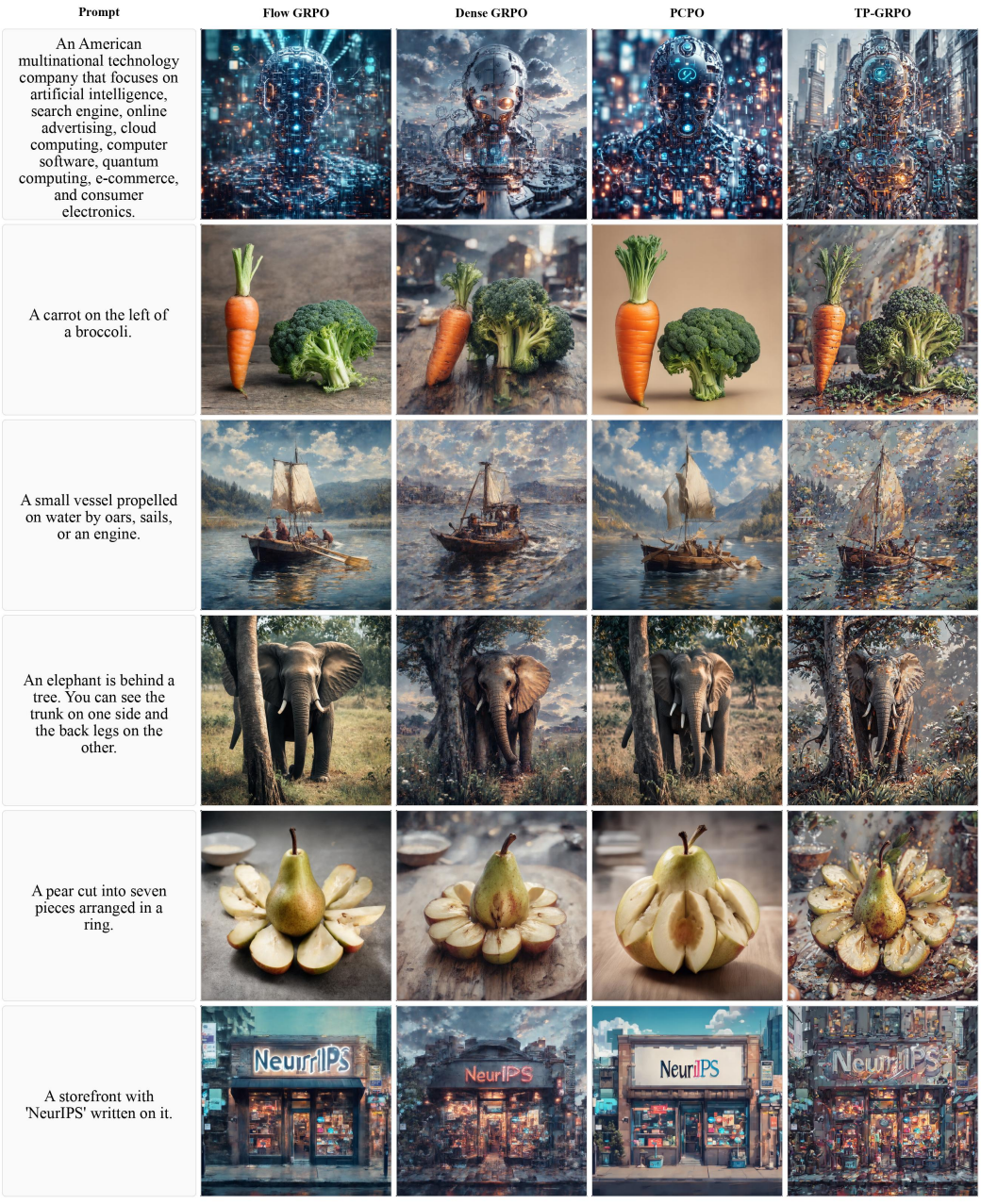


Figure 3: Examples from PickScore-trained clean checkpoints. Rows correspond to prompts and columns to credit-assignment methods. The selected samples illustrate method-dependent changes in texture, contrast, composition, and style.



Figure 4: Examples from HPSv2-trained clean checkpoints. Rows correspond to prompts and columns to credit-assignment methods. The samples provide qualitative context for the external-evaluator comparison.

