FIRE: FAILURE-INFORMED RECOVERY POST-TRAINING FOR REAL-TIME VISUAL COMBAT

Anonymous authors
Paper under double-blind review

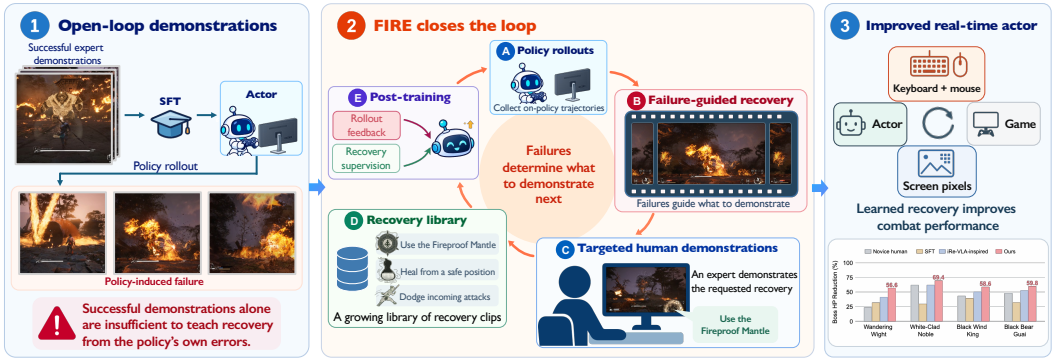


Figure 1: We propose *FIRE*, a closed-loop post-training framework that uses policy failures to determine what human recovery behaviors should be demonstrated next. Successful expert trajectories sparsely cover the recovery states induced by an imperfect policy. FIRE identifies recurring rollout failures, converts them into targeted recovery objectives, and incorporates the resulting demonstrations into subsequent policy updates. As the policy evolves, new failures redirect data collection toward its current weaknesses, forming a curriculum that improves real-time visual control.

ABSTRACT

Real-time visual combat exposes a mismatch between how human demonstrations are collected and where a learned policy needs supervision. Expert data are typically collected open loop and emphasize successful trajectories, leaving policy-induced failure states underrepresented. As the policy evolves, these deficiencies also shift, making additional generic demonstrations increasingly inefficient. We argue that post-training should instead close the loop between policy experience and human data acquisition, using rollout failures to determine *what should be demonstrated next*. We introduce *FIRE: Failure-Informed REcovery Post-Training*, a closed-loop framework that turns the policy's evolving failures into a curriculum for human supervision. Rather than collecting more generic expert play, FIRE uses recurring rollout failures to identify missing recovery behaviors, acquires demonstrations targeted to these deficiencies, and feeds them back into subsequent policy updates. A multimodal coach translates recurring failures into actionable recovery requests. The resulting demonstrations are organized in a recovery library and retrieved according to the encountered failure modes during subsequent supervised updates, while rollout feedback supports critic-free policy optimization and selective replay. Across seven *Black Myth: Wukong* bosses and two character configurations, FIRE improves normalized Boss HP Reduction over supervised fine-tuning (SFT) on all 14 boss–configuration pairs and Win Rate on 13. Under an equal additional demonstration-video budget, FIRE achieves higher final HP Reduction on all three controlled encounters than either additional successful demonstrations or Static recovery targeting only the initial SFT policy's failures. Supplementary videos are available on the anonymous project page.

1 INTRODUCTION

Vision-language game agents have progressed from language-conditioned navigation and interaction in 3D environments to screenshot-based control of commercial games (Raad et al., 2024; Tan

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.

et al., 2024). Real-time combat presents a particularly demanding setting: agents must continuously interpret rapidly changing scenes while executing precisely timed keyboard-and-mouse actions. A delayed dodge, an overlong key press, or poor camera control can quickly move the policy from a favorable state into one that requires recovery. Supervised fine-tuning (SFT) provides a natural starting point by teaching actors to imitate expert gameplay. However, demonstrations are typically collected *open loop*, independently of where the learned policy will later fail. This creates a mismatch between successful expert trajectories and the states visited by an imperfect policy. Once the actor deviates from expert behavior, it can enter poorly covered states and accumulate errors (Ross et al., 2011). Moreover, successful demonstrations naturally underrepresent recovery behaviors such as disengaging from a bad position, healing under pressure, restoring camera control, or safely re-engaging. The problem is therefore not simply to collect more demonstrations, but to collect the demonstrations that address the current policy's actual deficiencies.

Online interaction reveals these deficiencies. Reinforcement learning (RL) exposes the policy to states induced by its own actions, while interactive imitation learning can obtain corrective supervision from human experts (Kelly et al., 2019; Hoque et al., 2021). Yet scalar outcomes reveal only that a rollout was unsuccessful, not what behavior should have been demonstrated instead, while continuous expert intervention requires substantial human attention. This motivates a different use of on-policy experience: policy failures should not only drive policy updates, but also determine *what human data to collect next*.

