NOISERUB: NOISE-CALIBRATED DYNAMIC RUBRIC REWARDS FOR REAL-TIME VISUAL GAMES

Anonymous authors
Paper under double-blind review

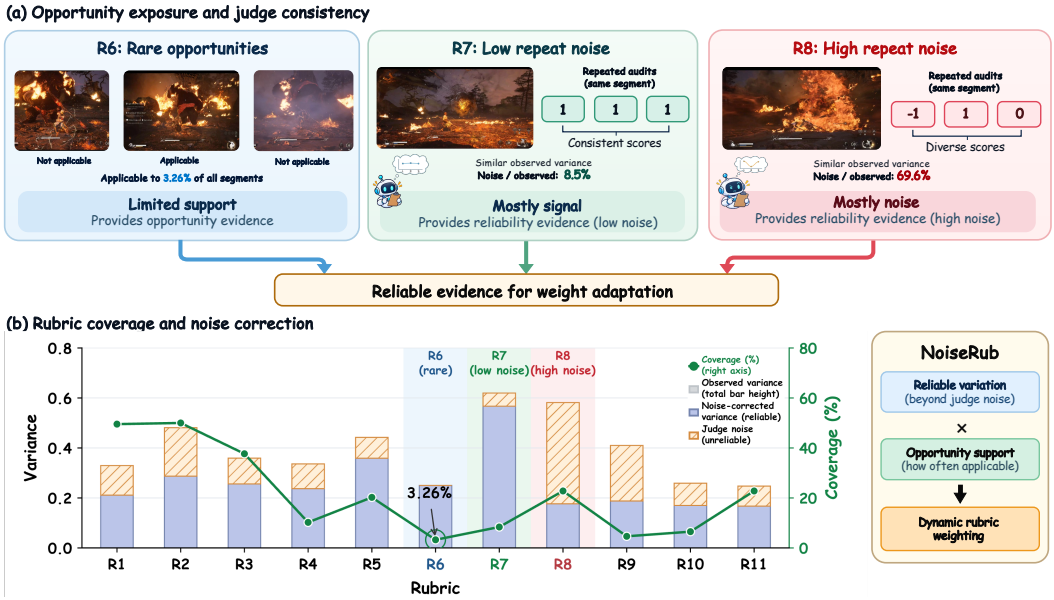


Figure 1: Raw rubric variance can mislead dynamic weighting. R7 and R8 exhibit similar across-segment variance but markedly different repeat-evaluation noise, while R6 is applicable to only 3.26% of segments. NOISERUB calibrates observed variation along two complementary dimensions: repeated audits estimate reliable variation beyond judge noise, while opportunity coverage measures the support for adapting each rubric's weight.

ABSTRACT

Training agents in real-time visual games from screen-only interaction requires dense feedback beyond sparse terminal outcomes. Vision-language judges can provide rubric-based rewards, but dynamically weighting these rewards is challenging because across-segment score variance can reflect behavioral differences, judge inconsistency, or limited opportunities for a rubric to apply. We introduce NOISERUB, a noise-calibrated approach to dynamic rubric rewards. Rather than treating observed score variance as evidence of an informative rubric, NOISERUB asks how much of that variation provides reliable evidence for adapting each rubric's weight and how well that evidence is supported by evaluation opportunities. It therefore treats dynamic weighting as a reliability-estimation problem: judge-induced variation should be discounted, while weight adaptation should reflect how often a rubric can be meaningfully evaluated. Each rubric's reliable variation is estimated by subtracting a within-segment noise floor, obtained from repeated audits, from its across-segment variance; the resulting nonnegative residual is then modulated by opportunity coverage. The resulting weights aggregate rubric scores into segment rewards for online policy optimization. In *Black Myth: Wukong*, NOISERUB achieves 63.5% agreement with human overall preferences, compared with 57.4% for fixed weights and 60.0% for POW3R-Game. Under matched initialization and interaction budgets, it achieves the highest HP reduction on all four evaluated bosses, outperforming POW3R-Game by 4.45–8.04 percentage points. Supplementary videos are available on the project website.

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.

1 INTRODUCTION

Deep reinforcement learning has shown that agents can learn complex control policies directly from visual observations (Mnih et al., 2015; Hafner et al., 2025), yet training rewards often still rely on privileged environment information. For example, DQN acts from game screens in the Arcade Learning Environment (Bellemare et al., 2013), while receiving rewards from game scores, and Auto MC-Reward constructs dense rewards from task descriptions and structured environment observations (Li et al., 2024). Such signals are unavailable when an agent interacts with a game only through the screen, keyboard, and mouse. Observable terminal outcomes such as wins and losses remain sparse over long action sequences, motivating process rewards that evaluate intermediate behavior directly from gameplay observations.

Vision-language models (VLMs) offer a natural way to construct such rewards by evaluating visual behavior against natural-language objectives (Rocamonde et al., 2024; Fan et al., 2022; Alakuijala et al., 2024). Rubric rewards further decompose an objective into behavioral criteria, such as attacking, dodging, and resource use, and aggregate their scores into a training signal (Gunjal et al., 2026). Because the usefulness of these criteria can change with policy behavior, their weights can be adapted from observed score differences. POW3R, for example, adjusts criterion weights using score differences across responses sampled from the current policy (Tyagi et al., 2026). This follows the intuition that greater score variation provides greater potential for discrimination.

In real-time visual games, however, observed score variation is not necessarily reliable evidence for weight adaptation. Across-segment variance can reflect behavioral differences, but also judge inconsistency caused by brief action windows, occlusions, or visual effects. Rubrics also differ substantially in how often they are applicable, so their observed variation can be supported by very different numbers of evaluation opportunities. Dynamic weighting must therefore consider two complementary dimensions of evidence: how much of the observed variation is reliable, and how much opportunity supports that variation.

Figure 1 illustrates both dimensions. R7 and R8 have similar observed variance, yet repeated audits yield noise-to-variance ratios of 8.5% and 69.6%, respectively, revealing substantially different reliable variation. In contrast, R6 applies to only 3.26% of segments, providing limited opportunity support for adapting its weight. Thus, repeated audits characterize the reliability of observed variation, while coverage characterizes the support behind it. Low coverage does not imply that a rubric is unimportant; rather, the current interaction data provide less evidence for adapting its weight.

Motivated by this view, we introduce NOISERUB, a noise-calibrated approach to dynamic rubric rewards that treats weight adaptation as an evidence-calibration problem. For each rubric, NOISERUB estimates reliable variation by subtracting a smoothed within-segment noise floor, obtained from repeated audits, from across-segment variance and clipping the residual at zero. It then modulates this residual by opportunity coverage, so that weight adaptation reflects both variation beyond judge noise and the support for that variation. The resulting weights aggregate applicable rubric scores into segment rewards for online policy optimization. We evaluate NOISERUB in *Black Myth: Wukong*. In sequential offline replay, it achieves 63.5% agreement with human overall preferences, compared with 57.4% for fixed weighting and 60.0% for POW3R-Game. Under matched initialization and interaction budgets, NOISERUB achieves the highest HP reduction on all four evaluated bosses, exceeding POW3R-Game by 4.45–8.04 percentage points.

Our main contributions are threefold:

- We formulate dynamic rubric weighting as an evidence-calibration problem, distinguishing the reliability of observed variation from the opportunity support behind it.
- We introduce NOISERUB, which combines repeated-audit noise calibration with opportunity coverage to construct adaptive process rewards for screen-only policy learning.
- NOISERUB improves human preference agreement over fixed weighting and POW3R-Game and achieves stronger policy performance across four bosses in *Black Myth: Wukong*.

2 RELATED WORK

Visual Game Agents. Visual game-agent research explores learning behavioral priors from game screens, following language instructions, and executing general-purpose controls (Lifshitz et al.,

2023; Zhou et al., 2024). VPT (Baker et al., 2022) uses an inverse dynamics model to label actions in internet gameplay videos, supporting behavioral pretraining and subsequent policy fine-tuning. SIMA (Raad et al., 2024) trains instruction-following policies across multiple 3D environments using image observations and keyboard-and-mouse actions. Cradle (Tan et al., 2024) uses a multimodal model to interpret screenshots and generate executable keyboard-and-mouse control code through high-level planning. MineDojo (Fan et al., 2022) uses video-language rewards for language-specified open-ended game tasks. We focus on online training feedback in this visual interaction setting, constructing process rewards from combat segments.

