TRUSTROBOREWARD: PREFERENCE-ORDERED ISOTONIC SCORE EDITING FOR MULTI-PARADIGM ROBOT REWARD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reward models are a core bottleneck for reinforcement learning in embodied AI. Long-horizon robotic manipulation requires scalable vision-based feedback, which cannot be fully satisfied by manually crafted reward functions or task-specific human annotations. Existing open-source VLM reward judges such as RoboReward adopt a simple 1–5 trajectory progress scoring scheme, lacking pairwise preferences essential for modern RLHF, DPO and Bradley-Terry frameworks, while also failing to optimize agents' video scene understanding directly. Augmenting RoboReward with pairwise comparison and video-QA supervision causes inherent inconsistency between pairwise preferences and pointwise scores, introducing training noise and hurting downstream performance—an issue that aggregation methods such as TrustJudge cannot resolve. To address this, we propose TrustRoboReward, a multi-paradigm reward modeling framework equipped with the Preference-Ordered Isotonic Score Editing (POISE) module. We construct a unified four-paradigm dataset containing trajectory progress scoring (Score-A), video-QA answer quality scoring (Score-B), and their pairwise preference counterparts (Pair-A, Pair-B). We find pairwise labels align better with human judgment than pointwise scores, inspiring us to calibrate pointwise scores to avoid score-pair reversals against pairwise preferences. The POISE module rectifies pointwise scores and eliminates cross-paradigm reversal conflicts unresolved by TrustJudge. Theoretically, POISE provably reduces training-corpus score-pair reversal conflicts from 20.15% to 0%, whereas TrustJudge still retains 20.46% reversal conflicts on the same training corpus. Evaluated on our multi-paradigm benchmark, the Qwen3-VL-4B model trained with POISE achieves an overall reward score of 77.96%, nearly matching GPT-5-mini (78.09%, gap 0.13%) and outperforming the strongest RoboReward-4B baseline by 10.13%. It also lifts test-time score-pair consistency to 71.90%, exceeding both RoboReward-4B (57.26%) and GPT-5-mini (68.09%). Further integrating TrustJudge aggregation during inference boosts our model's overall score to 78.57%, surpassing the proprietary GPT-5-mini teacher model.

1 INTRODUCTION

Despite advances in reinforcement learning (RL) for robot policy training (Ma et al., 2024; Luo et al., 2024; Liu et al., 2023; Guo et al., 2025b; Kim et al.; Ye et al., 2025; Ghasemipour et al.), its large-scale deployment is still limited by the lack of scalable and generalizable reward models. Such models often rely on labor-intensive manual labeling or fragile handcrafted functions (Ma et al., 2024; Christiano et al., 2017; Ouyang et al., 2022; Lee et al., 2026). Since robots operate in unstructured and dynamic environments, this problem is more challenging than in the mathematical/software domains that have automated validation mechanisms (Shao et al., 2024; Guo et al., 2025a; Guan et al.; Pan et al., 2025; Jimenez et al., 2023; Yang et al., 2024; Wang et al., 2024a). Vision-language models (VLMs) (Liu et al., 2023; Achiam et al., 2023; Chen et al., 2024b; Bai et al., 2025) offer a solution, with the RoboReward (Lee et al., 2026) framework serving as a typical example. This framework uses VLMs to assign task-progress rewards without per-task reward engineering, and through a consistent 1–5 scale discrete scoring, it trains Qwen (Bai et al., 2025) VLM reward models that outperform larger closed-source VLMs.

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.

However, RoboReward is limited along two key dimensions that we address in this paper. **(1) Narrow single-paradigm supervision.** RoboReward conducts training and evaluation via per-trajectory scoring on a 1-to-5 discrete rating scale. This single-formulation choice leaves two independent capability gaps in the reward signal. **Pointwise vs. pairwise.** RoboReward's outputs only take the form of pointwise scalar scores, while pairwise preference formats are widely adopted in modern reinforcement learning pipelines, including DPO (Rafailov et al., 2023), RLHF (Ouyang et al., 2022) reward modeling, Bradley–Terry (Bradley & Terry, 1952) and Elo ranking, as well as RLAIF-style (Lee et al., 2023) preference optimization. Pairwise comparisons better conform to human annotation practices. In addition to public benchmarks like AlpacaEval (Li et al., 2023) and Chatbot Arena (Chiang et al., 2024), our internal results presented in Figure 1 also confirm the superior alignment of pairwise evaluation with human annotations. A pointwise-only reward model must manufacture pairwise preferences by post-hoc score processing. However, as shown in Figure 1, real pairwise preferences conducted by humans/reward models often conflict with scalar score rankings. **Behavioral vs. grounded understanding.** Integer task-progress rates merely measure the robot's execution performance, yet never directly verify whether the VLM comprehends the scene on which it reasons. Progress-only task completion scoring is inherently vulnerable to reward hacking. This metric only quantifies the robot's final task execution outcome, with no mechanism to validate whether the underlying vision-language model has true grounded understanding of the target scene. This critical blind spot enables models to achieve high scores via spurious correlations instead of semantically correct task reasoning, as shown in Appendix 3, a hidden failure mode completely undetectable in outcome-only evaluation. **(2) Uncontrolled Cross-Paradigm Training Noise from Supervision Inconsistency.** A straightforward fix for the first limitation is to add pairwise comparison and grounded-QA supervision to existing completion-focused score supervision for joint training. Pointwise scoring labels and pairwise ranking labels, even when deployed exclusively to evaluate task completion, exhibit the fundamental inconsistency revealed in TrustJudge(Wang et al., 2026), and as a result, regularly produce conflicting judgments on the same sample: a sample earning a high completion score in pointwise assessment may be marked as the losing candidate in direct pairwise comparison. Critically, TrustJudge mitigates cross-paradigm inconsistency via probabilistic aggregation to recalibrate raw scoring outputs. This approach can only alleviate inherent cross-paradigm inconsistency rather than eradicating it entirely. If left unaddressed in the training corpus, these conflicting signals generate opposing training signals during training, thus preventing the reward model from converging to a stable, self-consistent judgment function.

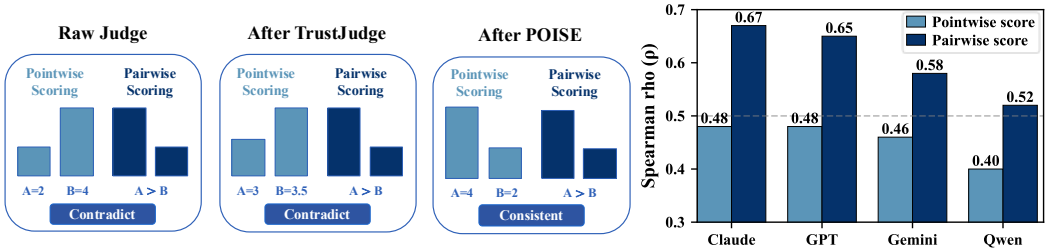


Figure 1: Calibration of score labels using POISE with pair preference signals corrects inconsistencies. Spearman's $\rho > 0.5$ indicates strong positive correlation between LLM and human expert rankings. Pairwise scores align better with human preferences than pointwise scores. Gemini, GPT, Claude, and Qwen denote Gemini 3.1 Pro, GPT-5-mini, Claude Opus 4.6, and Qwen3-VL-32B respectively.

These limitations are closely coupled and cannot be treated separately. Deriving pairwise preferences from score-only reward outputs naturally gives rise to cross-paradigm contradictions, and introducing pairwise supervision further brings cross-paradigm noise that must be regulated to enable robust multi-paradigm reward modeling. To address both limitations, we propose **TrustRoboReward**, a multi-paradigm reward modeling framework consisting of two complementary components. **Multi-paradigm supervision** (§3.1) queries a teacher to construct four complementary supervision paradigms—trajectory progress scoring (Score-A), video-QA answer quality scoring (Score-B), and their pairwise counterparts (Pair-A and Pair-B)—on a shared set of robot video samples, simultaneously closing the pointwise-versus-pairwise gap and the behavioral-versus-grounded gap. **POISE** (Preference-Ordered Isotonic Score Editing; §3.2) then eliminates the cross-paradigm training noise that multi-paradigm supervision inevitably introduces: it treats pairwise labels as a target partial order and projects raw pointwise scores onto the monotone cone of that order via the Pool-Adjacent-

Violators Algorithm, provably reducing training-corpus score-pair reversal conflicts to exactly 0%, which is a training-time guarantee that TrustJudge's probabilistic aggregation cannot provide.

We evaluate TrustRoboReward on a comprehensive multi-paradigm benchmark spanning all four supervision paradigms. With a Qwen3-VL-4B backbone, our model achieves an overall reward score of 77.96%, nearly matching the proprietary GPT-5-mini(Singh et al., 2025) teacher (78.09%, a gap of only 0.13%) while outperforming the strongest open-source RoboReward-4B baseline by 10.13%. On test-time score-pair consistency, our model reaches 71.90%, exceeding both RoboReward-4B (57.26%) and GPT-5-mini (68.09%). Combining our model with TrustJudge at inference time further lifts the overall score to 78.57%, surpassing the proprietary teacher. We additionally validate TrustRoboReward as a downstream reward signal for embodied policy optimization on PAIBench-G(Zhou et al., 2025), where it consistently outperforms RoboReward in guiding robot manipulation.

In conclusion, we contribute: **(i)** a four-paradigm supervision corpus (Score-A/B, Pair-A/B) that simultaneously closes the pointwise-versus-pairwise and behavioral-versus-grounded capability gaps; **(ii)** POISE, a linear-time isotonic denoising algorithm that provably reduces training-corpus score-pair reversal conflicts to exactly 0% and monotonically improves score quality (Theorem 1); **(iii)** an open-weight Qwen3-VL-4B reward model that nearly matches GPT-5-mini (gap 0.13%) and surpasses it with TrustJudge aggregation.

2 Related Work

VLMs as reward judges. VLMs such as LLaVA (Liu et al., 2023; Li et al.) and Qwen3-VL (Bai et al., 2025) have enabled the VLM-as-a-Judge paradigm (Zheng et al., 2023; Lee et al., 2024; Chen et al., 2024a; Xiong et al., 2025), in which a strong generalist model scores or ranks candidate outputs in lieu of human annotators. A growing body of work casts such judges as *generative* reward models that emit chain-of-thought rationales together with scalar verdicts (Mahan et al., 2024; Cao et al., 2024), and as zero-shot visual reward models for downstream RL (Rocamonde et al., 2024; Wang et al., 2024b; Baumli et al., 2023) or video generation (Xu et al., 2026; Liu et al., a). Dedicated reward-model benchmarks such as RewardBench (Lambert et al., 2024; Yasunaga et al., 2025; Li et al., 2025) systematically evaluate such judges and reveal persistent reliability gaps. In robotics specifically, RoboReward (Lee et al., 2026) distills a proprietary teacher into Qwen-based judges emitting a 1-to-5 trajectory-progress score. We follow this distillation recipe but extend supervision beyond single-paradigm pointwise scoring to pairwise preferences and grounded video-QA, and explicitly resolve the cross-paradigm inconsistency that naive multi-paradigm distillation introduces.