We study this problem with a compact visual actor that maps game-screen observations and action history directly to keyboard-and-mouse controls. As shown in Figure 1, the game continues to evolve while actions are predicted and executed, so recovery often requires a temporally extended sequence rather than a single correction. For example, a mistimed dodge may require retreating, healing, repositioning, and waiting for a safe opportunity to re-engage. Our key insight is to treat recurring policy failures as signals for *data acquisition*. Rather than collecting additional expert play independently of the policy, we use rollout failures to identify missing recovery behaviors and direct a limited human recording budget toward them. As the policy changes, newly observed failures redirect subsequent recording, creating an adaptive curriculum of human supervision that follows the policy's evolving weaknesses.

Based on this insight, we introduce *FIRE: Failure-Informed REcovery Post-Training*, a closed-loop framework that connects policy experience, targeted human demonstrations, and online policy improvement. After each rollout round, FIRE identifies recurring failure patterns and converts them into concrete recovery objectives for human recording. The resulting demonstrations are stored in a recovery library and retrieved by boss and failure mode for subsequent supervised updates. In parallel, combat outcomes and multimodal planning assessments support critic-free policy optimization and selective replay. Policy experience therefore improves the actor in two complementary ways: directly through online learning and indirectly by determining which recovery behaviors should be demonstrated next. The multimodal coach is used only during post-training, while deployment retains a compact real-time actor. Across seven in-domain *Black Myth: Wukong* bosses and two character configurations, FIRE improves normalized Boss HP Reduction over SFT on all 14 boss–configuration pairs and Win Rate on 13. Under an equal additional demonstration-video budget, targeted recovery demonstrations achieve higher final HP Reduction than additional successful boss-fight demonstrations on all three controlled encounters. FIRE also achieves higher final HP Reduction than Static recovery based only on initial SFT failures on all three encounters.

Our contributions are threefold:

1. We introduce *FIRE*, a closed-loop post-training framework in which recurring policy failures determine what human recovery behaviors should be demonstrated next.
2. We develop a failure-informed recovery acquisition mechanism that converts rollout deficiencies into targeted recording objectives and integrates the resulting recovery supervision with critic-free online policy optimization and selective replay.
3. We evaluate FIRE across seven in-domain bosses under two character configurations, and show that adaptive recovery acquisition outperforms both additional successful gameplay and Static recovery under an equal demonstration-video budget.

2 Related Work

Native visual game agents. Video PreTraining (VPT) learns keyboard-and-mouse behavior from inverse-dynamics-labeled Minecraft videos (Baker et al., 2022). SIMA learns instruction-following control across 3D environments, while Cradle uses multimodal planning to produce controls from screenshots (Raad et al., 2024; Tan et al., 2024). VARP combines VLM action planning with retrieval from human gameplay trajectories in *Black Myth: Wukong* (Chen et al., 2024). Lumine combines pixel-based control with adaptive reasoning for long-horizon missions and zero-shot transfer across games (Tan et al., 2025). SIMA and BALROG evaluations highlight continuing challenges in visual decision-making and spatial control (Raad et al., 2024; Paglieri et al., 2025). CombatVLA targets action-role-playing combat with action-of-thought training and truncated inference that emits actions before completing reasoning (Chen et al., 2025b). FIRE targets the post-training of compact real-time actors, using policy failures to identify recovery behaviors underrepresented in successful demonstrations.

Learning from demonstrations and online experience. DQfD combines demonstration supervision with temporal-difference learning and prioritized replay, while AWAC and RLPD integrate prior data with online RL (Hester et al., 2018; Nair et al., 2020; Ball et al., 2023). For pretrained robotic VLA policies, iRe-VLA alternates RL and supervised learning to combine exploration with the stability of supervised updates (Guo et al., 2025). PER prioritizes transitions by temporal-difference error, and self-imitation reinforces high-return decisions (Schaul et al., 2015; Oh et al., 2018). DreamerV3 and object-centric OC-STORM instead learn dynamics and train on imagined experience (Hafner et al., 2023; Zhang et al., 2026). FIRE couples online policy improvement with failure-informed data acquisition, using policy experience to determine what human recovery behaviors should be demonstrated next.