Foundation-Model Rewards and Visual Evaluation. Foundation-model rewards can be constructed through visual-language matching, preference learning, or reward-code generation. VLM-RM (Rocamonde et al., 2024) and Baumli et al. (2023) derive rewards from alignment between visual observations and language goals. LIV (Ma et al., 2023) jointly learns vision-language representations and progress-related rewards, while RoboCLIP (Sontakke et al., 2023) constructs rewards from a single video demonstration or text description. RL-VLM-F (Wang et al., 2024) queries a VLM for preferences between image observations and learns a reward function from the resulting labels. With access to environment variables or programmatic interfaces, Text2Reward (Xie et al., 2024), Eureka (Ma et al., 2024), and Auto MC-Reward (Li et al., 2024) generate executable reward code. Eureka and Auto MC-Reward iteratively refine it using training statistics or trajectory feedback. Visual and language-model judging can exhibit inconsistencies and evaluation biases (Chen et al., 2024; Shi et al., 2025), motivating attention to evaluation quality when deriving rewards from game screens. Reward Annotation Model (RAM) (Schmid & Frosio, 2026a) learns a linear combination of VLM outputs from a small human-annotated reference set to improve gameplay reward annotations. A companion study converts local visual reward annotations into returns over short time windows and trains a return-conditioned policy offline (Schmid & Frosio, 2026b). RAM focuses on improving annotation outputs; we use repeated-evaluation statistics to calibrate reward weights during online training, with human labels used only for offline evaluation.

Rubric Rewards and Policy-Aware Weighting. Rubric-based rewards decompose open-ended objectives into individually assessable requirements. RaR (Gunjal et al., 2026) evaluates criteria and aggregates their scores into rewards, extending reinforcement learning supervision beyond directly verifiable tasks. RLCF (Viswanathan et al., 2026) generates checklists from instructions, combines model judgments with specialized verifier programs, and averages repeatedly sampled judge scores. Rubric Anchors (Huang et al., 2025) defines explicit score tiers and importance weights for evaluation dimensions to construct rewards for open-ended tasks. POW3R (Tyagi et al., 2026) uses human base weights as a prior and adapts criterion weights from score differences across multiple policy outputs for the same prompt while maintaining category balance. We apply dynamic weighting across game segments collected during continuous interaction, where evaluation opportunities and scoring reliability can vary. NOISERUB uses repeated scores to aggregate rewards and estimate judge noise, combining the noise estimate with coverage to calibrate weight updates for reliability and opportunity support.

3 METHOD

3.1 VISUAL INTERACTION AND RUBRIC REWARDS

We study online policy learning in real-time visual games. At time t, the actor receives a screen observation o_t and selects a keyboard-and-mouse action from its observation–action history h_t:

$$u_t \sim \pi_\theta(\cdot \mid h_t), \tag{1}$$

where θ denotes the policy parameters. Real-time interaction requires the actor to produce actions with low latency, whereas evaluating attacks, dodges, and other behaviors requires observing a continuous sequence. We therefore organize action selection and segment-level evaluation separately. Collected trajectories are divided into ten-second segments, preserving local behavioral context while bounding the input length of each visual evaluation. These segments provide the units for reward computation and assignment. Let $\mathcal{B}_k$ denote the set of segment indices in calibration batch k, τ_i the segment indexed by $i \in \mathcal{B}_k$, and I_i its corresponding action-time indices. Figure 2 illustrates the overall framework.

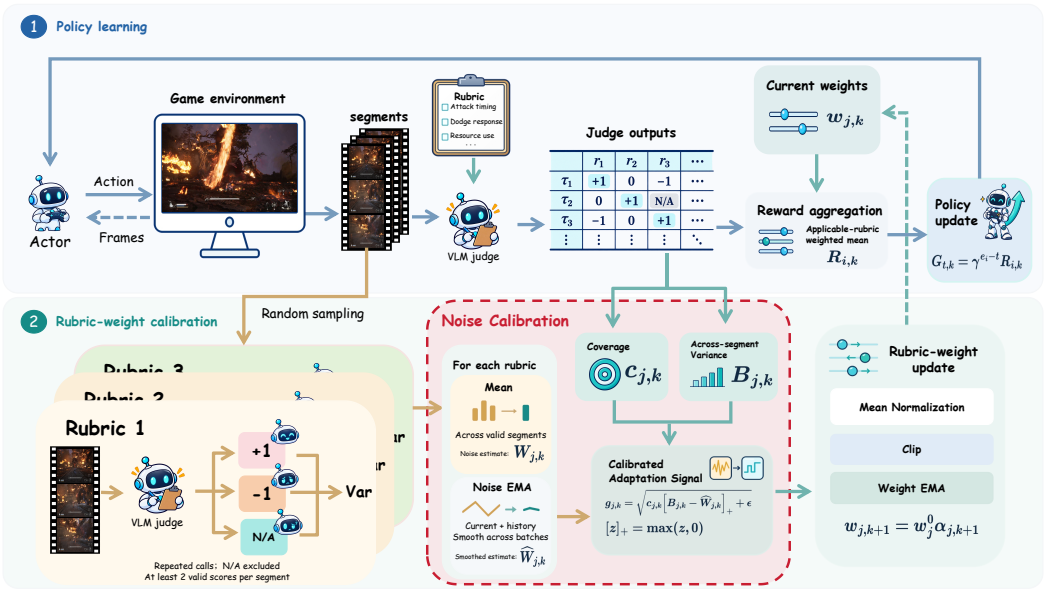


Figure 2: Overview of NOISERUB. Top: the actor collects visual interaction segments, which a VLM judge evaluates against the rubric bank. Current rubric weights aggregate applicable scores into segment rewards for policy learning. Reward aggregation uses the mean of valid scores, while coverage and across-segment variance use first-pass scores. Bottom: repeated audits estimate within-segment judge variance, which is averaged across valid audited segments and smoothed over batches. Coverage and across-segment variance combine with the noise estimate to form the calibrated contrast. Mean normalization, clipping, and weight EMA produce rubric weights for subsequent interaction batches.

A fixed rubric bank $\mathcal{R} = \{r_j\}_{j=1}^{J}$ describes task-relevant behaviors, including attack timing, dodge responses, and resource use. For segment τ_i and rubric r_j, the VLM judge returns an applicability indicator $a_{ij}^{(m)} \in \{0, 1\}$ on call m. When applicable, it assigns $s_{ij}^{(m)} \in \{-1, 0, +1\}$ for clear failure, partial success, or clear success, respectively. An N/A output indicates that no corresponding evaluation opportunity occurs or that visible evidence is insufficient. N/A outputs are excluded from aggregation rather than encoded as zero.

Let $\mathcal{V}_{ij} = \{m : a_{ij}^{(m)} = 1\}$ collect the valid evaluations. The reward score and its applicability are

$$\bar{s}_{ij} = \begin{cases} \dfrac{1}{|\mathcal{V}_{ij}|} \displaystyle\sum_{m \in \mathcal{V}_{ij}} s_{ij}^{(m)}, & |\mathcal{V}_{ij}| > 0, \\ 0, & \text{otherwise}, \end{cases} \qquad \bar{a}_{ij} = \mathbf{1}\{|\mathcal{V}_{ij}| > 0\}. \tag{2}$$

We average multiple valid evaluations and use the sole valid score when only one is available. Given the current rubric weights $w_{j,k}$, the segment reward is

$$R_{i,k} = \frac{\sum_{j=1}^{J} \bar{a}_{ij} w_{j,k} \bar{s}_{ij}}{\sum_{j=1}^{J} \bar{a}_{ij} w_{j,k} + \epsilon}, \tag{3}$$

where $\epsilon > 0$ is a numerical stability constant. This weighted mean includes only applicable rubrics and returns zero when none is applicable.

Reward aggregation uses the mean of valid scores, while coverage and across-segment variance use first-pass evaluations. These statistics retain a common single-call evaluation protocol for weight calibration.

3.2 NOISE-CALIBRATED ACROSS-SEGMENT CONTRAST

Dynamic weighting uses score differences across segments collected by recent policies to adjust each criterion's training influence. Evaluation opportunities vary across game segments, and judge scores can fluctuate even for a fixed segment. NOISERUB accounts for both opportunity coverage and judge noise when constructing the update signal.

For rubric r_j, define the first-pass applicable set

$$\mathcal{I}_{j,k} = \{i \in \mathcal{B}_k : a_{ij}^{(1)} = 1\}. \tag{4}$$

Coverage and observed across-segment variance are

$$c_{j,k} = \frac{|\mathcal{I}_{j,k}|}{|\mathcal{B}_k|}, \qquad B_{j,k} = \mathrm{Var}_{i \in \mathcal{I}_{j,k}} \left(s_{ij}^{(1)} \right). \tag{5}$$

Coverage is the proportion of segments judged applicable on the first evaluation, while variance measures score contrast among those segments. We compute the population variance of first-pass valid scores, using the number of valid segments as the denominator.