Preference learning and score–comparison consistency. Modern alignment pipelines consume *pairwise* preferences: RLHF (Ouyang et al., 2022) and DPO (Rafailov et al., 2023; Meng et al., 2024) train reward models under Bradley–Terry (Bradley & Terry, 1952) assumptions, and pairwise judgments align better with humans than Likert scoring (Zheng et al., 2023; Chen et al., 2024a; Liu et al., b)—a trend Figure 1 also confirms for embodied video. The recent shift toward AI-generated preference labels (RLAIF) (Lee et al., 2023; Bai et al., 2022) and self-improving judges (Mahan et al., 2024; Wu et al., 2025) further amplifies the value of *consistent* preference signals at scale. When the same judge produces both pointwise and pairwise labels, however, the two channels frequently disagree: TrustJudge (Wang et al., 2026) quantifies this score–comparison inconsistency and smooths it at *inference* via probabilistic aggregation. We address the same diagnosis from the opposite end, treating pairwise labels as a partial-order constraint on the *training* corpus and projecting pointwise scores onto its monotone cone via the Pool-Adjacent-Violators algorithm for isotonic regression (Ayer et al., 1955). This training-time correction is provably consistent ($\eta_{\text{cross}}{=}0$) and complements TrustJudge at inference (§3).

3 Method

We address the two limitations raised in Section 1 with two complementary components. To close the narrow single-paradigm gap (limitation 1), *multi-paradigm supervision* (§3.1) queries a teacher to construct supervision under four paradigms—two pointwise paradigms, namely trajectory progress scoring (Score-A) and video-QA answer quality scoring (Score-B), and two pairwise paradigms, namely their pairwise preference forms for dual-video preference (Pair-A) and dual-answer preference (Pair-B)—on a shared set of RoboReward video samples. To eliminate the cross-paradigm training

noise this supervision inevitably introduces (limitation 2), POISE (*Preference-Ordered Isotonic Score Editing*; §3.2) reads the pairwise labels as a target partial order, projects the raw pointwise scores onto the monotone cone of that order by isotonic correction with the Pool-Adjacent-Violators Algorithm (PAVA) (Ayer et al., 1955), and feeds the resulting self-consistent corpus into multi-paradigm training.

3.1 MULTI-PARADIGM SUPERVISION

To overcome RoboReward's narrow single-paradigm supervision, we distill four complementary reward paradigms from the teacher query onto a shared item pool, together spanning both capability axes identified in Section 1. Score-A asks the teacher to rate, on a 1-to-5 scale, a single trajectory's task-progress under a given prompt, matching RoboReward's original scoring protocol. Score-B rates video-QA candidate answers on a consistent scale, prompting VLMs to recognize objects, actions and underlying states rather than relying on surface-level cues. Pair-A compares two trajectories on the same task and emits a label in $\{\text{win}, \text{lose}, \text{tie}\}$, providing the pairwise format that DPO and RLHF reward modeling consume natively. Pair-B compares two candidate answers for the video-QA instance with a shared label vocabulary.

More crucially, the four paradigms share the same set of RoboReward video samples, so their labels can be compared directly on the same samples. We organize samples into groups by task context: each group consists of candidate trajectories sharing a common task instruction (Score-A / Pair-A) or candidate answers sharing a common video (Score-B / Pair-B). All supervision is defined within a group and never across groups, so every pairwise sample aligns with the score labels of its two endpoints. And these two endpoints have shared-grouping property that makes §3.2's cross-paradigm consistency constraint well-defined.

All four paradigms use a uniform Point-then-Explain output format (score or comparison token first, then rationale) and are consumed in a unified two-stage training pipeline (SFT(Wei et al.) then RL) under matched training configurations, so label-variant comparisons (raw vs corrected) isolate the label effect rather than differences in training recipe. The full prompt specification is provided in Appendix F, including the shared system prompt in Appendix F.1, the Score-A template in Appendix F.2, the Score-B template in Appendix F.3, the Pair-A template in Appendix F.4, and the Pair-B template in Appendix F.5.

Multi-paradigm supervision therefore resolves limitation 1 of Section 1: supervision under four complementary paradigms on the same underlying video samples simultaneously closes the pointwise-versus-pairwise and behavioral-versus-grounded capability gaps that RoboReward lefts open.

3.2 POISE: PREFERENCE-ORDERED ISOTONIC SCORE EDITING

The pairwise annotations and pointwise scores are often inconsistent on the same items. To address this mismatch, we propose POISE (Preference-Ordered Isotonic Score Editing) to recalibrate pointwise scores toward weak-order consistency, i.e., no pointwise score ranking contradicts the pairwise preference direction. We formalize the cross-paradigm noise, show why no label-local denoiser can reduce it, derive the partial-order isotonic correction, and give pseudocode with a worked example.

Cross-paradigm noise rate. We define the cross-paradigm noise rate η_{cross} on a single group as the fraction of jointly-labeled pairs whose score-implied ordering disagrees with the pairwise label. Fix a group with m items, score labels $s_i \in \{1, \ldots, 5\}$, and pairwise labels $v_{ij} \in \{\text{win}, \text{lose}, \text{tie}\}$ (item i wins, loses, or ties with j); the same problem applies once per axis (Score-A+Pair-A and Score-B+Pair-B). Formally, $\eta_{\text{cross}} = \Pr_{(i,j)}\big[(v_{ij}{=}\text{win} \wedge s_i{<}s_j) \vee (v_{ij}{=}\text{lose} \wedge s_i{>}s_j)\big]$.

Because η_{cross} couples $\{s_i\}$ and $\{v_{ij}\}$, label-local strategies fundamentally fail to alleviate this inconsistency. This category includes TrustJudge(Wang et al., 2026), which computes expected scores from probabilistic teacher outputs, and majority voting, which identifies the mode among predictions. Neither approach can align the derived scores with pairwise decisions on the same items.

Partial-order isotonic correction. POISE fixes the pairwise labels and rewrites the scores under a partial-order constraint. Declaring $i \prec j$ when $v_{ij} = \text{lose}$ (and tie as equivalence) turns the pairwise labels into a partial order, and we seek scores close to $\{s_i\}$ that are monotone along this order. The only precondition is that $\{v_{ij}\}$ induces a valid partial order; since RoboReward groups contain at most

Algorithm 1 POISE (Preference-Ordered Isotonic Score Editing).
Example (m=4): raw scores $(s_1, s_2, s_3, s_4) = (2, 1, 5, 3)$; pairwise labels $v_{24} = v_{41} = v_{13} = \text{lose}$ (the remaining labels follow by transitivity), yielding the score chain ② $\prec$ ④ $\prec$ ① $\prec$ ③(worst to best).

Require: per-group raw score label $s_1, \ldots, s_m \in \{1, \ldots, 5\}$; per-group pairwise label $\{v_{ij}\}_{i,j=1}^{m} \in \{\text{win}, \text{lose}, \text{tie}\}$ inducing a valid partial order

Ensure: corrected score label $\hat{s}_1, \ldots, \hat{s}_m \in \{1, \ldots, 5\}$ ($\eta_{\text{cross}} = 0$ reversal-conflict on the training corpus)

$(r_1, \ldots, r_m) \leftarrow$ REINDEXBYPAIRWISEORDER$(\{v_{ij}\}, s_1, \ldots, s_m)$ ▷ $(r_1, \ldots, r_m)$: chain order along v from worst to best; r_i is the original score index at chain position i

// Example: v induces chain order $(r_1, r_2, r_3, r_4) = (2, 4, 1, 3)$, so $(s_{r_1}, s_{r_2}, s_{r_3}, s_{r_4}) = (s_2, s_4, s_1, s_3) = (1, 3, 2, 5)$.

$(\hat{s}'_1, \ldots, \hat{s}'_m) \leftarrow \arg\min_{s' \in \mathbb{R}^m} \sum_{i=1}^{m} \big(s'_i - s_{r_i}\big)^2$ s.t. $s'_1 \le s'_2 \le \ldots \le s'_m$ ▷ minimize score change subject to chain-monotonicity; $\hat{s}'_i$ may be non-integer. Detailed implementation is given by Eq. (2).

// Example: PAV pools chain positions $2, 3$ to mean 2.5; $(\hat{s}'_1, \hat{s}'_2, \hat{s}'_3, \hat{s}'_4) = (1, 2.5, 2.5, 5)$.

// each $\hat{s}'_i$ is the constrained optimum corresponding to the score at chain position i, i.e., $(\hat{s}'_1, \hat{s}'_2, \hat{s}'_3, \hat{s}'_4)$ corresponds to (s_2, s_4, s_1, s_3)

for $i = 1, \ldots, m$ **do**

  $\hat{s}_{r_i} \leftarrow \max\big(1, \min(5, \lfloor \hat{s}'_i + \frac{1}{2} \rfloor)\big)$ ▷ round-half-up, then clip back to $\{1, \ldots, 5\}$

end for

// Example: round-half-up writes $(\hat{s}_2, \hat{s}_4, \hat{s}_1, \hat{s}_3) = (1, 3, 3, 5)$ in chain order from worst to best.

return $(\hat{s}_1, \ldots, \hat{s}_m)$

// Example: original scores are $(s_1, s_2, s_3, s_4) = (2, 1, 5, 3)$; corrected scores become $(\hat{s}_1, \hat{s}_2, \hat{s}_3, \hat{s}_4) = (3, 1, 5, 3)$ in the original score order; the corrected scores satisfy $\hat{s}_2 \le \hat{s}_4 \le \hat{s}_1 \le \hat{s}_3$.

three items, cycles occur in $< 1\%$ of groups and are fixed by expert re-annotation. Further discussion on limitations regarding pairwise label quality and noise robustness is provided in Appendix A.

With a valid $\{v_{ij}\}$ in hand, we construct the chain order $r = (r_1, \ldots, r_m)$, i.e., a linear extension of the induced partial order, where r_i is the original score index at chain position i (worst to best). We then solve chain isotonic regression. Note that this linear extension orders incomparable items only for computational tractability, and the squared-error objective minimizes unintended score shifts from the original values. Any assignment monotone along this chain is monotone under the original partial order, so this is a conservative sufficient constraint for eliminating reversal conflicts:

$$(\hat{s}'_1, \ldots, \hat{s}'_m) \;=\; \arg\min_{s' \in \mathbb{R}^m} \sum_{i=1}^{m} \big(s'_i - s_{r_i}\big)^2 \quad \text{s.t.} \quad s'_1 \le s'_2 \le \cdots \le s'_m, \tag{1}$$

which the Pool-Adjacent-Violators iteration (Ayer et al., 1955) solves exactly in time complexity of $O(m)$. The solution partitions $\{1, \ldots, m\}$ into K consecutive blocks $B_1, \ldots, B_K$ along the chain. Each block pools a run of chain positions whose raw scores would otherwise violate the non-decreasing constraint of (1), and all positions in the block share a single corrected score equal to the mean of the pooled raw scores. Corrected scores rise between adjacent blocks, so each block is a maximal contiguous segment sharing one corrected value:

$$\hat{s}'_i \;=\; \mu_k \;\equiv\; \frac{1}{|B_k|} \sum_{j \in B_k} s_{r_j}, \qquad i \in B_k. \tag{2}$$

With strictly increasing block means $\mu_1 < \mu_2 < \cdots < \mu_K$, we apply a single left-to-right pass. Each chain position is initialized as a single block. Whenever the newly added block has a smaller mean than its predecessor, we merge the two blocks and replace them with their weighted-average block mean until adjacent block means are nondecreasing. The real-valued outputs are then clipped and mapped back to the integer score range $\{1, \ldots, 5\}$, and written to item r_i by $\hat{s}_{r_i} = \max\big(1, \min(5, \lfloor \hat{s}'_i + \frac{1}{2} \rfloor)\big)$, preserving the induced chain order and hence the partial order.