Foundation-model feedback for policy learning. LIV and VLM-RM derive visual alignment or progress signals, while RL-VLM-F learns rewards from VLM-generated visual preferences (Ma et al., 2023; Rocamonde et al., 2024; Wang et al., 2024). MineDojo and RoboCLIP use video-language representations or reference videos for rewards, whereas Eureka generates reward code (Fan et al., 2022; Sontakke et al., 2023; Ma et al., 2024). VLM-Guided Experience Replay uses frozen-VLM scores to prioritize sub-trajectories (Sharony et al., 2026). FPC-VLA uses a supervisor to assess action viability and provide corrective strategies (Yang et al., 2026). Game-agent research also explores VLM reward annotation and offline return-conditioned learning (Schmid & Frosio, 2026b;a). FIRE uses VLM diagnosis to translate recurring failures into recovery objectives for human recording, while rollout assessments support policy optimization and replay.

Human guidance and recovery acquisition. DAgger aggregates expert actions at learner-visited states (Ross et al., 2011). HG-DAgger and intervention-weighted regression learn from human takeovers, while ThriftyDAgger allocates interventions using novelty and risk (Kelly et al., 2019; Mandlekar et al., 2020; Hoque et al., 2021). ReIL combines interventions with replay and RL; Human-Robot Copilot fine-tunes on takeover segments and original demonstrations (Parnichkun et al., 2022; Yan et al., 2026). ASkDAgger queries teachers about planned skills, and SOP integrates fleet experience with human interventions (Luijkx et al., 2025; Pan et al., 2026). HELP combines rollout segmentation with human interventions and recovery demonstrations for post-training (Zhai et al., 2026). FIRE aggregates recurring failures by boss and failure mode between rounds to request targeted human demonstrations. These demonstrations populate a recovery library for supervised updates, while new policy failures redirect subsequent recording and retrieval.

3 Recovery-Centric Online Post-Training

3.1 Problem Setting and Data Units

We study online post-training for a compact visual actor in native real-time visual combat. At deployment, the actor receives game-screen pixels and predicts keyboard-and-mouse controls conditioned on visual–action history. The game continues during perception and inference. The actor maintains a bounded visual–action context to support low-latency control, with context management and action timing specified in Appendix A.1.

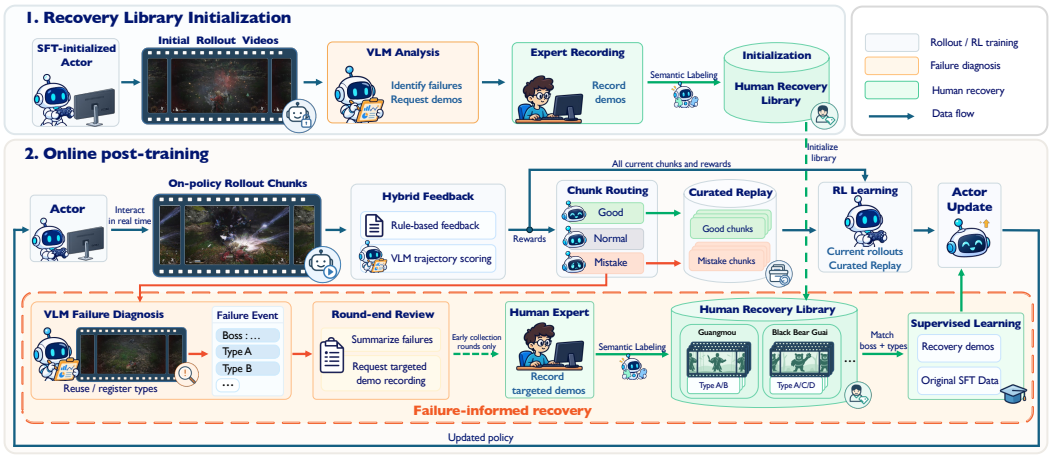


Figure 2: Overview of FIRE. Initial rollout failures guide human demonstrations that initialize the recovery library. During online post-training, the highlighted failure-informed recovery branch converts recurring failures into targeted recording requests at collection rounds and retrieves recovery demonstrations by boss and failure mode for supervised updates. Updated policies reveal new failures that redirect subsequent recording and retrieval, closing the loop. In parallel, hybrid feedback and curated replay support critic-free policy optimization.

An interaction turn n pairs a screen observation o_n with an output sequence A_n of keyboard-and-mouse action groups, with $A_n \sim \pi_\theta(\cdot \mid h_n)$ and $h_n = (o_{\le n}, A_{<n})$. Each supervised target contains four action groups. The client executes decoded groups sequentially, and we use t to index these action steps in the reward and optimization equations below. A complete boss encounter forms a rollout. A *trajectory chunk* contains at most 50 image–action turns and is the common training unit for original demonstrations, recovery demonstrations, on-policy rollouts, and replay. One rollout may yield multiple chunks.

3.2 FIRE Framework Overview

FIRE closes the loop between policy experience and human recovery acquisition (Figure 2). An SFT actor π_{θ_0} trained on original demonstrations D_0 first produces fixed-actor rollouts for VLM review. Diagnosed failure modes guide human demonstrations that initialize the recovery library D_{rec}. These rollouts are excluded from policy updating and curated replay.