To relate across-segment score variation to repeated-evaluation fluctuations, we consider the conditional law of total variance. Let I denote a randomly selected segment, M a random judge call, S the resulting score, and $A = 1$ an applicable evaluation. At the population level,

$$\underbrace{\mathrm{Var}(S \mid A = 1)}_{\text{total score variance}} = \underbrace{\mathrm{Var}_I(\mathbb{E}_M[S \mid I, A = 1] \mid A = 1)}_{\text{between-segment variation}} + \underbrace{\mathbb{E}_I[\mathrm{Var}_M(S \mid I, A = 1) \mid A = 1]}_{\text{within-segment variation}}. \tag{6}$$

The two terms describe variation in expected scores across segments and scoring fluctuations within a fixed segment. Motivated by this decomposition, we use repeated within-segment scoring variability as an empirical calibration term, without interpreting the residual as an unbiased estimate of between-segment behavioral variation.

We repeatedly audit a subset of segments, holding the frames, rubric, prompt, and judge configuration fixed. For each rubric, let $\mathcal{Q}_{j,k}$ contain audited segments with at least two valid scores. We compute variance within each segment and then average across segments:

$$v_{ij} = \frac{1}{|\mathcal{V}_{ij}|} \sum_{m \in \mathcal{V}_{ij}} \left(s_{ij}^{(m)} - \bar{s}_{ij} \right)^2, \qquad W_{j,k} = \frac{1}{|\mathcal{Q}_{j,k}|} \sum_{i \in \mathcal{Q}_{j,k}} v_{ij}. \tag{7}$$

Within each audited segment, we exclude N/A outputs and compute the population variance of its valid repeated scores. We then average these variances equally across eligible audited segments to obtain the empirical noise-calibration term. Segments with more valid judgments therefore do not receive greater weight in this average. When a valid audit estimate is available, an exponential moving average smooths it across batches:

$$\widehat{W}_{j,k} = \eta_W \widehat{W}_{j,k-1} + (1 - \eta_W) W_{j,k}, \tag{8}$$

where $\eta_W \in [0, 1)$ controls the contribution of historical estimates. The first available estimate initializes the EMA; an existing estimate is retained when no new estimate is available. Combining noise correction with coverage gives

$$g_{j,k} = \sqrt{c_{j,k} \left[B_{j,k} - \widehat{W}_{j,k} \right]_+ + \epsilon}, \qquad [z]_+ = \max(z, 0). \tag{9}$$

The nonnegative projection sets the residual to zero when the audit-noise estimate exceeds the observed variance. For a given across-segment variance, greater estimated judge variability or lower coverage reduces the calibrated signal. We use $g_{j,k}$ as the input to subsequent rubric-weight adaptation.

3.3 DYNAMIC RUBRIC-WEIGHT ADAPTATION

Following POW3R's policy-aware weighting principle (Tyagi et al., 2026), we use calibrated signals $g_{j,k}$ to adjust human-assigned base weights w_j^0. For the nonempty set $\mathcal{C}_k$ of eligible criteria, define

$$\bar{g}_k = \frac{1}{|\mathcal{C}_k|} \sum_{j \in \mathcal{C}_k} g_{j,k}, \qquad \rho_{j,k} = \frac{g_{j,k}}{\bar{g}_k}, \quad j \in \mathcal{C}_k. \tag{10}$$

Each criterion contributes equally to the mean. Mixing the relative signal with one and clipping gives the target multiplier:

$$\widetilde{\alpha}_{j,k} = \mathrm{clip}\left((1-\lambda) + \lambda\rho_{j,k}, \alpha_{\min}, \alpha_{\max}\right), \quad j \in \mathcal{C}_k, \tag{11}$$

where $\lambda \in [0,1]$ controls adaptation strength. An EMA with coefficient $\beta \in [0,1]$ updates the bounded multipliers:

$$\begin{aligned} \alpha_{j,k+1} &= \mathrm{clip}\left((1-\beta)\alpha_{j,k} + \beta\widetilde{\alpha}_{j,k}, \alpha_{\min}, \alpha_{\max}\right), \\ w_{j,k+1} &= w_j^0 \alpha_{j,k+1}. \end{aligned} \tag{12}$$

Multipliers start at one. Batch k uses $w_{j,k}$, and updated weights apply to subsequent batches. Eligibility and fallback settings are given in Appendix A.2.

3.4 Policy Optimization

For segment τ_i ending at action step $e_i = \max I_i$, we propagate its reward backward with discount $\gamma \in (0,1]$:

$$G_{t,k} = \gamma^{e_i - t} R_{i,k}, \qquad t \in I_i. \tag{13}$$

We use $A_{t,k} = G_{t,k}$ directly as the advantage, without a value network or advantage normalization. Let $\mathcal{M}$ contain the token positions included in the loss, y_ℓ denote an action token conditioned on x_ℓ, and $t(\ell)$ its action step. With the old policy π_{old} used to collect the training data held fixed during the update, the probability ratio is

$$r_\ell(\theta) = \frac{\pi_\theta(y_\ell \mid x_\ell)}{\pi_{\mathrm{old}}(y_\ell \mid x_\ell)}. \tag{14}$$

The token-level PPO (Schulman et al., 2017) clipped loss is

$$\mathcal{L}_{\mathrm{RL}}(\theta) = -\frac{1}{|\mathcal{M}|} \sum_{\ell \in \mathcal{M}} \min\left(r_\ell(\theta) A_{t(\ell),k}, \mathrm{clip}(r_\ell(\theta), 1-\epsilon_{\mathrm{low}}, 1+\epsilon_{\mathrm{high}})\, A_{t(\ell),k}\right), \tag{15}$$

where ϵ_{low} and ϵ_{high} set the lower and upper clipping limits. Training settings appear in Appendix A.1.

4 Experiments

4.1 Experimental Setup

Task and training configuration. We evaluate NoiseRub in the purely visual interaction setting of *Black Myth: Wukong*. Offline evaluation examines reward rankings on Black Bear Guai combat segments. Online evaluation tests policy performance on White-Clad Noble, Black Wind King, Black Bear Guai, and Wandering Wight. All post-training methods start from the same Qwen3.5-2B SFT policy, trained on 79 hours of demonstrations covering first-chapter Boss encounters, and use the critic-free, token-level PPO clipped objective described in the method section. For each post-training method, we conduct one independent training run per Boss to obtain its policy. Training lasts 30 rounds with 200 seconds of interaction per round, giving 6,000 seconds per Boss and 24,000 seconds across four Bosses for each method, approximately 6.67 hours. This budget excludes judge evaluation and model optimization time.

Offline data. The offline dataset contains 430 ten-second combat segments sampled by Boss HP strata and evaluated against 11 fixed criteria. The judge evaluates each segment once. Segments selected with probability 20% receive two additional evaluations with identical frames, criteria, prompts, and judge configuration. Human applicability annotations comprise 257 labels across 188 segments, including 53 labels in the test split. Human pairwise preferences evaluate reward rankings and do not enter weight estimation.

Compared methods. Offline evaluation compares RaR-Equal (equal weights), RaR-Static (fixed weights), POW3R-Game, and Ours. Online evaluation compares SFT, Outcome Reward, Direct VLM Reward, RaR-Static, POW3R-Game, and Ours. SFT is the shared initial policy. Outcome Reward assigns victory $(+1)$ or defeat (-1) as the terminal reward of every chunk in the episode. Direct VLM Reward directly scores each visual segment in $[-1,1]$. RaR-Static uses fixed criterion weights, while POW3R-Game adapts weights using score variance. Reward propagation and

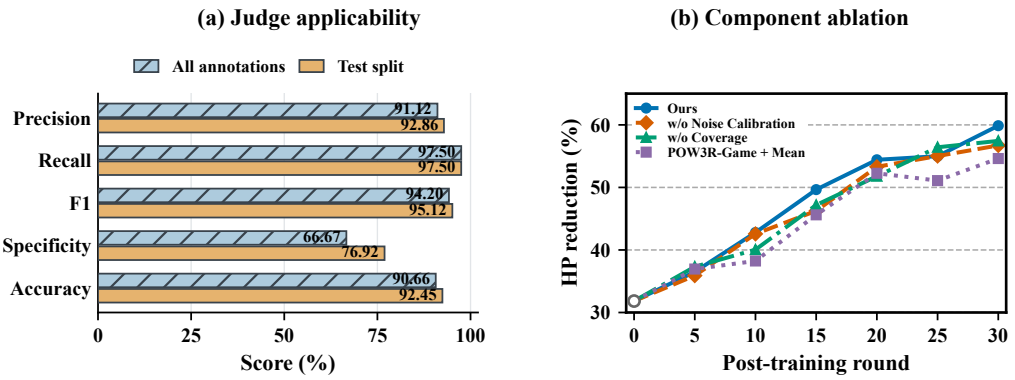


Figure 3: Judge validation and online ablation. (a) Agreement between judge applicability decisions and human annotations; all annotations include the test split. (b) HP reduction over 30 training rounds on Black Bear Guai. Each point averages 20 test episodes, including failures, under the shared protocol in Section 4.1.

policy optimization are shared across post-training methods. Appendix A.5 defines their rewards and weight updates. Black Bear Guai ablations compare Ours with variants removing noise calibration or coverage modulation, and with POW3R-Game + Mean. These four settings share SFT initialization, interaction budget, repeated-audit sampling, and valid-score averaging.