Algorithm 1 instantiates POISE as a per-group linear-time step. The procedure runs independently across groups and twice per dataset(Score-A+Pair-A and Score-B+Pair-B).

Data splits. POISE induces six splits for downstream training. $\mathcal{P}_1$ and $\mathcal{S}_1$ are the raw pairwise and score splits from §3.1. We then partition them as $\mathcal{P}_1 = \mathcal{P}_2 \cup \mathcal{P}_3$ and $\mathcal{S}_1 = \mathcal{S}_2 \cup \mathcal{S}_3$, where the unions are disjoint. The two sides are produced by different mechanisms: **(score side)** $\mathcal{S}_2, \mathcal{S}_3$ come from POISE score editing (pointwise labels unchanged in $\mathcal{S}_2$ and rewritten in $\mathcal{S}_3$); **(pairwise side)** $\mathcal{P}_2, \mathcal{P}_3$ come from pairwise-order repair (cycle/transitivity conflicts corrected by expert re-annotation, with unchanged-vs-rewritten labels split into $\mathcal{P}_2$ vs. $\mathcal{P}_3$). Accordingly, $\mathcal{S}_2, \mathcal{P}_2$ are SFT-admissible (teacher rationale still valid), while $\mathcal{S}_3, \mathcal{P}_3$ are RL-only residuals (labels/scores rewritten, so the original rationale may be invalidated).

Our default deployed recipe (**Ours**) cleans only the score side ($\mathcal{S}_1 \to \mathcal{S}_2, \mathcal{S}_3$) and keeps $\mathcal{P}_1$ intact, because pairwise cycle/transitivity conflicts affect fewer than 1% of groups and are already repaired in pairwise-order preprocessing; detailed routing definitions are listed in Appendix C.

Theoretical guarantee. We now provide a formal justification for the effectiveness of POISE. Under pairwise-order consistency, POISE's isotonic projection monotonically improves score quality.The detailed proof is provided in Appendix B. The main conclusion is stated below.

Theorem 1 (Isotonic Alignment Improves Score Quality)**.** *Let $g \in \{1, \dots, 5\}^m$ be the ground-truth golden scores and $s \in \{1, \dots, 5\}^m$ the raw noisy scores for a group of m items. Let $\preceq$ be the partial order induced by pairwise labels, and assume it is order-consistent with the golden scores, namely $i \preceq j$ implies $g_i \le g_j$. Let $\hat{s}' \in \mathbb{R}^m$ denote the real-valued chain-isotonic regression of s on a linear extension r of $\preceq$, solved by PAVA* before *quantization. Then*

$$\|\hat{s}' - g\|_2 \;\le\; \|s - g\|_2. \tag{3}$$

Furthermore, if s violates the order $\preceq$ on at least one comparable pair, the improvement is strict in the squared sense:

$$\|\hat{s}' - g\|_2^2 \;\le\; \|s - g\|_2^2 - \|\hat{s}' - s\|_2^2 \;<\; \|s - g\|_2^2. \tag{4}$$

POISE therefore resolves limitation 2 of Section 1: by construction, the training-corpus reversal-conflict rate $\eta_{\text{cross}} = 0$ after one $O(m)$ sweep per group with no additional teacher queries; together with the quality guarantee in Theorem 1, this shows that POISE removes cross-paradigm reversal inconsistencies while preserving (indeed improving) score quality. Downstream SFT/RL training remains unchanged from the Vanilla-SFT baseline, so any improvement over Vanilla-SFT isolates the effect of our cross-paradigm correction.

4 Experiments

We evaluate our approach in five stages. First, we detail the experimental setup including datasets, baselines, and evaluation metrics (§4.1). Next, we present main results comparing our models against proprietary and open-source baselines (§4.2). We then conduct ablations on data-routing, $\mathcal{S}_3$-RL contribution, and TrustJudge realignment mechanisms. Fourth, we validate our reward models through a human correlation study. Finally, we demonstrate our reward model's superiority in real-world embodied AI scenarios, showing it provides winning guidance for robot policy optimization. Additionally, supplementary experiments are provided in Appendix D, including a detailed performance heatmap across all test sub-datasets in Appendix D.1.

4.1 Preliminaries

Datasets and Splits. We conduct our experiments using a total of $181{,}692$ distillation training samples and a held-out gold set of $7{,}030$ human-annotated test samples. The construction of specific tasks (Score-A/B and Pair-A/B) from the RoboReward dataset is detailed in §3. Following POISE pipeline as described in §3, the raw pointwise ($\mathcal{S}_1$) and pairwise ($\mathcal{P}_1$) samples are processed into cleaned SFT-admissible subsets ($\mathcal{S}_2, \mathcal{P}_2$) and RL-only residual subsets ($\mathcal{S}_3, \mathcal{P}_3$). For detailed statistical distributions of all training and evaluation splits, please refer to Appendix C.

Models and Baselines. We instantiate our reward models using **Qwen3-VL-4B** and **Qwen3-VL-8B** backbones. We evaluate against three baselines: (i) **Raw-distillation SFT** ($\mathcal{S}_1 \cup \mathcal{P}_1, -$), representing the same backbones trained on uncleaned multi-paradigm distillation data; (ii) **RoboReward** (Lee et al., 2026), the dominant prior recipe; and (iii) **GPT-5-mini**, which serves as a soft upper bound for the recoverable performance under our distillation setup. We also include the un-tuned Qwen3-VL backbones as a reference floor to quantify the absolute headroom. Implementation details are provided

Table 1: Main comparison on the four sub-tasks. "–" denotes that the corresponding stage is skipped, and all trained rows use 3 epochs for each stage that is run.

Model	Recipe	SFT data	RL data	Overall	scoreAB	pairAB	Score-A	Score-B	Pair-A	Pair-B	Cons.
Proprietary teacher (upper-bound reference)											
GPT-5-mini	teacher	–	–	78.09	74.52	81.65	75.05	73.98	82.44	80.89	68.09
Ours											
Qwen3-VL-4B	Ours	$\mathcal{S}_2 \cup \mathcal{P}_1$	$\mathcal{S}_3$	**77.96**	74.49	**81.43**	75.98	73.00	**83.18**	79.74	71.90
Qwen3-VL-8B	Ours	$\mathcal{S}_2 \cup \mathcal{P}_1$	$\mathcal{S}_3$	77.46	**74.51**	80.41	**75.59**	**73.44**	80.80	**80.03**	**73.58**
Baselines											
Qwen3-VL-8B	raw-distillation SFT	$\mathcal{S}_1 \cup \mathcal{P}_1$	–	77.21	73.79	80.63	73.70	73.87	83.04	78.30	63.66
Qwen3-VL-4B	raw-distillation SFT	$\mathcal{S}_1 \cup \mathcal{P}_1$	–	77.09	73.18	80.99	73.15	73.20	82.44	79.60	64.81
Qwen3-VL-8B	RoboReward	–	–	69.82	74.28	65.35	77.95	70.61	60.86	69.68	56.14
Qwen3-VL-4B	RoboReward	–	–	67.83	72.58	63.08	74.13	71.04	58.33	67.67	57.26
Reference floor (no fine-tune)											
Qwen3-VL-4B	–	–	–	62.06	66.31	57.82	65.96	66.65	55.80	59.77	62.20
Qwen3-VL-8B	–	–	–	60.47	65.47	55.48	71.80	59.13	52.83	58.05	54.87

in Appendix E, including detailed hyperparameters, training specifics, and compute resource usage in Appendix E.1.

Metrics. We index the four sub-tasks by $t \in \{\text{SCORE-A}, \text{SCORE-B}, \text{PAIR-A}, \text{PAIR-B}\}$, where A denotes the trajectory-progress modality and B denotes the video-QA modality. The performance is evaluated using the following metrics:

- **Pointwise Scores (SCORE-A and SCORE-B):** Let $\mathcal{D}_t$ be the pointwise test set for $t \in \{\text{SCORE-A}, \text{SCORE-B}\}$. For each example $(x_i, s_i) \in \mathcal{D}_t$, x_i denotes the input context, $s_i \in \{1, \dots, 5\}$ is the integer ground-truth rating, and $\hat{s}_i \in \{1, \dots, 5\}$ represents the predicted rating from the reward model. We normalize by the maximal possible error on the 1 to 5 scale (which is 4) and scale the results by 100 so that the scores lie in $[0, 100]$: $\textbf{SCORE-}t = \mathbf{100} \times \left(\mathbf{1} - \frac{\mathbf{1}}{\mathbf{4}\,|\boldsymbol{\mathcal{D}_t}|} \sum_{(\boldsymbol{x_i}, \boldsymbol{s_i}) \in \boldsymbol{\mathcal{D}_t}} \left|\boldsymbol{\hat{s}_i} - \boldsymbol{s_i}\right|\right), \quad \boldsymbol{t} \in \{\text{SCORE-A}, \text{SCORE-B}\}.$
- **Pairwise Accuracies (PAIR-A and PAIR-B):** Let $\mathcal{D}_t$ be the pairwise test set for $t \in \{\text{PAIR-A}, \text{PAIR-B}\}$, where each example consists of an input x_i, two candidate responses, and a dataset label $z_i \in \{\text{win}, \text{lose}, \text{tie}\}$ indicating whether the first response wins, loses, or ties against the second response. Let $\hat{z}_i$ be the label predicted by the reward model. The accuracy is calculated as: $\textbf{PAIR-}t = \mathbf{100} \times \frac{\mathbf{1}}{|\boldsymbol{\mathcal{D}_t}|} \sum_{\boldsymbol{i=1}}^{|\boldsymbol{\mathcal{D}_t}|} \mathbf{1}\big[\boldsymbol{\hat{z}_i} = \boldsymbol{z_i}\big], \quad \boldsymbol{t} \in \{\text{PAIR-A}, \text{PAIR-B}\}.$
- **Paradigm Aggregates (SCORE_{AB} and PAIR_{AB}):** The two paradigm aggregates are formed by sample-weighted (micro) averaging. We calculate this by summing the sub-task numerators and denominators rather than averaging the sub-task ratios. This ensures that the smaller of the two test sets is not over-weighted when the modalities are unbalanced. Let $M \in \{\text{SCORE}, \text{PAIR}\}$ denote the metric type, and let $\mathcal{D}_t$ be the respective test set for sub-task t. The aggregated metric M_{AB} is computed as: $\boldsymbol{M_{\text{AB}}} = \frac{|\boldsymbol{\mathcal{D}_{M\text{-A}}}| \cdot \boldsymbol{M}\text{-A} + |\boldsymbol{\mathcal{D}_{M\text{-B}}}| \cdot \boldsymbol{M}\text{-B}}{|\boldsymbol{\mathcal{D}_{M\text{-A}}}| + |\boldsymbol{\mathcal{D}_{M\text{-B}}}|}, \quad \boldsymbol{M} \in \{\text{SCORE}, \text{PAIR}\}.$
- **Overall Score:** The **Overall** score is the equally weighted average of the micro-averaged pointwise score (SCORE_{AB}) and the micro-averaged pairwise accuracy (PAIR_{AB}) over both modalities. It is calculated as: $\textbf{OVERALL} = \mathbf{(SCORE_{AB} + PAIR_{AB})/2}$.
- **Cross-Paradigm Self-Consistency** (Cons.)**:** This metric measures whether the ordering implied by the pointwise scores agrees with the direct pairwise judgments on the same comparable pairs. Following TrustJudge, we calculate the score-comparison inconsistency rate, also known as the Conflict Ratio (CR). Let CR_{A} and CR_{B} denote the inconsistency rates on A and B, respectively. We define the overall consistency as one minus their average: $\textbf{Cons.} = \mathbf{100} \times \big(\mathbf{1} - \frac{\mathbf{CR_A + CR_B}}{\mathbf{2}}\big)$.