During online post-training, failure-informed recovery turns recurring policy failures into targeted human supervision. Hybrid feedback identifies mistake chunks, whose diagnosis yields failure-mode labels and recovery objectives. At collection rounds, accumulated failures guide targeted recording requests. The resulting demonstrations expand the recovery library and are retrieved by boss and failure mode for supervised updates. In parallel, current rollouts and curated replay support critic-free policy optimization. As the actor evolves, newly observed failures redirect recording and retrieval toward its current recovery needs. Rollout diagnosis and human recording remain outside the real-time interaction loop. Section 3.6 gives the collection schedule, and Algorithm 1 in Appendix A.3 specifies the full procedure.

3.3 Hybrid Trajectory Feedback and Routing

Hybrid feedback provides learning signals for current rollouts and identifies mistake chunks for the failure-informed recovery loop. Rule rewards reflect HP changes and terminal combat events. They are discounted and clipped within each chunk once per interaction turn, yielding R_t^{rule} shared by its four action groups. A VLM reviews each chunk's 5 Hz video with player and boss HP overlaid, assessing tactical choices and temporal coordination with a planning score $v_j \in [0, 1]$. This score and boss-damage progress define a calibrated contribution R_t^{vlm} shared by the chunk's action steps. For current rollouts, the hybrid return at action step t is

$$R_t^{\text{hyb}} = R_t^{\text{rule}} + R_t^{\text{vlm}}. \tag{1}$$

Appendix A.4 details rule computation, trajectory-score calibration, and chunk-to-step assignment.

We use two fixed routing thresholds, $\eta_{\text{mistake}} < \eta_{\text{good}}$. For a trajectory chunk c, the routing rule is

$$q(c) = \begin{cases} \textit{mistake}, & \min_{t \in c} R_t^{\text{hyb}} < \eta_{\text{mistake}}, \\ \textit{good}, & \min_{t \in c} R_t^{\text{hyb}} \geq \eta_{\text{mistake}} \text{ and } \exists\, t \in c : R_t^{\text{hyb}} > \eta_{\text{good}}, \\ \textit{normal}, & \text{otherwise.} \end{cases} \tag{2}$$

All current chunks remain available for policy optimization. Routing considers every action step, saving good and mistake chunks for replay but excluding normal chunks. Mistake chunks also undergo failure diagnosis for targeted recovery acquisition (Section 3.4). Appendix A.4 gives the thresholds.

3.4 Failure-Informed Recovery Acquisition

Failure diagnosis turns policy mistakes into concrete recovery objectives for human demonstration. A VLM reviews each mistake chunk selected by Equation 2, specifying a visual trigger, failure description, desired response, and human recording request. Each chunk is one event e with semantic failure-mode labels $\mathcal{L}_e$. Diagnosis reuses registered labels whose definitions cover the observed combat situation and registers a new situation-specific label and definition otherwise.

At collection rounds, events are aggregated by boss and failure mode to determine what experts should demonstrate next. Modes without human recovery data receive first priority. Other modes are ranked by failure frequency divided by existing demonstration duration for the same boss and mode. Experts follow the resulting requests, create the relevant combat conditions through gameplay, and record the desired recovery behaviors.

Shared failure-mode labels link diagnosed events to recovery demonstrations. The VLM labels each human segment by matching its demonstrated recovery behaviors to registered failure-mode definitions. Accepted segments store these labels and contextual metadata, and may address multiple modes. During supervised cross-entropy (CE) sampling, recovery demonstrations are preferentially retrieved by matching the boss and failure-mode labels of diagnosed events. Appendix A.6 gives the eligibility and retrieval rules.

The collection budget is reported in Section 4.1, with the CE sampling mixture specified in Appendix A.8.

3.5 Policy Optimization with Recovery Supervision

FIRE updates the actor through reinforcement learning from policy experience and supervised learning from original and failure-guided recovery demonstrations.

Policy-generated chunks use a critic-free, token-level CISPO-style REINFORCE objective (Chen et al., 2025a). Let $\widetilde{R}_t$ be the hybrid return for current rollouts or, for replay, the return recomputed from stored turn scores and filtered by the reference policy (Appendix A.5). Each turn shares this final signal across its four action groups. Let $\widetilde{\mathcal{M}}$ contain valid keyboard and mouse-button target tokens with nonzero final signals, excluding mouse-displacement tokens. For token y_i at step $t(i)$, write $\pi_\theta(y_i) = \pi_\theta(y_i \mid \xi_i)$, where ξ_i is the preceding visual–dialogue context, shared with the reference policy. Define

$$\rho_i = \exp(\text{clamp}(\log \pi_\theta(y_i) - \log \pi_{\text{ref}}(y_i), \ell, u)), \tag{3}$$

where π_{ref} is a snapshot held fixed within each 200-chunk training stage and shared by current rollouts and curated replay. The RL objective is

$$\mathcal{L}_{\text{RL}} = -\frac{1}{\max(1, |\widetilde{\mathcal{M}}|)} \sum_{i \in \widetilde{\mathcal{M}}} \text{stopgrad}[\min(\rho_i, \bar{\rho})]\, \widetilde{R}_{t(i)} \log \pi_\theta(y_i). \tag{4}$$