Evaluation protocol. Offline evaluation replays scoring records in their original round order. Weights are updated after accumulating at least 100 new segments and take effect in subsequent rounds. Each segment retains the reward computed with the weights active in its historical round for comparison with human preferences. Online evaluation tests the policy after round 30 using 20 episodes per method and Boss. Ablations are evaluated using 20 episodes at rounds 0, 5, 10, 15, 20, 25, and 30, with a shared SFT starting point.

Metrics. We evaluate judge applicability decisions using precision, recall, F1, specificity, and accuracy. Offline rewards are evaluated using Overall Pairwise Accuracy (PAcc), the proportion of reward rankings that agree with human overall preferences. Human and model ties are excluded, and we report 95% confidence intervals. Online evaluation reports win rate and HP reduction. Ablations use HP reduction. For each episode, HP reduction is the decrease in Boss HP divided by its initial HP, averaged over all test episodes, including failures. For multi-phase Bosses, HP is summed across all phases, following Appendix A.1. Metric definitions are given in Appendices A.1 and A.3.

4.2 Offline Reward Evaluation

Judge applicability. We first assess whether the judge identifies the criteria that apply to each segment. Figure 3(a) compares its decisions with human annotations. Test precision and recall are 92.86% and 97.50%, with an F1 of 95.12%. The judge therefore recovers most human-labeled applicable cases, while most predicted applicable cases agree with the annotations. These results support using its applicability decisions to select criteria for reward aggregation.

Repeated-audit analysis. Among the 77 audited segments, each criterion has between 1 and 47 segments with at least two valid scores. Applicability can change across repeated calls, so the eligible audit set differs from the first-pass applicable set. Weight updates therefore use criterion-specific eligibility checks; Appendix A.3 provides the full statistics and computation details.

Table 1: Reward ranking agreement under sequential replay (%).

Method	PAcc ↑	95% CI
RaR-Equal	58.8	[41.5, 81.8]
RaR-Static	57.4	[41.2, 75.7]
POW3R-Game	60.0	[43.3, 77.5]
Ours	**63.5**	[48.7, 78.6]

Agreement with human preferences. We next evaluate whether the resulting rewards rank combat segments in agreement with human overall preferences.

Table 1 reports Overall PAcc under sequential replay. RaR-Equal and RaR-Static obtain 58.8% and 57.4%, respectively. POW3R-Game reaches 60.0%, while Ours achieves 63.5%, exceeding RaR-Static and POW3R-Game by 6.1 and 3.5 percentage points. These latter three methods are evaluated

Table 2: Online evaluation at round 30. WR: win rate; HR: HP reduction (both in %). HP reduction averages all 20 test episodes, including failures. Higher is better.

	White-Clad Noble		Black Wind King		Black Bear Guai		Wandering Wight	
Method	WR ↑	HR ↑	WR ↑	HR ↑	WR ↑	HR ↑	WR ↑	HR ↑
SFT	0	29.50	0	39.10	0	31.83	5	44.24
Outcome Reward	0	6.14	0	4.48	0	7.19	0	8.07
Direct VLM Reward	0	53.06	0	47.74	0	41.17	5	53.61
RaR-Static	5	58.84	0	48.83	0	46.09	5	59.95
POW3R-Game	5	61.57	0	52.46	5	51.83	**15**	70.34
Ours	**15**	**66.02**	**5**	**59.85**	**10**	**59.87**	**15**	**75.14**

on the same 115 valid pairs: Ours agrees with human preferences on 73 pairs, compared with 66 for RaR-Static and 69 for POW3R-Game.

Ours achieves the highest Overall PAcc point estimate among the compared methods, with reward rankings more closely aligned with human overall preferences in this evaluation. Appendix A.3 provides the evaluation protocol and treatment of ties.

4.3 Online Policy Evaluation

We evaluate whether the reward schemes improve policy performance after 30 training rounds. Table 2 reports results on four Bosses under a shared SFT initialization and an interaction budget of 6,000 seconds per Boss. Ours achieves the highest HP reduction on every task and the highest or jointly highest win rate.

Segment evaluation and terminal feedback. Outcome Reward reaches only 4.48%–8.07% HP reduction, below SFT on all four tasks. Direct VLM Reward and RaR-Static both improve upon SFT across these tasks. On Black Bear Guai, they reach 41.17% and 46.09%, respectively, compared with 31.83% for SFT. Under the evaluated training budget, visual segment feedback improves combat progress, whereas training with episode outcomes alone degrades performance. The advantage of both segment-level rewards suggests that evaluating intermediate combat behavior provides useful learning signals even when successful episodes are rare.

Dynamic rewards and fixed weights. POW3R-Game exceeds RaR-Static in HP reduction on all four Bosses, and Ours further improves upon POW3R-Game. The gains are 4.45, 7.39, 8.04, and 4.80 percentage points on White-Clad Noble, Black Wind King, Black Bear Guai, and Wandering Wight, respectively. Both dynamic reward schemes outperform fixed weighting across the evaluated encounters, with NoiseRub achieving the highest final HP reduction.

Combat progress and wins. Ours achieves win rates of 15%, 5%, and 10% on White-Clad Noble, Black Wind King, and Black Bear Guai, exceeding POW3R-Game by 10, 5, and 5 percentage points, respectively. On Wandering Wight, both methods achieve a win rate of 15%, but Ours increases HP reduction from 70.34% to 75.14%. Thus, HP reduction across all episodes, including failures, distinguishes combat progress even when win rates are equal.

4.4 Component Ablation

We examine the contributions of noise calibration and coverage modulation on Black Bear Guai. Removing noise calibration retains coverage modulation but omits subtraction of the judge-noise estimate. Removing coverage modulation retains noise calibration but omits coverage from the update signal. Both component ablations modify only the corresponding term in Equation 9. All other settings remain identical to NoiseRub, including criterion eligibility thresholds, fallback rules, repeated-score aggregation, and weight-update settings. POW3R-Game + Mean retains POW3R-Game's weight updates and uses valid-score averaging. The four settings share the audit, aggregation, and training protocols specified in Section 4.1.

Final performance. At round 30, Ours achieves 59.87% HP reduction. Removing noise calibration reduces this to 56.71%, a decrease of 3.16 percentage points. Removing coverage modulation reduces it to 57.44%, a decrease of 2.43 percentage points. Each removal lowers final performance, indicating that both components contribute under the evaluated training configuration.

Comparison with POW3R-Game + Mean. With the same interaction budget, repeated-audit sampling procedure, and valid-score averaging rule, Ours reaches 59.87% HP reduction compared with

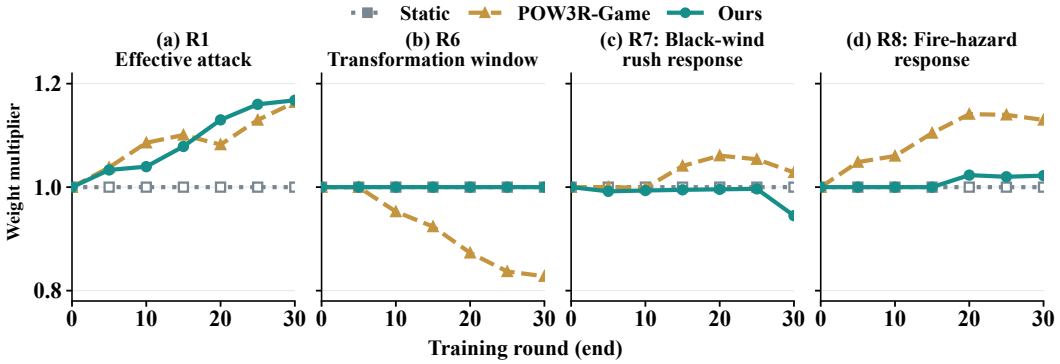


Figure 4: Representative evaluation-criterion weight multipliers during online training on Black Bear Guai. Each panel compares Static, POW3R-Game, and Ours using a common vertical scale. A multiplier of one denotes the base weight. Round zero denotes initialization; subsequent markers show multipliers recorded at the end of every fifth round. Lines connect the recorded checkpoints.