4.2 EXPERIMENTS

Main comparison. Table 1 reports the head-to-head comparison of our method against baselines on the Qwen3-VL-4B and Qwen3-VL-8B architectures. Overall, **Ours 4B** strictly matches the performance of the proprietary teacher, GPT-5-mini, within a 0.13 points margin. Furthermore, it

Table 2: Effect of the SFT and RL data routing on Qwen3-VL-4B. "–" denotes that the corresponding stage is skipped. Sub-row drops the GRPO stage from **Ours** to isolate the contribution of $\mathcal{S}_3$-RL. All rows use 3 epochs for each stage that is run.

Recipe	SFT data	RL data	Overall	scoreAB	pairAB	Score-A	Score-B	Pair-A	Pair-B	Cons.
Raw distillation	$\mathcal{S}_1 \cup \mathcal{P}_1$	–	77.09	73.18	<u>80.99</u>	73.15	73.20	<u>82.44</u>	<u>79.60</u>	64.81
Fully-cleaned	$\mathcal{S}_2 \cup \mathcal{P}_2$	$\mathcal{S}_3 \cup \mathcal{P}_3$	<u>77.53</u>	**75.02**	80.04	**76.04**	**73.99**	81.10	79.02	<u>69.79</u>
Pairwise-only-cleaned	$\mathcal{S}_1 \cup \mathcal{P}_2$	$\mathcal{P}_3$	75.48	72.17	78.80	72.07	72.26	79.76	77.87	64.94
Ours w/o RL	$\mathcal{S}_2 \cup \mathcal{P}_1$	–	76.59	72.49	80.70	73.04	71.94	82.89	78.59	67.08
Ours	$\mathcal{S}_2 \cup \mathcal{P}_1$	$\mathcal{S}_3$	**77.96**	74.49	**81.43**	<u>75.98</u>	73.00	**83.18**	**79.74**	**71.90**

outperforms every open-weight baseline at both scales and delivers the most substantial gains in cross-paradigm self-consistency. The key findings are as follows:

- **Approaching the performance of proprietary teachers.** Remarkably, Qwen3-VL-4B trained with our methodology reaches an Overall score of 77.96, trailing the massive GPT-5-mini teacher (78.09) by a mere 0.13 points. It even surpasses the teacher on the Pair-A sub-task (83.18 vs. 82.44). This demonstrates that multi-paradigm distillation establishes a robust, highly competitive foundation for absolute task accuracy.

  In summary, multi-paradigm distillation lifts the 4B open-weight student to teacher-level accuracy and narrows the observed 4B-8B gap under our protocol. Human-annotated cycle-breaking removes structural conflicts in distillation data, suggesting reliable vision-language reward modeling depends more on supervision coverage and targeted data correction than on parameter scaling alone.

- **Driving self-consistency via POISE after human-annotated cycle-breaking.** Naively concatenating pointwise ($\mathcal{S}_1$) and pairwise ($\mathcal{P}_1$) datasets introduces conflicting reward signals, which causes outputs to disagree on roughly one-third of all evaluated pairs. By leveraging targeted human annotation to break these logical cycles, we successfully reduce this residual disagreement to approximately one-quarter. This mechanism drives massive improvements in cross-paradigm self-consistency (Cons.), where scores jump from $56.14 \rightarrow 63.66 \rightarrow 73.58$ on the 8B backbone (a $+17.44$ total gain) and from $57.26 \rightarrow 64.81 \rightarrow 71.90$ on the 4B backbone, all while maintaining robust performance without compromising absolute task accuracy.

In summary, multi-paradigm distillation lifts 4B open-weight student to teacher-level accuracy and narrows the performance gap between the 4B and 8B models measured under our evaluation protocol. Human-annotated cycle-breaking further removes structural conflicts in the distillation data, suggesting that reliable vision-language reward modeling depends more on broad supervision coverage and targeted data correction than on parameter scaling alone.

Data-routing ablation. Table 2 evaluates four (SFT, RL) data-routing recipes on Qwen3-VL-4B; their explicit tuple definitions are given in Appendix C. Our optimal configuration, **Ours 4B** achieves a top score of 77.96. This performance surpasses the fully-cleaned (77.53), raw distillation (77.09), and pairwise-only-cleaned (75.48) alternatives. Notably, substituting $\mathcal{P}_1$ with the refined $\mathcal{P}_2$ in the SFT stage slightly degrades performance. This suggests that with pointwise data cleaned to $\mathcal{S}_2$, the self-consistent anchors effectively mitigate residual cycles in $\mathcal{P}_1$, allowing its superior diversity to outweigh the cycle costs. We identify a critical asymmetry: while cleaning the pointwise side ($\mathcal{S}_1 \rightarrow \mathcal{S}_2$) is essential, cleaning the pairwise side ($\mathcal{P}_1 \rightarrow \mathcal{P}_2$) imposes a sub-optimal trade-off between diversity and consistency.

$\mathcal{S}_3$-RL contribution. Ablating the GRPO stage (*Ours w/o RL*) reveals that $\mathcal{S}_3$-RL serves as a targeted lever for score–pair alignment instead of a general accuracy booster. While the RL stage yields a modest 1.37 point gain in Overall accuracy ($76.59 \rightarrow 77.96$), it triggers a substantial $+4.8$ point surge in self-consistency ($\text{Cons.} : 67.08 \rightarrow 71.90$). This disproportionate impact is approximately $3.5\times$ greater on consistency than on accuracy, which validates the design of $\mathcal{S}_3$.

POISE is more effective than training-time TrustJudge. Table 3 shows a clear stage difference. Training-time TrustJudge lowers **Ours 4B** from 77.96 to 77.02 Overall, mainly through pointwise drops (Score-A -3.0, Score-B -0.9). This suggests that training-time TrustJudge may be less effective than POISE, as it samples multiple TrustJudge judgments for each training example, aggregates them into a single corrected label, and uses that label as the SFT target, which may reduce label diversity and weaken the benefits of probability-based aggregation. In contrast, inference-time TrustJudge raises **Ours 4B** to 78.57 Overall, achieves the best Pair-A result (85.27), and also improves RoboReward-4B and raw Qwen3-VL-4B. Its slight Cons. drop after RL/GRPO likely comes from

Table 3: TrustJudge ablation on Qwen3-VL-4B. We compare applying TrustJudge alignment to the training-time labels, to the inference-time outputs, or not at all ("–").

Reward model	TrustJudge	Overall	scoreAB	pairAB	Score-A	Score-B	Pair-A	Pair-B	Cons.
Ours	inference	**78.57**	**74.54**	**82.60**	**76.08**	**73.00**	**85.27**	**80.03**	68.81
Ours	–	77.96	74.49	81.43	75.98	**73.00**	83.18	79.74	**71.90**
Ours	training	77.02	72.54	81.51	72.94	72.14	83.33	79.74	62.02
RoboReward	inference	70.29	73.03	67.54	74.95	71.11	61.16	73.71	65.55
Qwen3-VL-4B (raw)	inference	64.56	66.11	63.01	65.69	66.52	61.31	64.66	60.46

reduced output entropy: a more confident model leaves less useful uncertainty for probability-based aggregation, making TrustJudge less effective and slightly weakening score–pair self-consistency.

Human-annotation correlation.

To evaluate reward-model reliability beyond automatic metrics, we sample a subset of $1{,}000$ random test instances with fine-grained human preference ratings to report the Spearman correlation ρ with human preferences. Figure 2 shows that **Ours 4B** achieves the highest correlation in both formats ($\rho = 0.53$ pointwise, $\rho = 0.69$ pairwise) and is the only 4B-scale model exceeding the $\rho = 0.50$ strong-correlation threshold. Compared to the RoboReward baseline $(0.38, 0.49)$, **Ours 4B** improves by $+0.15$ and $+0.20$, respectively.Across all models, pairwise outputs correlate more strongly with humans than pointwise ones, matching prior findings that pairwise judgments yield higher agreement in Table 1. This larger pairwise gain suggests our cleaning pipeline strengthens the model precisely where traditional single-paradigm supervision is most vulnerable.

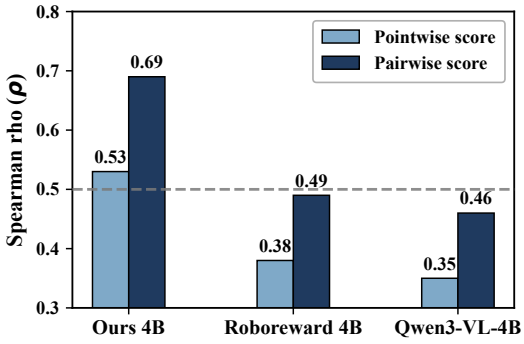


Figure 2: Spearman ρ between reward-model judgments and human annotations, broken down by output format. Dashed line: conventional strong-correlation threshold ρ=0.50. Higher is better.

Reward-Guided Optimization on PAIBench-G Robot Tasks.

We evaluate reward-guided policy optimization on a sampled subset of PAIBench-G robot tasks. The task is image-to-video prediction from an initial robot observation and a manipulation instruction. We conduct a head-to-head comparison between our 4B reward model and RoboReward-4B over 68 paired comparisons, judged by 3 annotators with majority vote. Our method achieves approximately 69% win rate against RoboReward-4B, demonstrating that our reward model provides a better optimization signal for embodied task completion. More parameter details and a qualitative example are provided in Appendix E.2 and Figure 4.

5 Conclusion

We introduced TrustRoboReward, a multi-paradigm reward modeling framework for embodied AI that extends RoboReward-style supervision from pointwise trajectory-progress scoring to four complementary paradigms: trajectory scoring, video-QA scoring, and their pairwise counterparts. The central challenge is that multi-paradigm supervision exposes score–preference conflicts, which create noisy and contradictory training signals. To address this, POISE treats pairwise labels as ordering constraints and edits pointwise scores through isotonic projection, producing a self-consistent corpus with provably zero training-time score–pair reversal conflicts under valid pairwise orders. Experiments show that a Qwen3-VL-4B model trained with TrustRoboReward nearly matches GPT-5-mini, substantially improves over open-weight RoboReward baselines, and achieves stronger score–pair consistency. Inference-time TrustJudge further complements POISE and surpasses the teacher overall. Finally, PAIBench-G optimization shows that our reward model provides a stronger embodied task signal while preserving visual quality, suggesting that consistent multi-paradigm supervision is a practical path toward scalable robot reward modeling. We discuss the limitations of the current framework in Appendix A.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, et al. World simulation with video foundation models for physical ai. *arXiv preprint arXiv:2511.00062*, 2025.

Miriam Ayer, H Daniel Brunk, George M Ewing, William T Reid, and Edward Silverman. An empirical distribution function for sampling with incomplete information. *The annals of mathematical statistics*, pp. 641–647, 1955.