The loss averages over retained tokens and is zero when none remain. The log-ratio is clamped to $[\ell, u]$ before exponentiation, and the detached weight is capped at $\bar{\rho}$. Appendix A.5 gives the constants and reference refresh procedure.

Recovery demonstrations from Section 3.4 provide supervised targets for the actor's diagnosed failure modes. Together with original demonstrations, they form

$$\mathcal{D}_{\mathrm{CE}} = D_0 \cup D_{\mathrm{rec}}. \tag{5}$$

For a demonstration chunk $(x, y) \sim \mathcal{D}_{\mathrm{CE}}$, the supervised objective is

$$\mathcal{L}_{\mathrm{CE}} = \lambda_{\mathrm{key}} \mathcal{L}_{\mathrm{key}} + \lambda_{\mathrm{mouse}} \mathcal{L}_{\mathrm{mouse}}. \tag{6}$$

$\mathcal{L}_{\mathrm{key}}$ is the teacher-forced cross-entropy over keyboard and mouse-button action targets, whereas $\mathcal{L}_{\mathrm{mouse}}$ covers mouse x/y displacement targets. Both use the demonstrations' own observation–action sequences.

Each training stage mixes current-rollout, curated-replay, and demonstration chunks at a target ratio of 1:1:1. Current rollouts and replay use $\mathcal{L}_{\mathrm{RL}}$, while demonstrations use $\mathcal{L}_{\mathrm{CE}}$. Replay chunks are sampled uniformly from the corresponding boss's pool.

3.6 Training Procedure and Deployment

Training procedure. FIRE runs ten online rounds, each with seven stages of 200 mixed chunks. The actor is updated throughout each stage, and its current weights are then copied to the reference policy. At the end of Rounds 1–5, failure events accumulated across all seven stages guide targeted human recovery recording for subsequent rounds. Rounds 6–10 reuse the library without further recording or new failure-mode registrations. Algorithm 1 in Appendix A.3 gives the full procedure.

Deployment. The Qwen3.5-2B actor autoregressively generates keyboard-and-mouse action groups from screen observations and bounded visual–action history. Decoded groups are streamed to the game client for sequential execution with deterministic action post-processing while the game continues running. The multimodal coach and human recovery collection are confined to post-training. Appendix A.1 details the action format, context management, and execution timing.

4 Experiments

4.1 Experimental Setup

We evaluate FIRE on seven in-domain and four held-out later-chapter *Black Myth: Wukong* bosses, and on all 13 practical tasks reported by CombatVLA across *Black Myth: Wukong* and *Sekiro: Shadows Die Twice* (Chen et al., 2025b). Training uses native Chapter-1 attributes attainable through standard gameplay. For the in-domain bosses, the same trained checkpoint is evaluated under native attributes and controlled Player Attack $= 50$, Player Defense $= 600$.

All post-training methods share the SFT initialization π_{θ_0}, trained on approximately 75 hours of original demonstrations. Each RL-based method uses ten online rounds totaling approximately 770 minutes of regular rollout. FIRE initializes its recovery library from failure analysis of 35 fixed-actor rollouts, five per training boss, adding approximately 90 minutes of interaction before updating. Human recovery data comprise approximately 84 minutes of demonstrations: 14 minutes for initialization and 70 minutes collected after Rounds 1–5. Appendix A.8 details the training and collection schedule.

Alongside SFT, we compare three RL baselines with curated replay but no CE: Outcome-reward RL, Rule reward only, and Hybrid-reward RL. They use terminal win/loss, rule-based, and hybrid feedback, respectively, and omit the CE third of FIRE's training chunks. The iRe-VLA-inspired baseline (Guo et al., 2025) interleaves hybrid-reward RL and SL at a 1:2 ratio, matching FIRE's total training-chunk budget. Its supervision combines original demonstrations and terminal-success online trajectories. Recovery controls use additional successful demonstrations (w/o Human Recovery Library) or recovery data addressing only the initial SFT policy's failures (Static recovery). GPT-5.6 Luna and Gemini-3.7 Flash serve as direct gameplay actors. Appendix A.7 provides the baseline implementations.