54.62% for POW3R-Game + Mean at round 30. The 5.25-point gap remains when both methods average repeated scores, so averaging alone does not explain the observed advantage. This comparison links the final performance difference to how the evaluation criteria are weighted, beyond the use of repeated scores in reward aggregation.

Training progression. All settings start from 31.83% HP reduction in Figure 3(b). Ours has the highest mean at rounds 10, 15, 20, and 30, while the variant without coverage modulation leads at rounds 5 and 25. Every setting improves over SFT by round 30. The full method achieves the highest final performance, although the ordering of methods varies during training.

4.5 Weight Dynamics

Figure 4 examines how the calibration schemes redistribute criterion weights during Black Bear Guai training. Multipliers express weights relative to their base values. Effective attack (R1) receives an overall increase under both methods, reaching 1.164 for POW3R-Game and 1.168 for Ours at round 30. Transformation window utilization (R6) instead separates the two schemes: POW3R-Game lowers its multiplier to 0.828, while Ours stays at one at every recorded checkpoint.

For Black Wind rush response (R7), Ours stays near the base multiplier from rounds 5 to 25 and falls to 0.945 at round 30, whereas POW3R-Game ends at 1.029. For fire hazard response (R8), Ours makes a smaller upward adjustment, ending at 1.022 compared with 1.130 for POW3R-Game.

NoiseRub selectively adjusts criterion weights: its allocation is close to POW3R-Game for effective attack but differs in direction or magnitude for the other three criteria.

4.6 Cross-Game Evaluation

We further compare SFT and NoiseRub on two Boss tasks in *Honor of Kings: World*: Yuange Puppet (Phase 3) and Qingyue Zhujiang. Starting from an SFT policy trained on 8.3 hours of gameplay demonstrations, NoiseRub post-training improves both win rate and mean HP reduction on both tasks. Each method is evaluated over 20 episodes per Boss; Appendix A.6 provides the full settings and results.

5 Conclusion

We introduced NoiseRub, which uses repeated audits to estimate judge scoring variability and combines this empirical calibration term with opportunity coverage to adapt rubric reward weights. Current weights aggregate segment rewards, while calibrated weights apply to subsequent interaction batches. NoiseRub achieves the highest offline Overall PAcc among the compared methods and the highest HP reduction on all four evaluated Bosses, together with the highest or jointly highest win rate. Component ablations on Black Bear Guai show higher final HP reduction when both noise calibration and coverage modulation are retained under the evaluated training configuration.

AI USE STATEMENT

We used generative AI tools to assist with code implementation, literature search and summarization, manuscript organization, English drafting, and language polishing. The authors developed the research ideas and designed the experiments, with AI tools used for discussion and consistency checks. Experimental data came from actual gameplay and were reviewed by the authors; VLM scoring and repeated audits are described in the paper, while applicability and pairwise preference labels were provided by human annotators. All final analyses, editorial decisions, and conclusions were determined by the authors, who take full responsibility for the final content of this work.

REFERENCES

Minttu Alakuijala, Reginald McLean, Isaac Woungang, Nariman Farsad, Samuel Kaski, Pekka Marttinen, and Kai Yuan. Video-language critic: Transferable reward functions for language-conditioned robotics. *arXiv preprint arXiv:2405.19988*, 2024. 2

Bowen Baker, Ilge Akkaya, Peter Zhokov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video pretraining (vpt): Learning to act by watching unlabeled online videos. *Advances in Neural Information Processing Systems*, 35:24639–24654, 2022. 3

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. 3

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of artificial intelligence research*, 47: 253–279, 2013. 2

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. *arXiv preprint arXiv:2402.04788*, 2024. 3

Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge. *Advances in Neural Information Processing Systems*, 35: 18343–18362, 2022. 2, 3

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. In *International Conference on Learning Representations*, volume 2026, pp. 127924–127945, 2026. 2, 3

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025. 2

Zenan Huang, Yihong Zhuang, Guoshan Lu, Zeyu Qin, Haokai Xu, Tianyu Zhao, Ru Peng, Jiaqi Hu, Zhanming Shen, Xiaomeng Hu, et al. Reinforcement learning with rubric anchors. *arXiv preprint arXiv:2508.12790*, 2025. 3

Hao Li, Xue Yang, Zhaokai Wang, Xizhou Zhu, Jie Zhou, Yu Qiao, Xiaogang Wang, Hongsheng Li, Lewei Lu, and Jifeng Dai. Auto mc-reward: Automated dense reward design with large language models for minecraft. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16426–16435. IEEE, 2024. 2, 3

Shalev Lifshitz, Keiran Paster, Harris Chan, Jimmy Ba, and Sheila McIlraith. Steve-1: A generative model for text-to-behavior in minecraft. *Advances in Neural Information Processing Systems*, 36: 69900–69929, 2023. 2

Yecheng Jason Ma, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. Liv: Language-image representations and rewards for robotic control. In *International Conference on Machine Learning*, pp. 23301–23320. PMLR, 2023. 3

Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Jim Fan, et al. Eureka: Human-level reward design via coding large language models. In *International conference on learning Representations*, volume 2024, pp. 26516–26560, 2024. 3

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015. 2

Maria Abi Raad, Arun Ahuja, Catarina Barros, Frederic Besse, Andrew Bolt, Adrian Bolton, Bethanie Brownfield, Gavin Buttimore, Max Cant, Sarah Chakera, et al. Scaling instructable agents across many simulated worlds. *arXiv preprint arXiv:2404.10179*, 2024. 3

Juan Rocamonde, Victoriano Montesinos, Elvis Nava, Ethan Perez, and David Lindner. Vision-language models are zero-shot reward models for reinforcement learning. In *International Conference on Learning Representations*, volume 2024, pp. 28446–28463, 2024. 2, 3

Katrin Schmid and Iuri Frosio. Vlms for videogame data annotation. *arXiv preprint arXiv:2608.05949*, 2026a. 3

Katrin Schmid and Iuri Frosio. Training a conditioned video game agent on a vlm annotated dataset. *arXiv preprint arXiv:2608.05954*, 2026b. 3

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 6

Lin Shi, Chiyu Ma, Wenhua Liang, Xingjian Diao, Weicheng Ma, and Soroush Vosoughi. Judging the judges: A systematic study of position bias in llm-as-a-judge. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pp. 292–314, 2025. 3

Sumedh Sontakke, Jesse Zhang, Séb Arnold, Karl Pertsch, Erdem Bıyık, Dorsa Sadigh, Chelsea Finn, and Laurent Itti. Roboclip: One demonstration is enough to learn robot policies. *Advances in Neural Information Processing Systems*, 36:55681–55693, 2023. 3

Weihao Tan, Wentao Zhang, Xinrun Xu, Haochong Xia, Ziluo Ding, Boyu Li, Bohan Zhou, Junpeng Yue, Jiechuan Jiang, Yewen Li, et al. Cradle: Empowering foundation agents towards general computer control. *arXiv preprint arXiv:2403.03186*, 2024. 3

Utkarsh Tyagi, Xingang Guo, MohammadHossein Rezaei, Daniel George, Anas Mahmoud, Jackson Lee, Bing Liu, and Yunzhong He. Not every rubric teaches equally: Policy-aware rubric rewards for rlvr. *arXiv preprint arXiv:2605.20164*, 2026. 2, 3, 5, 20

Vijay Viswanathan, Yanchao Sun, Xiang Kong, Meng Cao, Graham Neubig, and Sherry Wu. Checklists are better than reward models for aligning language models. *Advances in Neural Information Processing Systems*, 38:114728–114754, 2026. 3

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. *arXiv preprint arXiv:2402.03681*, 2024. 3

Tianbao Xie, Siheng Zhao, Chen Wu, Yitao Liu, Qian Luo, Victor Zhong, Yanchao Yang, and Tao Yu. Text2reward: Reward shaping with language models for reinforcement learning. In *International Conference on Learning Representations*, volume 2024, pp. 35663–35699, 2024. 3

Enshen Zhou, Yiran Qin, Zhenfei Yin, Yuzhou Huang, Ruimao Zhang, Lu Sheng, Yu Qiao, and Jing Shao. Minedreamer: Learning to follow instructions via chain-of-imagination for simulated-world control. *arXiv preprint arXiv:2403.12037*, 2024. 3

A Additional Details

A.1 Training and Evaluation Details

All online post-training methods start from the same Qwen3.5-2B SFT policy, trained on 79 hours of demonstrations covering Boss encounters in the first chapter of Black Myth: Wukong. We train a separate policy for White-Clad Noble, Black Wind King, Black Bear Guai, and Wandering Wight. Each policy undergoes 30 training rounds with 200 seconds of game interaction per round, giving 6,000 seconds per Boss for each method. All training and evaluation runs use the same player attributes: initial HP of 360, attack of 40, and defense of 116.