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.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

Kate Baumli, Satinder Baveja, Feryal Behbahani, Harris Chan, Gheorghe Comanici, Sebastian Flennerhag, Maxime Gazeau, Kristian Holsheimer, Dan Horgan, Michael Laskin, et al. Vision-language models as a source of rewards. *arXiv preprint arXiv:2312.09187*, 2023.

Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Maosong Cao, Alexander Lam, Haodong Duan, Hongwei Liu, Songyang Zhang, and Kai Chen. Compassjudger-1: All-in-one judge model helps model evaluation and evolution. *arXiv preprint arXiv:2410.16256*, 2024.

Dongping Chen, Ruoxi Chen, Shilin Zhang, Yinuo Liu, Yaochen Wang, Huichi Zhou, Qihui Zhang, Yao Wan, Pan Zhou, and Lichao Sun. MLLM-as-a-Judge: Assessing multimodal LLM-as-a-Judge with vision-language benchmark. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024a.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24185–24198, 2024b.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E Gonzalez, et al. Chatbot arena: An open platform for evaluating llms by human preference. *arXiv preprint arXiv:2403.04132*, 2024.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Seyed Kamyar Seyed Ghasemipour, Ayzaan Wahid, Jonathan Tompson, Pannag R Sanketi, and Igor Mordatch. Self-improving embodied foundation models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. rstar-math: Small llms can master math reasoning with self-evolved deep thinking. In *Forty-second International Conference on Machine Learning*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv e-prints*, pp. arXiv–2501, 2025a.

Yanjiang Guo, Jianke Zhang, Xiaoyu Chen, Xiang Ji, Yen-Jen Wang, Yucheng Hu, and Jianyu Chen. Improving vision-language-action model with online reinforcement learning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 15665–15672. IEEE, 2025b.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. Swe-bench: Can language models resolve real-world github issues? In *The twelfth international conference on learning representations*, 2023.

Dongyoung Kim, Sumin Park, Huiwon Jang, Jinwoo Shin, Jaehyung Kim, and Younggyo Seo. Robot-r1: Reinforcement learning for enhanced embodied reasoning in robotics. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. RewardBench: Evaluating reward models for language modeling. *arXiv preprint arXiv:2403.13787*, 2024.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. RLAIF vs. RLHF: Scaling reinforcement learning from human feedback with AI feedback. *arXiv preprint arXiv:2309.00267*, 2023.

Seongyun Lee, Seungone Kim, Sue Park, Geewook Kim, and Minjoon Seo. Prometheus-vision: Vision-language model as a judge for fine-grained evaluation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 11286–11315, 2024.

Tony Lee, Andrew Wagenmaker, Karl Pertsch, Percy Liang, Sergey Levine, and Chelsea Finn. Roboreward: General-purpose vision-language reward models for robotics. *arXiv preprint arXiv:2601.00675*, 2026.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *Transactions on Machine Learning Research*.

Lei Li, Yuancheng Wei, Zhihui Xie, Xuqing Yang, Yifan Song, Peiyi Wang, Chenxin An, Tianyu Liu, Sujian Li, Bill Yuchen Lin, et al. Vl-rewardbench: a challenging benchmark for vision-language generative reward models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 24657–24668, 2025.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models, 2023.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Jie Liu, Gongye Liu, Jiajun Liang, Ziyang Yuan, Xiaokun Liu, Mingwu Zheng, Xiele Wu, Qiulin Wang, Menghan Xia, Xintao Wang, et al. Improving video generation with human feedback. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, a.

Yinhong Liu, Han Zhou, Zhijiang Guo, Ehsan Shareghi, Ivan Vulić, Anna Korhonen, and Nigel Collier. Aligning with human judgement: The role of pairwise preference in large language model evaluators. In *First Conference on Language Modeling*, b.

Jianlan Luo, Zheyuan Hu, Charles Xu, You Liang Tan, Jacob Berg, Archit Sharma, Stefan Schaal, Chelsea Finn, Abhishek Gupta, and Sergey Levine. Serl: A software suite for sample-efficient robotic reinforcement learning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 16961–16969. IEEE, 2024.

Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Eureka: Human-level reward design via coding large language models. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

Dakota Mahan, Duy Van Phung, Rafael Rafailov, Chase Blagden, Nathan Lile, Louis Castricato, Jan-Philipp Frazier, Chris Cundy, Sumanth Dahl, and Avi Singh. Generative reward models. *arXiv preprint arXiv:2410.12832*, 2024.

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235, 2024.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Jiayi Pan, Xingyao Wang, Graham Neubig, Navdeep Jaitly, Heng Ji, Alane Suhr, and Yizhe Zhang. Training software engineering agents and verifiers with swe-gym. In *International Conference on Machine Learning*, pp. 47717–47737. PMLR, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Juan Rocamonde, Victoriano Montesinos, Elvis Nava, Ethan Perez, and David Lindner. Vision-language models are zero-shot reward models for reinforcement learning. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Xingyao Wang, Boxuan Li, Yufan Song, Frank F Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, et al. Openhands: An open platform for ai software developers as generalist agents. *arXiv preprint arXiv:2407.16741*, 2024a.

Yidong Wang, Yunze Song, Tingyuan Zhu, Xuanwang Zhang, Zhuohao Yu, Hao Chen, Chiyu Song, Qiufeng Wang, Zhen Wu, Xinyu Dai, Yue Zhang, Cunxiang Wang, Wei Ye, and Shikun Zhang. Trustjudge: Inconsistencies of LLM-as-a-judge and how to alleviate them. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=4uPyOCeN6U.

Yufei Wang, Zhanyi Sun, Jesse Zhang, Zhou Xian, Erdem Biyik, David Held, and Zackory Erickson. Rl-vlm-f: reinforcement learning from vision language foundation model feedback. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 51484–51501, 2024b.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Tianhao Wu, Weizhe Yuan, Olga Golovneva, Jing Xu, Yuandong Tian, Jiantao Jiao, Jason E Weston, and Sainbayar Sukhbaatar. Meta-rewarding language models: Self-improving alignment with llm-as-a-meta-judge. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 11548–11565, 2025.

Tianyi Xiong, Xiyao Wang, Dong Guo, Qinghao Ye, Haoqi Fan, Quanquan Gu, Heng Huang, and Chunyuan Li. Llava-critic: Learning to evaluate multimodal models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 13618–13628, 2025.

Jiazheng Xu, Yu Huang, Jiale Cheng, Yuanming Yang, Jiajun Xu, Yuan Wang, Wenbo Duan, Shen Yang, Qunlin Jin, Shurun Li, Jiayan Teng, Zhuoyi Yang, Wendi Zheng, Xiao Liu, Ming Ding, Xiaohan Zhang, Xiaotao Gu, Shiyu Huang, Minlie Huang, Jie Tang, and Yuxiao Dong. VisionReward: Fine-grained multi-dimensional human preference learning for image and video generation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2026.

John Yang, Carlos E Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering. *Advances in Neural Information Processing Systems*, 37:50528–50652, 2024.

Michihiro Yasunaga, Luke Zettlemoyer, and Marjan Ghazvininejad. Multimodal rewardbench: Holistic evaluation of reward models for vision language models. *arXiv preprint arXiv:2502.14191*, 2025.

Weirui Ye, Yunsheng Zhang, Haoyang Weng, Xianfan Gu, Shengjie Wang, Tong Zhang, Mengchen Wang, Pieter Abbeel, and Yang Gao. Reinforcement learning with foundation priors: Let embodied agent efficiently learn on its own. In *Conference on Robot Learning*, pp. 185–208. PMLR, 2025.

Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, et al. Swift: a scalable lightweight infrastructure for fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 29733–29735, 2025.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Fengzhe Zhou, Jiannan Huang, Jialuo Li, Deva Ramanan, and Humphrey Shi. Pai-bench: A comprehensive benchmark for physical ai. *arXiv preprint arXiv:2512.01989*, 2025.

A LIMITATIONS

TrustRoboReward relies on the assumption that pairwise preferences provide a more reliable ordering signal than pointwise scores. This assumption is supported by our human-correlation study and by prior observations that pairwise judgments often align better with human preferences, but it may not hold uniformly across all robot tasks or video-QA settings. When pairwise labels are systematically biased or incorrect, POISE will preserve the biased order and may propagate the error into the edited pointwise scores. Thus, the quality of pairwise supervision remains a central factor in the reliability of the final reward model.

POISE also assumes that the pairwise labels within each group induce a valid partial order. The linear extension may also induce algorithmic ordering between incomparable items; sensitivity to alternative valid extensions is not evaluated in this work. In our dataset, cycles are rare because each RoboReward group contains at most three items, and the remaining cycle or transitivity conflicts are repaired by expert re-annotation. However, cycle repair only addresses structural inconsistency. It cannot eliminate systematic annotation noise, positional bias, or pairwise judgments that are fundamentally misaligned with the intended notion of task quality. Because POISE enforces the observed pairwise order directly as hard constraints, noisy or biased pairwise labels will propagate into the edited pointwise scores. We do not conduct controlled stress-test experiments for pairwise label corruption in this work. However, larger candidate groups may produce more complex and frequent ordering conflicts.

Extending POISE to large-scale noisy preference graphs without manual cycle repair is an important direction for future work. Similarly, video-QA supervision acts as a complementary task-relevant supervision view rather than a direct ground-truth proxy for physical task completion. VLM-based QA judgments may suffer from hallucination of objects or actions, reliance on superficial visual or linguistic shortcuts, and inherent annotation bias. Consequently, a correct answer under the QA rubric does not guarantee an accurate assessment of real-world task completion status. We do not include a VideoQA-only ablation study that isolates the causal contribution of Score-B and Pair-B to final reward quality. Whether video-QA supervision consistently reduces reward hacking or improves grounded scene understanding remains to be validated with independent task-grounded annotations in future work.

Our theoretical guarantee applies to the real-valued isotonic projection before integer quantization and under the assumption that the induced pairwise order is consistent with the golden scores. The final training labels are discrete scores in $\{1, \ldots, 5\}$, so coordinate-wise rounding can introduce a small additional error. Although this effect is bounded and empirically small in our setting, a fully discrete isotonic projection may be preferable when exact discrete-label guarantees are required.

Our evaluation focuses on RoboReward-derived robot video samples, video-QA supervision, and a sampled subset of PAIBench-G reward-guided optimization tasks. These experiments cover diverse embodied settings, but they do not exhaust all manipulation regimes, long-horizon tasks, real-robot deployment conditions, or safety-critical failure modes. In particular, our downstream validation is based on video generation and human preference comparison rather than closed-loop policy execution on physical robots. Future work should evaluate whether the improved reward signal transfers to real robotic control, interactive policy learning, and out-of-distribution environments.

Finally, TrustRoboReward introduces additional data-construction and training cost compared with single-paradigm distillation. Building the four-paradigm corpus requires teacher queries, consistency analysis, and limited expert re-annotation for invalid pairwise orders, and training the final models requires SFT and GRPO(Shao et al., 2024) stages. These additional data-construction and training steps may limit rapid iteration or deployment in resource-constrained settings. Their precise wall-clock and human-cost trade-off is not separately quantified in the current study.