Post-training results use the final Round-10 checkpoint without selection on test-task performance. Each model–task pair is evaluated over 20 independent episodes. Five initially game-naive players provide the human reference, each completing four episodes per setting after watching a corresponding boss-fight video. All actors receive game-screen pixels only. We report Normalized Boss HP

Table 1: Performance on seven in-domain bosses under native Chapter-1 attributes and Player Attack $= 50$, Player Defense $= 600$. Cells report Normalized Boss HP Reduction / Win Rate (%) over 20 episodes. Bold marks the best model result per metric within each panel, including ties; humans provide a reference.

Method	Guangzhi	Wandering Wight	Lingxuzi	Guangmou	White-Clad Noble	Black Wind King	Black Bear Guai
Panel A: Native Chapter-1 state							
Novice human players	57.89 / 5	23.51 / 0	64.50 / 20	76.81 / 30	61.74 / 15	43.15 / 5	47.84 / 10
Gemini direct gameplay	4.70 / 0	3.04 / 0	16.48 / 0	10.79 / 0	4.71 / 0	2.96 / 0	2.65 / 0
GPT direct gameplay	7.97 / 0	6.01 / 0	13.51 / 0	6.37 / 0	6.15 / 0	4.16 / 0	3.29 / 0
SFT	65.88 / 20	31.71 / 0	72.73 / 30	64.34 / 25	29.50 / 0	39.10 / 0	31.83 / 0
Outcome-reward RL	8.27 / 0	0.00 / 0	9.68 / 0	7.63 / 0	6.14 / 0	4.48 / 0	7.19 / 0
Rule reward only	67.30 / 20	29.32 / 0	73.89 / 30	68.67 / 30	46.12 / 0	33.92 / 0	48.11 / 5
Hybrid-reward RL	68.15 / 25	30.17 / 0	71.52 / **35**	67.34 / 30	53.48 / 0	40.85 / 0	53.07 / 5
iRe-VLA-inspired	68.70 / 30	41.01 / 0	**74.17 / 35**	70.81 / 30	61.78 / 15	50.25 / 0	52.42 / 10
Ours	**71.01 / 35**	**56.59 / 5**	73.10 / 30	**73.85 / 35**	**69.43 / 20**	**58.57 / 10**	**59.76 / 20**
Panel B: Player Attack $= 50$**, Player Defense** $= 600$							
Novice human players	83.47 / 50	72.96 / 20	90.62 / 65	93.95 / 70	88.29 / 50	66.02 / 20	70.30 / 25
Gemini direct gameplay	16.01 / 0	8.52 / 0	35.04 / 0	19.59 / 0	7.58 / 0	9.91 / 0	4.96 / 0
GPT direct gameplay	22.94 / 0	10.48 / 0	27.12 / 0	16.56 / 0	6.85 / 0	11.27 / 0	7.92 / 0
SFT	87.73 / 55	68.54 / 10	93.19 / 75	81.48 / 50	74.24 / 30	57.76 / 15	57.31 / 20
Outcome-reward RL	17.30 / 0	12.95 / 0	22.17 / 0	18.06 / 0	11.45 / 0	13.35 / 0	22.05 / 0
Rule reward only	94.28 / **80**	69.22 / 15	93.12 / 75	81.48 / 60	77.36 / 30	52.08 / 10	62.76 / 20
Hybrid-reward RL	93.51 / 75	70.45 / 20	92.47 / 75	82.65 / 65	77.27 / 35	55.14 / 15	64.83 / 25
iRe-VLA-inspired	90.19 / 70	74.05 / 30	91.44 / 75	83.06 / 65	80.49 / 35	59.84 / 20	65.24 / 25
Ours	**95.53 / 80**	**80.68 / 40**	**96.62 / 85**	**92.17 / 85**	**87.05 / 50**	**70.95 / 35**	**75.07 / 40**

Reduction as the primary continuous metric and Win Rate as the terminal metric, boss by boss. Appendix A.10 gives the metric formulas, evaluation configurations, validity checks, and human protocol. Held-out results and gameplay sequences appear in Appendices A.14 and A.15.

4.2 Main Results on In-Domain Bosses

FIRE improves HP Reduction over SFT on all 14 boss–configuration pairs and Win Rate on 13, with one tie (Table 1). FIRE leads model baselines in both metrics on all seven 50/600 bosses, including Win Rate ties. Both direct VLM actors obtain 0% Win Rate under the API control protocol (Appendix A.7).