The policy receives game images at a resolution of 512×512 and predicts actions at 5 Hz. We fine-tune the language model while freezing the vision encoder and multimodal projector. Optimization uses a PPO clipped objective with action returns directly serving as advantages, without a critic or advantage normalization. The lower and upper clipping parameters are 0.2 and 0.28, respectively. We use AdamW with a learning rate of 10^{-6}, weight decay of 0.01, and a maximum gradient norm of 1.0, training for one epoch per round. The reward discount factor is 0.98.

For the main evaluation, each method is tested on 20 episodes per Boss after round 30. Black Bear Guai ablations are evaluated on 20 episodes at rounds 0, 5, 10, 15, 20, 25, and 30. Boss HP reduction is averaged over all test episodes, including failures. Victory against a multi-phase Boss requires defeating every phase. For multi-phase Bosses, initial and remaining HP are summed across all phases. Defeated phases have zero remaining HP, while phases not yet reached retain their full HP.

Let N be the number of test episodes, z_e indicate victory in episode e, H_e^{init} denote its initial Boss HP, and ΔH_e denote the Boss HP decrease used in the episode evaluation. We report

$$\text{WinRate} = \frac{1}{N}\sum_{e=1}^{N} z_e, \qquad \text{HPReduction} = \frac{1}{N}\sum_{e=1}^{N} \frac{\Delta H_e}{H_e^{\text{init}}}. \tag{16}$$

Both metrics are expressed as percentages in the results.

A.2 Rubric Construction, Evaluation, and Weight Calibration

We use GPT-5.6-Sol to generate initial evaluation criteria and finalize them through human expert review. Black Bear Guai uses a fixed set of 11 criteria; the complete criteria and prompts appear in Appendix A.4. The judge is `gemini-3.6-flash`. Each input contains a ten-second target segment with two seconds of context on either side, sampled at 5 FPS and resized to a short-side resolution of 720 pixels. Scoring is restricted to events that begin within the target interval. Prompts consist of a fixed System Prompt and a segment-specific User Message.

Applicable criteria receive scores of -1, 0, or $+1$, indicating clear failure, partial success, and clear success, respectively. Events that cannot be fully judged receive N/A and are excluded from score aggregation. Base weights are withheld from the judge and used only during reward aggregation. Our method selects segments for repeated auditing with a probability of 20%, giving each selected segment three evaluations in total. For each criterion, reward aggregation uses the mean of available valid scores. Coverage and across-segment variance are computed from first-pass scores, while repeated scores estimate judge noise.

Weights are updated at the end of a training round once at least 100 new segments with first-pass evaluations have accumulated. Each new segment is counted once toward this threshold; repeated audit calls do not add segments to the batch. Updates use non-overlapping batches, and new weights take effect in the next round. Calibrated signals are normalized by the arithmetic mean across eligible criteria. Adaptation strength is 0.5 and multiplier bounds are 0.67–1.5. The noise EMA uses a history coefficient of 0.9, and the multiplier EMA assigns a coefficient of 0.2 to the new target. When coverage is below 5%, fewer than four eligible audited segments are available, or the variance cannot be estimated, the target multiplier is set to one and the current multiplier returns toward one through EMA smoothing.

If no criterion meets the calibration requirements, all target multipliers are set to one. A window with fewer than four eligible audited segments can still provide a noise estimate for the noise EMA,

while its target multiplier follows the fallback rule. The noise EMA is initialized by the first available estimate and retains its previous value when no new estimate is available.

A.3 Offline Evaluation and Audit Analysis

Offline evaluation uses 430 ten-second Black Bear Guai combat segments and replays scoring records in their original round order, following the calibration protocol in Appendix A.2. Each segment retains the weights and reward from its original evaluation for comparison with human preferences. Offline PAcc evaluates complete reward schemes, including score aggregation and weight adaptation, complementing the online component ablations.

Human annotations are provided by two annotators who have completed the game. Applicability labels follow the conditions of the corresponding evaluation criteria. Pairwise preferences reflect the contribution of the player's behavior in each segment toward ultimately defeating the Boss. We select 20% of samples for annotation by both annotators and reannotate samples with disagreements to determine the final labels. Human labels are used only for offline evaluation and do not enter weight calibration.

Applicability evaluation uses 257 human labels across 188 segments, including 53 labels in the test split. Treating applicable as the positive class, let TP, FP, TN, and FN denote true positives, false positives, true negatives, and false negatives relative to human labels. The metrics are

$$\begin{aligned} \text{Precision} &= \frac{\text{TP}}{\text{TP}+\text{FP}}, & \text{Recall} &= \frac{\text{TP}}{\text{TP}+\text{FN}}, \\ \text{F1} &= \frac{2\text{TP}}{2\text{TP}+\text{FP}+\text{FN}}, & \text{Specificity} &= \frac{\text{TN}}{\text{TN}+\text{FP}}, \\ \text{Accuracy} &= \frac{\text{TP}+\text{TN}}{\text{TP}+\text{FP}+\text{TN}+\text{FN}}. \end{aligned} \tag{17}$$

For pairwise evaluation, let $y_{ij} \in \{-1, 0, +1\}$ denote the human preference, with $+1$ favoring segment i, -1 favoring segment j, and zero denoting a tie. Let R_i and R_j be their rewards under the evaluated method. We define the valid pair set and pairwise accuracy as

$$\begin{aligned} \mathcal{P} &= \{(i,j) : y_{ij} \neq 0,\ R_i \neq R_j\}, \\ \text{PAcc} &= \frac{1}{|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} \mathbf{1}[\text{sign}(R_i - R_j) = y_{ij}]\,. \end{aligned} \tag{18}$$

Of the 115 human non-tie pairs, RaR-Equal produces 30 model ties and is evaluated on the remaining 85 pairs. Each of the other three methods is evaluated on all 115 pairs. The reported 95% confidence intervals use round-clustered bootstrap resampling.

Under the repeated-audit protocol in Appendix A.2, 77 of the 430 segments receive three evaluations, yielding 847 segment–criterion combinations. Table 3 reports applicability patterns and sample counts for noise estimation. Always applicable and always N/A denote agreement across all three calls. Switching denotes a mixture of applicable and N/A outputs. Combinations with at least two valid numerical scores support the within-segment variance estimate in Equation 7.

These counts cover the full offline audit set, not individual calibration windows.

Eligible sample counts range from 1 to 47 per criterion, with only one each for R6 and R9. Across all 847 combinations, 684 (80.76%) have identical applicability decisions across the three calls, including 567 that are always N/A. The 163 switching combinations account for 19.24% of all combinations and 58.21% of those with at least one applicable evaluation. Of the 202 combinations applicable on the first call, 30 have fewer than two valid scores. Conversely, 23 of the 195 eligible combinations are N/A on the first call. Thus, audit support varies across criteria, and the eligible audit set differs from the first-pass applicable set. We use the noise statistic for empirical calibration, with the sample-count and coverage requirements in Appendix A.2 governing weight adaptation.

Table 3: Offline repeated-audit statistics. Each criterion is evaluated on the same 77 audited segments. The three applicability-pattern columns partition these segments; the last column counts segments eligible for noise estimation. Criterion IDs follow Figure 1.

Criterion	Always applicable	Always N/A	Switching	At least two valid scores
R1	22	25	30	38
R2	29	22	26	47
R3	14	30	33	32
R4	3	65	9	7
R5	12	43	22	19
R6	0	75	2	1
R7	8	67	2	9
R8	11	50	16	19
R9	1	71	5	1
R10	4	66	7	5
R11	13	53	11	17
Total	117	567	163	195

A.4 Policy and Judge Prompts

We provide the policy and Judge prompts in English translation, followed by the complete Black Bear Guai evaluation configuration. Message fields, criterion identifiers, and action tokens retain their original names.