B PROOF OF THEOREM 1

For a group of m items, let $r = (r_1, \ldots, r_m)$ be the chain order (linear extension of $\preceq$) constructed by POISE, indexing items from worst to best. Define the *chain monotone cone*

$$\mathcal{C}_r \;=\; \left\{x \in \mathbb{R}^m : x_{r_1} \le x_{r_2} \le \cdots \le x_{r_m}\right\}. \tag{5}$$

Because r respects $\preceq$ (if $i \preceq j$ then i appears no later than j in r), any vector monotone along r is automatically monotone under $\preceq$. By the order-consistency assumption (A2), the golden scores satisfy $g_{r_1} \leq \cdots \leq g_{r_m}$, hence

$$g \in \mathcal{C}_r. \tag{6}$$

POISE computes the real-valued isotonic regression

$$\hat{s}' \;=\; \arg\min_{x \in \mathcal{C}_r} \|x - s\|_2^2, \tag{7}$$

which is exactly the metric projection of s onto the closed convex cone $\mathcal{C}_r$, denoted $\hat{s}' = P_{\mathcal{C}_r}(s)$.

Lemma 1 (Projection onto Closed Convex Set)**.** *Let $\mathcal{C} \subseteq \mathbb{R}^m$ be a non-empty closed convex set. For any $y \in \mathbb{R}^m$ and any $z \in \mathcal{C}$,*

$$\|P_{\mathcal{C}}(y) - z\|_2^2 \;\leq\; \|y - z\|_2^2 - \|y - P_{\mathcal{C}}(y)\|_2^2. \tag{8}$$

Proof. By the variational characterization of metric projections onto convex sets,

$$\langle y - P_{\mathcal{C}}(y),\, z - P_{\mathcal{C}}(y) \rangle \;\leq\; 0 \qquad \forall z \in \mathcal{C}. \tag{9}$$

Decomposing $y - z = \big(y - P_{\mathcal{C}}(y)\big) + \big(P_{\mathcal{C}}(y) - z\big)$ yields

$$\begin{aligned} \|y - z\|_2^2 &= \|y - P_{\mathcal{C}}(y)\|_2^2 + \|P_{\mathcal{C}}(y) - z\|_2^2 + 2\langle y - P_{\mathcal{C}}(y),\, P_{\mathcal{C}}(y) - z \rangle \\ &= \|y - P_{\mathcal{C}}(y)\|_2^2 + \|P_{\mathcal{C}}(y) - z\|_2^2 - 2\langle y - P_{\mathcal{C}}(y),\, z - P_{\mathcal{C}}(y) \rangle \\ &\geq \|y - P_{\mathcal{C}}(y)\|_2^2 + \|P_{\mathcal{C}}(y) - z\|_2^2, \end{aligned} \tag{10}$$

where the inequality uses (9). Rearranging (10) gives the claim. □

Proof of Theorem 1. Apply Lemma 1 with $y = s$, $z = g$, and $\mathcal{C} = \mathcal{C}_r$. Using (6) and (7) we obtain

$$\|\hat{s}' - g\|_2^2 \;\leq\; \|s - g\|_2^2 - \|\hat{s}' - s\|_2^2 \;\leq\; \|s - g\|_2^2, \tag{11}$$

which proves (4). Taking square roots gives (3).

If s violates $\preceq$ on some comparable pair, then s is not monotone along the chain r, i.e., $s \notin \mathcal{C}_r$. Because the projection is unique and $\hat{s}' \in \mathcal{C}_r$, we have $\hat{s}' \neq s$ and thus $\|\hat{s}' - s\|_2^2 > 0$, yielding the strict inequality. □

Remark 1 (Quantization to Integer Labels)**.** *POISE quantizes $\hat{s}'$ back to $\{1, \ldots, 5\}^m$ by coordinate-wise round-half-up, yielding the final discrete scores $\hat{s}$. While rounding is not a metric projection onto a convex set, the per-coordinate error is bounded by 0.5. Consequently,*

$$\|\hat{s} - g\|_2 \;\leq\; \|\hat{s}' - g\|_2 + \frac{\sqrt{m}}{2}. \tag{12}$$

In our setting $m \leq 3$, so the additional error is at most 0.87. In practice, the reduction $\|s - \hat{s}'\|_2^2$ from isotonic regression (which is positive exactly when cross-paradigm noise exists) overwhelmingly dominates this small quantization overhead. Hence $\hat{s}$ remains substantially closer to g than s in all empirical regimes.

If a fully discrete theoretical guarantee is desired, one may replace round-half-up by an exact projection onto the discrete monotone set $\mathcal{C}_r \cap \{1, \ldots, 5\}^m$; this is computationally trivial for $m \leq 3$ and preserves the validity of Theorem 1 with $\hat{s}$ in place of $\hat{s}'$.

C Detailed statistical distributions of training and evaluation splits

We consider four recipes that exhaust the natural choices of which side(s) of the corpus to clean, each specified by its (SFT data, RL data) tuple over $\mathcal{S}_{1,2,3}$ and $\mathcal{P}_{1,2,3}$:

- $\big(\mathcal{S}_1 \cup \mathcal{P}_1,\; -\big)$ — the raw single-decode distillation corpus is used directly for SFT, with no DAG/PAVA cleaning and no RL stage. SFT $181{,}692$, RL 0.

- $\big(\mathcal{S}_2 \cup \mathcal{P}_2,\ \mathcal{S}_3 \cup \mathcal{P}_3\big)$ — both pointwise and pairwise sides are cleaned; the residual conflict samples on both sides are routed into RL. SFT 173,335 ($\mathcal{S}_2/\mathcal{P}_2$ counts $41{,}682 + 42{,}517$ pointwise plus $43{,}816 + 45{,}320$ pairwise), RL 8,357 ($\mathcal{S}_3/\mathcal{P}_3$ counts $3{,}390 + 2{,}555$ pointwise plus $712 + 1{,}700$ pairwise).
- $\big(\mathcal{S}_2 \cup \mathcal{P}_1,\ \mathcal{S}_3\big)$ — our deployed recipe **Ours**: clean only the pointwise side, keep $\mathcal{P}_1$ intact. SFT 175,747 (pointwise $41{,}682 + 42{,}517$, pairwise $44{,}528 + 47{,}020$), RL 5,945 (pointwise $3{,}390 + 2{,}555$).
- $\big(\mathcal{S}_1 \cup \mathcal{P}_2,\ \mathcal{P}_3\big)$ — the symmetric counterpart of **Ours** that cleans only the pairwise side. SFT 179,280 (pointwise $45{,}072 + 45{,}072$, pairwise $43{,}816 + 45{,}320$), RL 2,412 (pairwise $712 + 1{,}700$).

We additionally report a sub-row that drops the GRPO stage from **Ours**, $\big(\mathcal{S}_2 \cup \mathcal{P}_1,\ -\big)$, to isolate the marginal contribution of $\mathcal{S}_3$-RL. All recipes use Qwen3-VL-4B with 3 epochs of SFT and (where applicable) 3 epochs of GRPO RL.

D SUPPLEMENTARY EXPERIMENTS

D.1 SUBSET-LEVEL GENERALIZATION

Table 4: **Subset-level generalization across RoboReward test subsets.** (a) maps each subset ID to its corresponding RoboReward test subset and sample count. (b) reports Overall and per-subset scores for the nine main-table configurations. Colors in (b) are normalized globally over all displayed values, with warmer colors indicating higher performance.

(a) Subset index.

Column	Subset	n	Column	Subset	n
S01	robo_arena	2000	S13	cmu_play_fusion	236
S02	fractal20220817_data	356	S14	austin_sirius_dataset_converted_externally_to_rlds	228
S03	ucsd_kitchen_dataset_converted_externally_to_rlds	326	S15	taco_play	228
S04	bridge	292	S16	berkeley_rpt_converted_externally_to_rlds	222
S05	berkeley_autolab_ur5	250	S17	iamlab_cmu_pickup_insert_converted_externally_to_rlds	218
S06	jaco_play	250	S18	utokyo_xarm_bimanual_converted_externally_to_rlds	202
S07	roboturk	250	S19	utokyo_pr2_tabletop_manipulation_converted_externally_to_rlds	182
S08	berkeley_fanuc_manipulation	248	S20	berkeley_mvp_converted_externally_to_rlds	178
S09	droid	248	S21	viola	156
S10	tokyo_u_lsmo_converted_externally_to_rlds	248	S22	kaist_nonprehensile_converted_externally_to_rlds	142
S11	ucsd_pick_and_place_dataset_converted_externally_to_rlds	248	S23	dlr_edan_shared_control_converted_externally_to_rlds	80
S12	stanford_hydra_dataset_converted_externally_to_rlds	242			

(b) Subset-level performance heatmap.

Model	Recipe	SFT data	RL data	Overall	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20	S21	S22	S23
Proprietary teacher (upper-bound reference)																											
GPT-5-mini	teacher	–	–	0.781	0.724	0.810	0.819	0.874	0.858	0.735	0.812	0.759	0.643	0.729	0.850	0.800	0.675	0.900	0.653	0.737	0.721	0.738	0.834	0.879	0.784	0.795	0.834
Ours																											
Qwen3-VL-4B	Ours	$\mathcal{S}_2 \cup \mathcal{P}_1$	$\mathcal{S}_3$	0.780	0.724	0.812	0.795	0.887	0.877	0.711	0.733	0.774	0.651	0.711	0.808	0.745	0.696	0.853	0.725	0.775	0.785	0.759	0.778	0.861	0.785	0.844	0.853
Qwen3-VL-8B	Ours	$\mathcal{S}_2 \cup \mathcal{P}_1$	$\mathcal{S}_3$	0.775	0.721	0.799	0.738	0.858	0.843	0.688	0.748	0.728	0.709	0.712	0.895	0.758	0.689	0.839	0.760	0.812	0.759	0.700	0.825	0.844	0.835	0.795	0.857
Baselines																											
Qwen3-VL-8B	raw-distillation SFT	$\mathcal{S}_1 \cup \mathcal{P}_1$	–	0.772	0.719	0.805	0.801	0.882	0.882	0.723	0.710	0.745	0.625	0.751	0.812	0.753	0.694	0.846	0.670	0.750	0.766	0.726	0.810	0.852	0.736	0.840	0.831
Qwen3-VL-4B	raw-distillation SFT	$\mathcal{S}_1 \cup \mathcal{P}_1$	–	0.771	0.713	0.824	0.791	0.877	0.891	0.674	0.752	0.754	0.637	0.644	0.798	0.730	0.665	0.841	0.676	0.756	0.748	0.757	0.814	0.881	0.800	0.823	0.819
Qwen3-VL-8B	RoboReward	–	–	0.698	0.723	0.740	0.755	0.830	0.863	0.713	0.639	0.697	0.708	0.517	0.794	0.546	0.681	0.624	0.640	0.685	0.769	0.573	0.795	0.730	0.731	0.647	0.699
Qwen3-VL-4B	RoboReward	–	–	0.678	0.707	0.759	0.755	0.738	0.772	0.692	0.499	0.731	0.651	0.512	0.846	0.482	0.659	0.702	0.578	0.729	0.658	0.573	0.788	0.653	0.671	0.715	0.789
Reference floor (no fine-tune)																											
Qwen3-VL-4B	–	–	–	0.621	0.685	0.675	0.649	0.638	0.576	0.520	0.507	0.625	0.631	0.600	0.748	0.344	0.638	0.693	0.557	0.645	0.680	0.535	0.670	0.664	0.570	0.552	0.768
Qwen3-VL-8B	–	–	–	0.605	0.677	0.640	0.594	0.641	0.635	0.533	0.560	0.590	0.631	0.456	0.662	0.483	0.646	0.571	0.548	0.630	0.701	0.525	0.743	0.584	0.697	0.543	0.691

Subset-level robustness. Table 4 visualizes the performance of the nine models configurations across 23 RoboReward test subsets, using a single global color scale where warmer colors indicate higher scores. **Ours** exhibits consistently strong performance across heterogeneous robot datasets and is visually close to the proprietary GPT-5-mini teacher on most subsets. In particular, Qwen3-VL-4B with POISE achieves the best open-weight Overall score and remains competitive on both large subsets such as ROBO_ARENA and smaller specialized subsets such as BERKELEY_MVP, KAIST_NONPREHENSILE, and DLR_EDAN. Compared with raw Qwen3-VL backbones and the single-paradigm RoboReward baselines, the POISE-trained models show a much warmer and more stable pattern across columns, suggesting that the gains are not driven by a single dominant subset but reflect broader cross-dataset generalization. The contrast between POISE and raw-distillation SFT further indicates that multi-paradigm distillation provides strong overall coverage, while our consistency-oriented cleaning improves the reliability of the learned reward signal across diverse embodiment and task distributions.