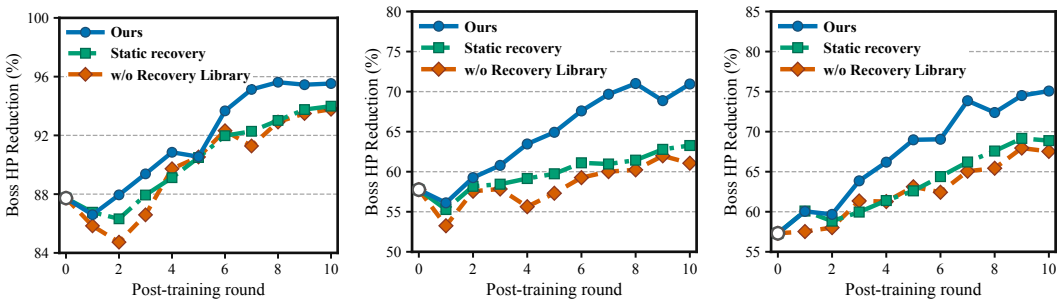
Outcome-reward RL yields no wins and lower HP Reduction than SFT throughout, indicating that sparse terminal feedback alone is insufficient here. Rule-based and hybrid feedback differ by at most 2.37 percentage points in HP Reduction on the first four bosses. Relative to rule-only feedback, hybrid feedback improves native-state HP Reduction by 4.96–7.36 points on White-Clad Noble, Black Wind King, and Black Bear Guai, consistent with its value in encounters with richer attack patterns. Under 50/600, HP differences narrow, but all three gain 5 percentage points in Win Rate.

FIRE adds CE updates on original and targeted recovery demonstrations, improving HP Reduction over Hybrid-reward RL on all 14 settings and Win Rate on 13. On the latter three bosses, gains across both configurations are 6.69–17.72 and 10–20 percentage points, respectively. Together with the matched-duration controls (Section 4.3) and recovery cases (Section 4.5), these results support using targeted human supervision to learn recovery behaviors underrepresented in open-loop successful demonstrations.

FIRE improves both metrics over iRe-VLA-inspired on 13 of 14 settings, with 15-point Win Rate gains on each of the latter three bosses under 50/600. This baseline omits curated replay and targeted recovery acquisition. The gains support FIRE's combined framework.

4.3 Effect of Failure-Informed Recovery

The recovery controls replace FIRE's adaptive acquisition with either additional successful demonstrations (w/o Human Recovery Library) or uniform recording for each boss's initial SFT recovery needs (Static recovery). All three retain hybrid feedback, curated replay, RL, and CE, with the same staged data availability. FIRE and Static recovery also share the retrieval procedure. Figure 3



(a) Guangzhi (b) Black Wind King (c) Black Bear Guai

Figure 3: Learning curves with 84 minutes of additional demonstrations per method. Round 0 is the shared SFT initialization; Rounds 1–10 are post-training checkpoints. Points show mean Boss HP Reduction over 20 episodes under 50/600; vertical ranges are boss-specific.

contrasts the simpler Guangzhi with Black Wind King and Black Bear Guai to test whether gains increase with attack-pattern complexity. Evaluation uses Player Attack $= 50$, Player Defense $= 600$ to increase combat tolerance and reduce variability from early deaths.

FIRE reaches higher HP Reduction earlier (Appendix A.9). At Round 10, its gains over additional successful demonstrations are 9.89 and 7.54 percentage points on Black Wind King and Black Bear Guai, compared with 1.75 on Guangzhi. Static recovery also exceeds that control, but FIRE further improves over Static recovery by 7.68, 6.19, and 1.54 points, respectively. These comparisons support adapting recovery acquisition to evolving policy failures, with larger gains on the harder encounters.

4.4 Comparison with Reported CombatVLA Results

Table 2 compares FIRE with published CombatVLA results (Chen et al., 2025b). For Wandering Wight and Guangzhi, Player Attack $=$ 100 and Player Defense $= 600$ match the reported attributes; other tasks retain their respective configurations. CombatVLA evaluates 10 trials per task, pausing the game during inference. FIRE evaluates 20 episodes with continuous game execution.

On all ten *Black Myth: Wukong* tasks, FIRE matches or exceeds the reported Win Rate. The largest gaps are on Wandering Wight (95% versus 10%) and Guangzhi (100% versus 10%). Without training on *Sekiro*, FIRE ties on Katana, scores higher on Hassou Stance, and lower on Shigenori Yamauchi. Appendix A.10 specifies the control mapping.

Table 2: Win Rate (%) on CombatVLA's 13 practical tasks. Best results are bold, including ties.