Policy dialogue format. The policy receives an alternating history of game images and action responses. The example below contains two historical exchanges and the current image. The final assistant marker opens the turn in which the policy generates the next action sequence. In the schematic action responses, `X` and `Y` stand for integer mouse displacements, and `K1` and `K2` stand for valid key tokens. The system prompt uses `[GAME NAME]` as the game-name placeholder.

Policy dialogue format

SYSTEM

`<|im_start|>system`

Role

You are an experienced player of `[GAME NAME]` who is proficient in keyboard-and-mouse control. Based on the current and historical game images and any optional user instruction, predict continuous keyboard-and-mouse actions for the next several frames.

Output format

`X Y k1 k2 k3...;X Y k1 k2 k3...;...`

Instructions

1. **Mouse displacement:** `X` is the horizontal displacement, positive to the right, and `Y` is the vertical displacement, positive downward. Both are integers.
2. **Key sequence:** `k1`, `k2`, `k3`, and so on are the keyboard or mouse keys pressed at that instant. A group can contain zero or more keys.
3. Separate `X`, `Y`, and keys with spaces.
4. Output only a correctly formatted plain string, without line breaks or quotation marks.

Key naming conventions

- Number keys `0--9`: lowercase English names, such as `zero`, `one`, and `nine`.

- Function keys `F1--F12`: initial-capital English names, such as `One`, `Two`, and `Twelve`.
- Mouse buttons: `LB` (left), `RB` (right), and `MB` (middle).
- Letter keys: uppercase letters, such as `W`, `A`, `S`, and `D`.
- Special keys: `Space`, `Shift`, `Ctrl`, `Alt`, `Tab`, `Esc`, `Caps`, `Left`, `Right`, `Up`, and `Down`.

```
<|im_end|>
```

HISTORY AND CURRENT INPUT

```
<|im_start|>user
<|vision_start|>{image_1}<|vision_end|><|im_end|>
<|im_start|>assistant
X Y K1 K2; X Y K1 K2; X Y K1 K2; X Y K1 K2<|im_end|>
<|im_start|>user
<|vision_start|>{image_2}<|vision_end|><|im_end|>
<|im_start|>assistant
X Y K1 K2; X Y K1 K2; X Y K1 K2; X Y K1 K2<|im_end|>
<|im_start|>user
<|vision_start|>{image_current}<|vision_end|><|im_end|>
<|im_start|>assistant
```

Judge dialogue format. The Judge uses a fixed System Prompt and a segment-specific User Message. The placeholder `{rubric_text}` contains the criteria and their score definitions. The complete Black Bear Guai criteria are provided in the JSON listing below. The assistant entry is an output-format schematic with placeholders, not an actual evaluation. An actual JSON response contains one entry for every supplied criterion, using a score of -1, 0, or 1 when applicable and null otherwise. In the original User Message, identifiers in `{rubric_id_list}` are separated by ideographic commas.

Judge dialogue format

SYSTEM

Role

You are a strict, conservative game-combat video Rubric Judge.

Task

You will receive `sample_id` (sample identifier), `target_interval` (the video interval to evaluate), `subject` (the evaluated subject), `rubric_bank` (the list of rubrics to use), and the corresponding video. Judge each rubric in the Rubric Bank separately.

Rubric Bank

`{rubric_text}`

Judgment procedure

For each rubric:

1. Find the first complete candidate event corresponding to the rubric within `target_interval`.
2. Determine whether the trigger, subject response, and outcome are sufficiently clear.
3. If there is no complete event or key evidence is not visible, output `applicable=false` and `score=null`.
4. If the event is fully visible, select the score strictly according to that rubric's `scores` definitions. Do not invent scoring criteria or intermediate scores.

5. Boundary rule: an event that begins within `target_interval` may be scored if its outcome is visible in the following context. An event that began before the target interval does not belong to the current interval.

Scoring rules

Global score definition: N/A means there is no complete corresponding situation that can be judged, or the key outcome is not visible.

1. When `applicable=false`, `score` must be `null`. Do not encode N/A as 0.
2. When `applicable=true`, `score` must be -1, 0, or 1, strictly following the rubric-specific definitions.
3. Judge only from visible video evidence. If evidence is unclear, use `applicable=false`.
4. Write a brief, verifiable visual basis in `reason`. Do not invent hidden states.
5. Every `rubric_id` must match the Rubric Bank exactly. Do not omit, merge, or add IDs.

Output format

Output only plain JSON, without Markdown or other text. Return one entry per rubric.

USER

```
sample_id: {chunk_uid}
target_interval: [{t0:.1f}s, {t1:.1f}s]
```
(Video-relative time; evaluate only behavior within this interval.)
```
subject: {subject}
rubric_bank: {n} items ({rubric_id_list})
```
The corresponding video is provided below.

ASSISTANT

```
{"rubrics": [
{"rubric_id": "{rubric_id}", "applicable": {true_or_false},
"score": {score_or_null}, "reason": "{visual_evidence}"}
]}
```

Black Bear Guai evaluation criteria. The complete Black Bear Guai configuration is reproduced below in English translation, preserving the JSON structure, criterion identifiers, score values, and base weights. The Judge prompt uses the applicability conditions and score definitions. Base weights are applied after scoring during reward aggregation.