D.2 Qualitative Case Study

Analysis. Figure 3 and Table 5 illustrate representative cases where our model aligns its reward prediction with concrete visual evidence, while the baselines often rely on coarse object or action cues. In SCORE-A, the instruction asks the robot to place the onion into the big pot, but the video shows that the onion remains on the table away from the pot while the robot places an egg on the pot instead. Our model correctly predicts the lowest score (1), whereas RoboReward-4B, RoboReward-8B, and untuned Qwen3-VL-4B/8B overestimate the outcome as 4, 5, 5, and 5, respectively. In SCORE-B, the candidate answer "Pick up the yellow cup" accurately describes the visible robot action, and our model assigns the gold score (5), while the baselines underestimate the answer quality. The two pairwise cases show the same pattern in comparative form: in PAIR-A, our model selects video A because it makes clearer progress toward placing the ranch bottle into the pot; in PAIR-B, it selects candidate B because it is semantically closer to the visible orange and blue chip bag, despite being imperfect. Overall, these examples suggest that our training pipeline improves visual grounding across both pointwise and pairwise reward formats, rather than merely increasing aggregate benchmark scores.

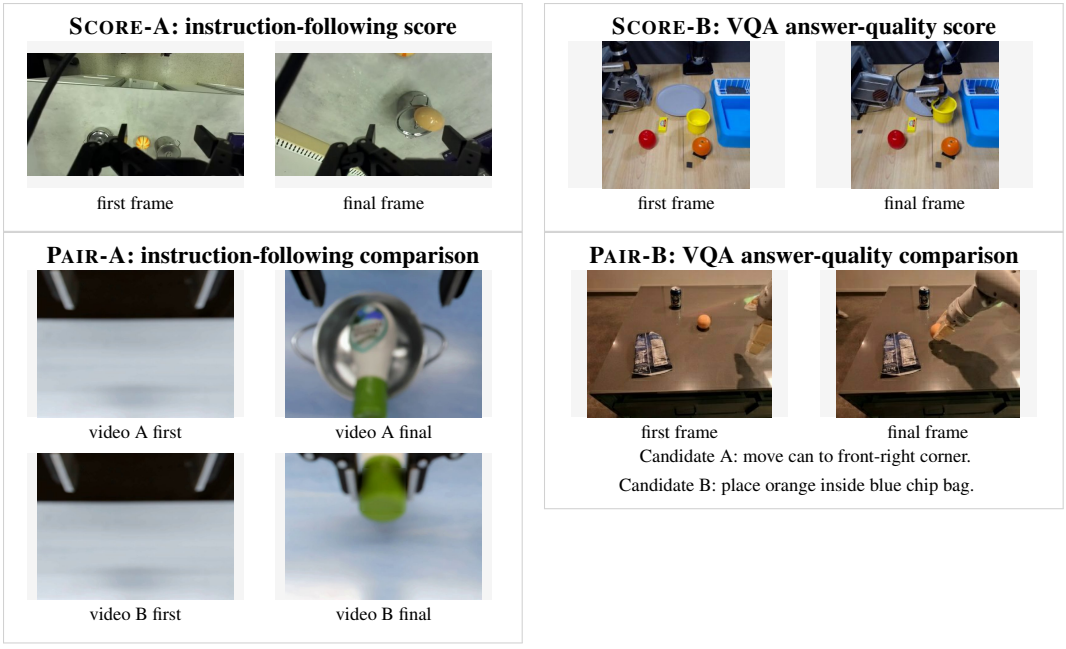


Figure 3: **Qualitative case study across four reward-supervision views.** We show representative examples from pointwise scoring and pairwise comparison tasks.

Table 5: **Model predictions for the qualitative case study.** Green cells match the gold label, while red cells indicate errors.

Case	GT	Ours	RoboReward-4B	RoboReward-8B	Qwen3-VL-4B	Qwen3-VL-8B
SCORE-A	**1**	**1**	4	5	5	5
SCORE-B	**5**	**5**	3	3	1	2
PAIR-A	**A**	**A**	B	C	C	B
PAIR-B	**B**	**B**	C	A	C	C

D.3 Embodied Case Study

Figure 4 provides a qualitative comparison on a representative PAIBench-G robot manipulation task. Given the instruction to pick up the pot and place it on the right side, the policy optimized with RoboReward produces visually plausible motion but fails to preserve the task semantics: the grasp is misaligned and the object placement becomes unstable. In contrast, optimization with our reward model yields a trajectory that first establishes a more accurate grasp and then maintains object stability during placement. This example illustrates the same trend observed in Table 1: our reward

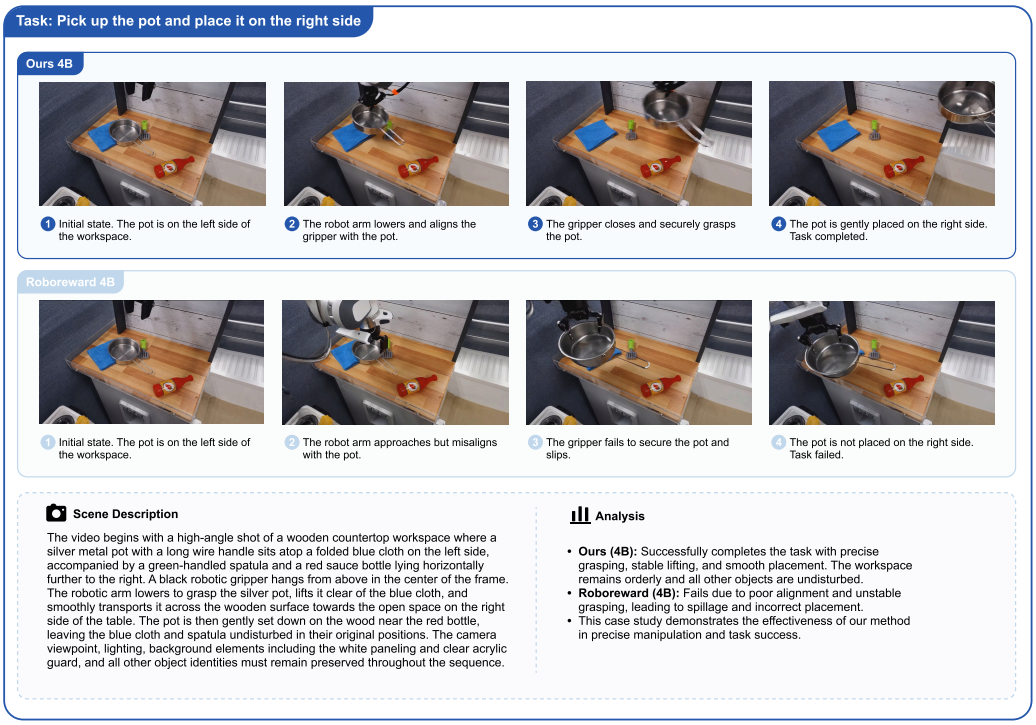


Figure 4: Case study comparison between our method and RoboReward on the task of picking up the pot and placing it on the right side. Our method achieves accurate grasping and stable placement, whereas the RoboReward baseline fails due to reward misalignment and unstable manipulation.

model gives a stronger embodied-task signal, especially for semantic correctness and manipulation stability, while preserving comparable visual quality.

E Implementation Details

E.1 Reward Model and Baseline Training Settings

Table 6: Hyperparameters for Supervised Fine-Tuning (SFT) and Group Relative Policy Optimization (GRPO) stages. Both Qwen3-VL-4B and 8B backbones share identical training configurations unless otherwise specified.

Hyperparameter	**Value**
General & Hardware	
Backbones	Qwen3-VL-4B-Instruct, Qwen3-VL-8B-Instruct
Hardware	8× NVIDIA A800 (80 GB) GPUs
Trainable parameters	Full-parameter (vision tower frozen; multimodal aligner & LLM unfrozen)
Parallelism & Memory	FSDP full-shard, gradient checkpointing, FlashAttention-2
Precision	Mixed precision (GRPO Actor uses float16)
Optimizer	AdamW, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-6}$
Supervised Fine-Tuning (SFT) Stage	
Framework	ms-swift(Zhao et al., 2025)
Training epochs	3 (saving checkpoints at each epoch end)
Training time (approx.)	∼30 hours (4B) / ∼45 hours (8B)
Learning rate	5×10^{-6}
LR schedule	Cosine
Warmup ratio	0.05
Weight decay	0.05
Global batch size	32 (1 sample per device × 4 grad accumulation × 8 GPUs)
Max sequence length	32,768 (overlong samples dropped, not truncated)
Gradient clipping norm	1.0
Reinforcement Learning (GRPO) Stage	
Framework	VeRL(Sheng et al., 2025)
Initial policy	Corresponding SFT checkpoint
Training epochs	3 (saving checkpoints at each epoch end)
Training time (approx.)	∼20 hours (4B) / ∼30 hours (8B)
Rollout engine	vLLM (`gpu_memory_utilization=0.75`, tensor parallel size 1)
Prompt batch size	128
Samples per prompt (n)	8
Effective batch size	1,024 per step
Learning rate	2×10^{-6}
LR schedule	Constant (no warmup)
Max prompt length	14,336
Max response length	4,096
PPO clip ratio	0.2
PPO mini-batch size	16
PPO epochs per rollout	1
KL penalty & Entropy	Disabled (`use_kl_loss=False`, `kl_coef=0`, `entropy_coeff=0`)
Advantage normalization	Intra-group normalization ($\gamma = 1$, $\lambda = 1$)
Gradient clipping norm	1.0
Additional optimizations	Optimizer state CPU offload (specifically for the 8B backbone)

Table 6 lists the full set of hyperparameters and hardware configurations used for instantiating our reward models and baselines. Both the Qwen3-VL-4B and Qwen3-VL-8B backbones share the same hyperparameter setup across the SFT and GRPO training stages, with the only exception being the optimizer state CPU offloading, which is exclusively enabled for the 8B model during GRPO to manage memory constraints.