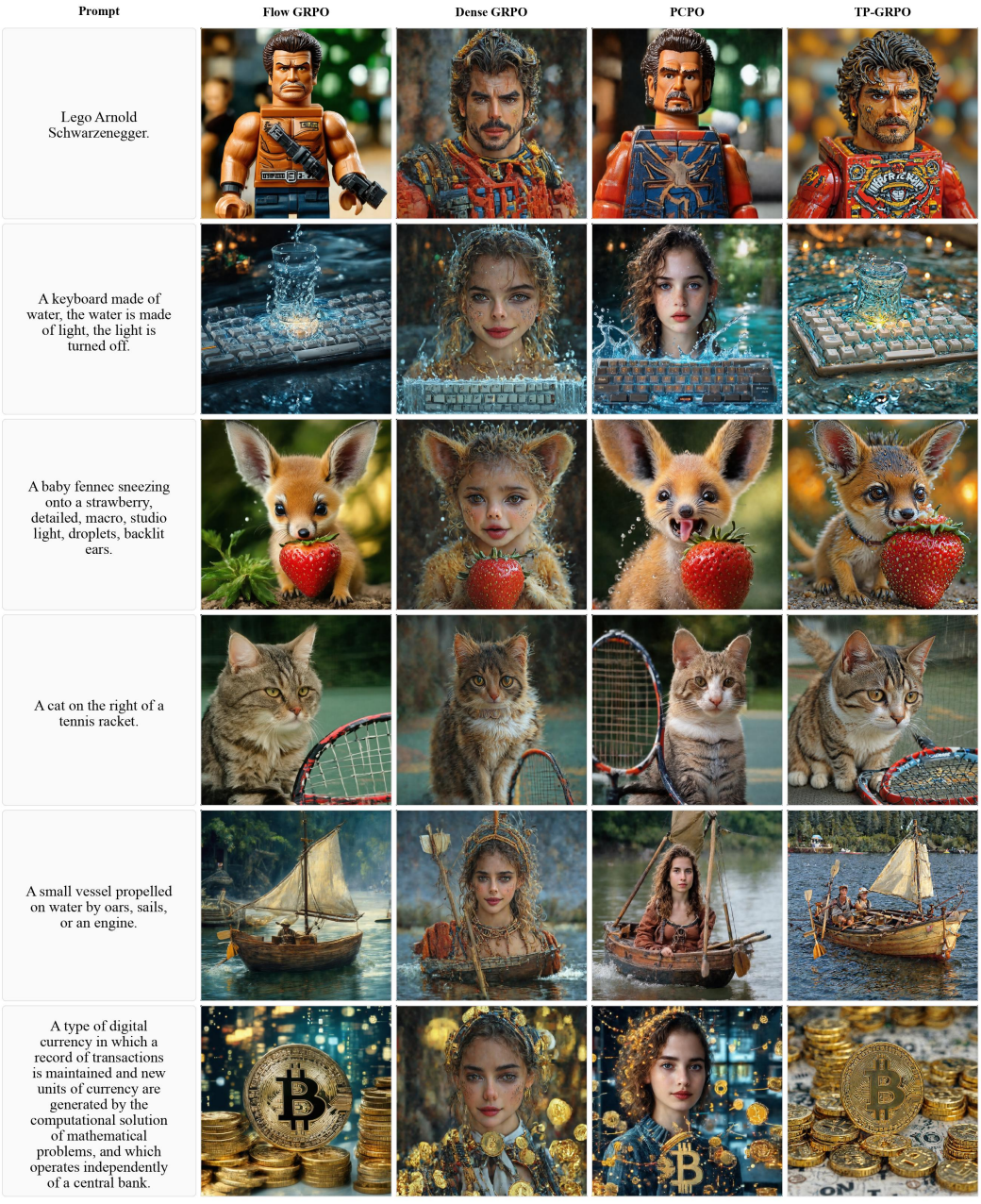


Figure 5: Examples from HPSv3-trained clean checkpoints. Rows correspond to prompts and columns to credit-assignment methods. These selected examples illustrate visual changes accompanying stronger optimization of the HPSv3 target; quantitative human-preference evidence is reported separately in Section 4.4.