Black Bear Guai evaluation criteria (JSON)


```
{
  "schema_version": "black-bear",
  "score_scale": {
    "N/A": "No complete corresponding situation can be judged, or the key
      outcome is not visible.",
    "-1": "Clear failure",
    "0": "The outcome is clear but only partially successful",
    "1": "Clear success"
  },
  "rubrics": [
    {
      "rubric_id": "R.effective_attack",
      "rubric": "If the player completes a normal attack sequence while the
        Boss is neither in a natural opening nor controlled, and the player
```

```
        is not transformed, evaluate whether the sequence produces effective
        hits without immediate punishment; otherwise, N/A.",
      "scores": {
        "-1": "The attack misses entirely, or the Boss clearly lands a heavy
          hit, knocks the player back, or knocks the player down before the
          attack ends.",
        "0": "At least one clear hit lands, but there are also missed
          swings, minor interruptions, or some damage taken.",
        "1": "Clear effective hits land, and the player takes no clear
          damage before the attack ends."
      },
      "base_weight": 1.0
    },
    {
      "rubric_id": "R.incoming_attack_response",
      "rubric": "If a normal Boss attack sequence is fully visible from
        wind-up to outcome, evaluate whether the player avoids its main
        damage. Black Wind rushes and ground flames are excluded; otherwise,
        N/A.",
      "scores": {
        "-1": "The player fails to respond effectively and clearly loses HP,
          staggers, is knocked back or airborne, or falls down.",
        "0": "The player attempts evasion or escape but is clearly hit by
          part of the attack without being knocked down.",
        "1": "The player avoids the main damage throughout the sequence and
          remains able to act after the attack ends."
      },
      "base_weight": 1.5
    },
    {
      "rubric_id": "R.opening_exploitation",
      "rubric": "If the Boss exhibits a complete natural recovery window
        after a combo, pounce, landing, rush, or missed attack, evaluate
        whether the player converts it into effective damage. Immobilization
        windows and windows used for healing or extinguishing flames are
        excluded; otherwise, N/A.",
      "scores": {
        "-1": "A full opening occurs, but the player lands no clear hit or
          clearly misses.",
        "0": "One clear hit lands, but entry is late, subsequent attacks
          miss, or the player takes some counterattack damage.",
        "1": "The player promptly lands multiple clear hits, or one
          high-value hit with clear damage or stagger evidence, without being
          hit during the window."
      },
      "base_weight": 1.2
    },
    {
      "rubric_id": "R.control_window_utilization",
      "rubric": "If Black Bear Guai is clearly controlled by Immobilize, a
        spirit skill, or another player ability, evaluate whether the player
        deals effective damage before control ends; otherwise, N/A.",
      "scores": {
        "-1": "No clear hit lands during control, or the player only swings
          and misses.",
        "0": "One clear hit lands, but entry is late or the remaining
          attacks clearly miss.",
        "1": "The player promptly lands multiple clear hits, or one
          high-value hit with clear damage or stagger evidence."
      },
      "base_weight": 1.0
    },
    {
```

```
    "rubric_id": "R.post_hit_stabilization",
    "rubric": "If the player is clearly hit, knocked down, or staggered
      and then regains control, evaluate whether they escape continued
      pressure before the next complete threat ends; otherwise, N/A.",
    "scores": {
      "-1": "After regaining control, the player immediately attacks
        recklessly or remains in danger and takes clear damage again.",
      "0": "After regaining control, the player moves or defends but still
        takes one lighter follow-up hit without being knocked down again.",
      "1": "The player escapes pressure through rolling, sprinting, a
        movement skill, distancing, or a safe counterattack, without taking
        further damage."
    },
    "base_weight": 1.2
  },
  {
    "rubric_id": "R.transformation_window_utilization",
    "rubric": "If at least three seconds of the controllable phase of a
      transformation are fully visible, evaluate whether the visible
      transformation time produces effective attacks without obvious waste;
      otherwise, N/A.",
    "scores": {
      "-1": "No clear hit lands, with repeated missed swings, prolonged
        separation from the target, or sustained inactivity without a
        threat.",
      "0": "Some hits land, but most attacks miss, the player remains away
        from the target for a long time, or severe interruptions occur.",
      "1": "Multiple clear hits land, or one high-value hit with clear
        damage or stagger evidence, without obvious waste."
    },
    "base_weight": 0.8
  },
  {
    "rubric_id": "R.black_wind_rush_response",
    "rubric": "If a Black Wind or flaming wind rush is fully visible from
      approach to contact outcome, evaluate whether the player evades it or
      clearly interrupts it before impact; otherwise, N/A.",
    "scores": {
      "-1": "The rush clearly hits, causing HP loss, stagger, an airborne
        knockback, or a knockdown.",
      "0": "The player attempts evasion or interruption but still takes
        clear partial damage.",
      "1": "The player clearly avoids the rush or successfully interrupts
        the Black Wind form, with no visible hit caused by this event."
    },
    "base_weight": 1.4
  },
  {
    "rubric_id": "R.fire_hazard_response",
    "rubric": "If a ground slam with fire, a ground fire pillar, or a
      persistent ground-fire event is fully visible, evaluate whether the
      player avoids the main burst and leaves the fire hazard. Ordinary
      flaming melee attacks are excluded; otherwise, N/A.",
    "scores": {
      "-1": "The slam, fire pillar, or ground fire clearly hits, causing
        evident HP loss, stagger, a knockdown, or burning.",
      "0": "The player avoids the main slam but then touches a fire pillar
        or ground fire and takes lighter damage.",
      "1": "The player avoids the main damage, remains able to act at the
        end of the event, and is outside visible fire hazards."
    },
    "base_weight": 1.3
  },
```

```
    {
      "rubric_id": "R.burning_status_recovery",
      "rubric": "If the player enters a clear burning state and the
        subsequent handling outcome is fully visible, evaluate whether
        ongoing burning is promptly stopped or substantially reduced;
        otherwise, N/A.",
      "scores": {
        "-1": "No effective action is taken, and flames persist with
          continued HP loss or further fire damage.",
        "0": "The player rolls, leaves the fire zone, uses medicine, or uses
          a fire-protection measure, but flames remain clearly visible at the
          end of observation without another severe hit.",
        "1": "Flames on the player or ongoing HP loss clearly stop during
          the observation period."
      },
      "base_weight": 1.2
    },
    {
      "rubric_id": "R.safe_gourd_use",
      "rubric": "If a gourd use is visible from the start of drinking
        through recovery of action, evaluate whether healing is completed
        safely. Do not infer that drinking is required solely from low HP;
        otherwise, N/A.",
      "scores": {
        "-1": "Drinking is interrupted, or the player takes clear damage
          before regaining the ability to act.",
        "0": "Drinking completes and restores HP, but the next Boss attack
          hits within 1.5 seconds after action recovery.",
        "1": "Drinking completes and clearly restores HP, with no hit from
          the start of drinking through 1.5 seconds after action recovery."
      },
      "base_weight": 1.3
    },
    {
      "rubric_id": "R.ability_use_effectiveness",
      "rubric": "If the player fully casts a spell, spirit skill, or
        movement skill, evaluate whether the period from activation to the
        first direct effect yields clear control, damage, evasion, or escape.
        Subsequent attacks are excluded; otherwise, N/A.",
      "scores": {
        "-1": "The skill clearly misses, targets an unaffected state,
          produces no visible benefit, or is interrupted before taking
          effect.",
        "0": "The skill has a partial effect, but the player takes some
          damage during activation or the direct effect ends quickly.",
        "1": "The skill directly produces clear control, effective damage,
          successful evasion, or safe escape, without clear punishment during
          activation."
      },
      "base_weight": 0.8
    }
  ]
}
```

A.5 Baseline Rewards and Weight Updates

Outcome Reward. The Outcome Reward baseline uses the final episode outcome as its reward: $+1$ for victory and -1 for defeat. This outcome is assigned as the terminal reward of every chunk from that episode.

Direct VLM Reward. The Direct VLM Reward baseline directly assigns an overall score in $[-1, 1]$ to each video segment and uses this score as the terminal reward of the corresponding chunk.

POW3R-Game. This baseline adapts the variance-driven weight calibration mechanism of POW3R (Tyagi et al., 2026) to game segments. It constructs the update signal from the batch variance of first-pass applicable scores and updates weights through relative-signal normalization, multiplier clipping, and EMA smoothing. The Black Bear Guai experiments use 11 evaluation criteria. Criteria with at least two first-pass applicable scores and coverage of at least 5% participate in calibration. Their update signals are normalized by their arithmetic mean. Update batches, multiplier bounds, and multiplier EMA settings match those of Ours in Appendix A.2.

POW3R-Game + Mean. This online ablation on Black Bear Guai retains the weight-update mechanism of POW3R-Game and matches Ours in interaction budget and repeated-audit protocol: each segment is evaluated once, and a randomly selected 20% receive two additional evaluations. Rewards use the mean of available valid scores for each criterion, retaining a single valid score when only one is available and excluding N/A. Weight-update statistics still use first-pass scores, without noise subtraction or an audit-count eligibility requirement. The update signal does not multiply variance by coverage; the minimum-coverage eligibility requirement is retained.

Component ablations. The variants without noise calibration or coverage modulation retain NOISERUB's minimum coverage threshold of 5%, minimum count of four eligible audited segments, and fallback to a target multiplier of one when eligibility conditions are not met. They differ from NOISERUB only in the corresponding term of Equation 9.

Shared training settings. All online post-training methods share the SFT initialization, interaction budget, within-segment reward discounting, and policy optimization settings in Appendix A.1. RaR-Static, POW3R-Game, and Ours use the same evaluation criteria, base weights, and judge configuration. Outcome Reward and Direct VLM Reward propagate their segment-terminal rewards to actions using Equation 13.

A.6 ADDITIONAL EVALUATION ON A SECOND GAME

Experimental settings. We compare SFT and NOISERUB on Yuange Puppet (Phase 3) and Qingyue Zhujiang in *Honor of Kings: World*. The SFT policy is trained on 8.3 hours of demonstrations from this game and serves as the initialization for NOISERUB post-training. Both methods use the same in-game character configuration and evaluation conditions. The Yuange Puppet task starts directly from Phase 3 and evaluates this phase only.

Apart from the game environment, SFT data, and task-specific evaluation criteria, training hyperparameters, judge configuration, and weight-calibration settings follow the *Black Myth: Wukong* experiments in Appendices A.1 and A.2. NOISERUB is trained separately for each Boss for 30 rounds with 200 seconds of interaction per round, totaling 6,000 seconds per Boss. SFT is evaluated without this additional reinforcement-learning stage.

Evaluation protocol. We evaluate each method on 20 episodes per Boss, using the round-30 checkpoint for NOISERUB. We report win rate and mean Boss HP reduction over all episodes, including failures. For Yuange Puppet (Phase 3), HP reduction is calculated relative to the initial HP of that phase using its remaining HP at the end of the episode.

Table 4: Additional evaluation in *Honor of Kings: World*. WR: win rate; HR: mean Boss HP reduction (both in %). Each method is evaluated on 20 episodes per Boss. HP reduction includes failed episodes. NOISERUB is evaluated after 30 training rounds. Higher is better.

	Yuange Puppet (Phase 3)		Qingyue Zhujiang	
Method	WR ↑	HR ↑	WR ↑	HR ↑
SFT	15	47.86	10	36.22
NOISERUB	**30**	**66.85**	**30**	**59.23**

Results. Table 4 shows higher win rates and mean HP reduction for NOISERUB than SFT on both tasks. On Yuange Puppet (Phase 3), win rate increases from 15% to 30%, and HP reduction increases from 47.86% to 66.85%. On Qingyue Zhujiang, win rate increases from 10% to 30%,

and HP reduction increases from 36.22% to 59.23%. These results provide additional evidence that NOISERUB post-training can improve an SFT policy in another game environment.