Compute Resource Usage. All training stages are conducted on a single compute node equipped with 8× NVIDIA A800 (80 GB) GPUs. For the Qwen3-VL-4B backbone, the SFT stage requires approximately 30 hours, while the GRPO stage takes 20 hours, totaling 400 GPU-hours. For the Qwen3-VL-8B backbone, the SFT and GRPO stages require approximately 45 hours and 30 hours respectively, totaling 600 GPU-hours. Overall, the total compute usage for training the final models reported in this work is approximately 125 node-hours (equivalent to 1,000 A800 GPU-hours).

E.2 PAIBench-G Robot Reward-Guided Optimization Settings

Table 7: PAIBench-G robot reward-guided optimization training hyperparameters. Both reward-model variants use the same configuration.

Hyperparameter	Value
Backbone	Cosmos-Predict2.5-2B(Ali et al., 2025)
Algorithm	DiffusionNFT
Benchmark	PAIBench-G robot subset
Reward	RoboReward / ours
Training epochs	50
Prompt groups	32 per epoch
Group size	12
Samples per epoch	384
Batch size	3 per device
Gradient accumulation	4
Sampling steps	10 for training, 36 for evaluation
Guidance scale	4.0 for training, 7.0 for evaluation
Learning rate	6×10^{-5}
Optimizer	AdamW, $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay 10^{-4}
LoRA	rank 32, alpha 64
Advantage clipping	2
Training beta	0.006
Reward beta	1.0
Precision	bf16
Video format	93 frames, 432×768, 16 fps
Seed	42

Table 7 lists the full set of hyperparameters used in the PAIBench-G robot reward-guided optimization experiment. The RoboReward baseline and our method share an identical configuration, differing only in the reward model.

F Prompts

This appendix reproduces, verbatim, the prompts used throughout our pipeline: at distillation time (querying the GPT-5-mini teacher to build the four-paradigm corpus of §3.1), at student training time, and at inference/evaluation time. Keeping a single set of prompt templates across training and inference eliminates a common distribution-shift source and ensures that teacher outputs, student training labels, and benchmark evaluations are all directly comparable. All four paradigms share a single system prompt and differ only in the user template. The placeholder `<video>` marks where the model receives the video frames; `{action_text}`, `{question}`, `{answer_text}`, `{answer_a}`, and `{answer_b}` are filled in at runtime.

F.1 Shared System Prompt

The same system message is prepended to all four paradigms. It pins the teacher to a strict, video-grounded judge persona and enforces the required XML-style answer envelope that our downstream parsers in `verl_common.py` (`parse_score_from_output`, `parse_pair_label_from_output`) expect.

Prompt: Shared System

```
You are a strict, impartial, and consistent video judge for robot task
evaluation.
You must evaluate robot performance based solely on what is visible in
the video.
Be concrete and factual. Do not invent objects or events that are not
visible.
Follow the provided rubric exactly and output ONLY in the required XML
format.
```

F.2 SCORE-A: TRAJECTORY PROGRESS SCORING

Score-A asks the teacher to rate a single trajectory's task progress on a 1-to-5 rubric, matching RoboReward's original scoring protocol. The rubric explicitly operationalizes the 3-vs-4 boundary (partial vs. near completion) in terms of primary object identity and primary spatial relation, which is what makes Score-A labels consistent enough with Pair-A labels for POISE's isotonic correction to be well-defined.

Prompt: Score-A User Template

```
<video>
Task Instruction:
{action_text}

Scoring Rubric (judge only the final state, ignore time limits):
1 - No Success: The final state shows no goal-relevant change for the
task (e.g., the robot handled a completely different object or made no
meaningful motion).
2 - Minimal Progress: The final state shows a small but insufficient
change toward the goal (e.g., the robot moved toward the target but
did not meaningfully interact with it).
3 - Partial Completion: The final state shows good progress toward the
goal but violates more than one requirement or a major requirement
(e.g., the primary spatial relation is incorrect, or the wrong object
identity was used).
4 - Near Completion: The final state is correct in region and intent
but misses a single minor requirement (e.g., the target object is
correctly placed relative to the reference, but an auxiliary
constraint like gripper release, orientation tolerance, or a secondary
support surface is missed).
5 - Perfect Completion: The final state fully satisfies all the
requirements of the task.

Clarification for scoring 3 vs 4 (MINOR vs MAJOR requirements):
- Treat a requirement as MINOR when the primary object identity and
primary spatial relation to the reference object are satisfied, but
an auxiliary constraint is missed.
- Examples of auxiliary (MINOR) constraints: remaining on the same
support surface, holding/releasing the gripper at the end, small
orientation or placement tolerances, or cosmetic positioning details
that do not change the core relation.
- Use score 4 (Near Completion) if all core elements are correct but
one auxiliary constraint is missed.
- Use score 3 (Partial Completion) only when multiple constraints are
missed or when the missed constraint is central to the task identity.

Important:
- Judge ONLY the final state shown in the video, not intermediate
steps.
- Be concrete and factual. Do not invent objects or events not visible
```

```
in the video.
- Pay special attention to the last frame for the final state
assessment.

Return an integer score from 1 to 5 and a brief reason (1-2 sentences)
    .
Output format:
<reason>...</reason>
<answer>Score: [X]</answer>
```

F.3 SCORE-B: VIDEO-QA ANSWER QUALITY SCORING

Score-B rates a candidate answer to a grounding question on the same 1-to-5 scale, forcing the judge to verify objects, actions, and states rather than rely on surface textual cues. Hallucination is explicitly named as the primary failure mode in the rubric.

Prompt: Score-B User Template

```
<video>
Question:
{question}

Candidate Answer:
{answer_text}

Answer Quality Rubric:
1 - Completely incorrect: The answer is entirely wrong, hallucinated,
or not grounded in the video content. It contradicts what is visible
or fabricates objects/events that do not appear.
2 - Slightly relevant: The answer shows some awareness of the video
topic but is mostly wrong, missing key details, or includes
significant hallucinations.
3 - Partially correct: The answer captures some correct elements but
has notable errors, misidentifies important objects or actions, or
omits important points that are clearly visible in the video.
4 - Mostly correct: The answer is largely accurate and grounded in the
video, with only minor omissions, imprecise descriptions, or trivial
inaccuracies.
5 - Completely correct: The answer is fully accurate, well-grounded in
the video, concise, and covers all key objects and actions relevant
to the question.

Important:
- Evaluate the answer strictly based on what is visible in the video.
- Penalize hallucinated content: objects, actions, or details not
present in the video.
- Consider factual accuracy, visual grounding, and coverage of key
details.

Return an integer score from 1 to 5 and a brief reason (1-2 sentences)
    .
Output format:
<reason>...</reason>
<answer>Score: [X]</answer>
```

F.4 PAIR-A: DUAL-TRAJECTORY PREFERENCE

Pair-A compares two robot execution videos on the same task and emits a label in {`[A]`, `[B]`, `[C]`} (A wins / B wins / tie), providing the pairwise format that DPO and Bradley-Terry pipelines consume natively. Two design choices in this prompt are relevant to the main paper:

- The prompt instructs the teacher to analyze *Candidate B first, then Candidate A*, which is our cheap position-bias mitigation at distillation time and complements the symmetric `order_ab`/`order_ba` aggregation in §3.1.
- Tie (`[C]`) is defined restrictively—"both videos genuinely show the same level of task completion"—so that Pair-A labels remain informative partial-order constraints for POISE's isotonic correction (§3.2) rather than collapsing into uninformative ties.

Prompt: Pair-A User Template

```
=== Video A (frames below) ===
<video>
=== Video B (frames below) ===
<video>
The FIRST video corresponds to Candidate A.
The SECOND video corresponds to Candidate B.

Task Instruction:
{action_text}

You are comparing two robot execution videos for the same task.
Choose the video that shows better task completion. Your evaluation
should consider:
- Whether the correct object was manipulated
- Whether the target spatial relation was achieved
- Whether all task requirements were satisfied
- The overall degree of progress toward the goal

The final outcome (what state the task is in at the end) is the
primary criterion. The execution process (efficiency, smoothness,
side effects) serves as a secondary tiebreaker when both candidates
reach similar final states.

IMPORTANT: To avoid order bias, analyze Candidate B FIRST, then
Candidate A.

Step 1: Watch Candidate B fully. Describe what the robot did and what
the final state looks like.
Step 2: Watch Candidate A fully. Describe what the robot did and what
the final state looks like.
Step 3: Compare the two. Which video shows more progress toward
completing the task? If one clearly achieved more, choose it.
Only declare [C] (tie) when both videos genuinely show the same level
of task completion and you cannot identify any meaningful difference.

Be concrete and factual - do not invent objects or events not visible
in the videos. Avoid any position biases and ensure that the order in
which the videos were presented does not influence your judgment.

Output format:
<analysis_B>Candidate B: what the robot did and the final state
</analysis_B>
<analysis_A>Candidate A: what the robot did and the final state
</analysis_A>
<reason>Brief comparison - which achieved more toward the task goal
</reason>
<answer>[A]</answer> or <answer>[B]</answer> or <answer>[C]</answer>
```

F.5 PAIR-B: DUAL-ANSWER PREFERENCE

Pair-B compares two candidate answers to the same grounding question with the same $\{$`[A]`, `[B]`, `[C]`$\}$ label vocabulary as Pair-A. The rubric elevates factual accuracy and absence

of hallucination over coverage or length so that Pair-B agrees, on jointly-labeled items, with Score-B's hallucination-penalizing rubric in §F.3. The same reverse-order instruction as in Pair-A is applied to mitigate order bias.

Prompt: Pair-B User Template

```
<video>
Question:
{question}

Candidate A Answer:
{answer_a}

Candidate B Answer:
{answer_b}

You are comparing two candidate answers to the above question about
the video.
Choose the answer that better addresses the question. Your evaluation
should consider:
- Factual accuracy: does the answer correctly describe what happened
in the video?
- Absence of hallucination: does the answer avoid fabricating objects,
actions, or details not visible?
- Visual grounding: is the answer based on what is actually shown in
the video?
- Coverage: does the answer address the key objects and actions
relevant to the question?

Factual accuracy and absence of hallucination are the primary
criteria. Coverage and level of detail serve as secondary tiebreakers
when both candidates are equally accurate.

Do not let the length of the answers influence your evaluation.
Evaluate strictly based on what is visible in the video.

IMPORTANT: To avoid order bias, analyze Candidate B FIRST, then
Candidate A.

Step 1: Read Candidate B. Assess its factual accuracy and whether it
hallucinates.
Step 2: Read Candidate A. Assess its factual accuracy and whether it
hallucinates.
Step 3: Compare the two. Which answer is more accurate and better
grounded in the video? If one is clearly more accurate, choose it.
Only declare [C] (tie) when both answers are genuinely equal in
quality and you cannot identify any meaningful difference.

Be concrete and factual - do not invent details not visible in the
video. Avoid any position biases and ensure that the order in which
the answers were presented does not influence your judgment.

Output format:
<analysis_B>Candidate B: accuracy assessment and key observations
</analysis_B>
<analysis_A>Candidate A: accuracy assessment and key observations
</analysis_A>
<reason>Brief comparison - which answer is more accurate and complete
</reason>
<answer>[A]</answer> or <answer>[B]</answer> or <answer>[C]</answer>
```

AI USE STATEMENT

During the preparation of this manuscript, we used generative AI tools solely to assist with language polishing, grammar correction, and LaTeX formatting. All AI-assisted outputs were reviewed and verified by the authors. The authors take full responsibility for the accuracy, originality, compliance, and final content of this submission.