Task	CombatVLA	**Ours**
Black Myth: Wukong		
WolfScout	**100**	**100**
WolfStalwart	**100**	**100**
WolfSwornsword	**100**	**100**
WolfSoldier	**100**	**100**
Croaky	**100**	**100**
Crow Diviner	80	**100**
Bandit Chief	70	**100**
Bullguard	60	**75**
Wandering Wight	10	**95**
Guangzhi	10	**100**
Sekiro: Shadows Die Twice		
Katana	**90**	**90**
Hassou Stance	70	**80**
Shigenori Yamauchi	**60**	30

4.5 Recovery Behavior Analysis

We assess three phase-specific responses using SFT and the Round-10 checkpoints of the two recovery controls and FIRE. At 50/600, we evaluate 20 episodes per method and boss, score only the first occurrence of the relevant situation, and exclude episodes with unavailable required skills. Table 3 reports attainment among eligible episodes. The criteria cover transformation during Guangmou's snake phase, retreat to the arena edge followed by a heavy counterattack after Black Wind King's landing, and Fireproof Mantle use with survival through Black Bear Guai's wind form. Appendix A.13 specifies the scoring rules.

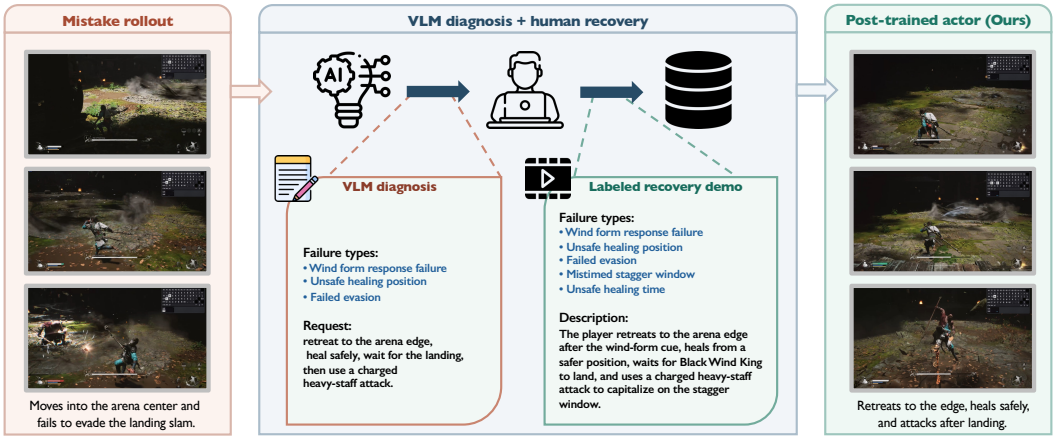


Figure 4: Failure-informed recovery on Black Wind King. Left: a policy failure during wind form. Center: failure diagnosis, the resulting human recording request, and an annotated recovery demonstration. Right: the post-trained actor's response.

FIRE has the highest observed attainment rate in all three cases. Only FIRE received demonstrations specifically recorded for the Guangmou transformation and Black Wind King counterattack. These responses occur in 31.3% and 35.0% of eligible FIRE episodes, versus at most 6.7% and 5.0% for the other methods. Both Static recovery and FIRE include Fireproof Mantle demonstrations, reaching 25.0% and 27.8% on the joint use-and-survival criterion, compared with 13.3% for additional successful demonstrations. These comparisons support targeted supervision for phase-dependent behaviors undercovered by open-loop successful play.

Table 3: Phase-specific recovery behaviors at 50/600. Cells report episodes meeting the criterion / eligible episodes (percentage), out of 20 episodes per method and boss. Bold marks the highest observed rate for each criterion.

Method	Guangmou Transformation	Black Wind King Counterattack	Black Bear Guai Protection + survival
SFT	0/14 (0.0)	0/19 (0.0)	1/13 (7.7)
w/o Human Recovery Library	1/15 (6.7)	0/19 (0.0)	2/15 (13.3)
Static recovery	1/15 (6.7)	1/20 (5.0)	4/16 (25.0)
FIRE	**5/16 (31.3)**	**7/20 (35.0)**	**5/18 (27.8)**

Figure 4 illustrates the Black Wind King response. The failed rollout already uses the gourd but does not evade the landing slam. The requested demonstration combines retreat, safer healing, waiting, and a charged counterattack. The post-trained actor performs this sequence, linking existing actions to the recovery situation in which they are needed. Appendix A.16 presents further examples for Guangmou and Black Bear Guai.

5 Conclusion

We introduced FIRE, a closed-loop post-training framework in which recurring policy failures guide the collection and reuse of human recovery demonstrations. Across seven *Black Myth: Wukong* bosses, FIRE improves HP Reduction over SFT on all 14 boss–configuration pairs and Win Rate on 13. With matched demonstration-video duration, FIRE reaches higher HP Reduction earlier than additional successful gameplay, with larger final gains on harder bosses. It also exceeds Static recovery based on initial SFT failures on all three controlled encounters under shared retrieval and staged data availability. These results support adapting human supervision to evolving policy failures, helping compact actors learn state-dependent recovery and re-engagement behaviors undercovered by open-loop demonstrations.

AI USE STATEMENT

We used generative AI tools for language polishing of the manuscript. The authors developed the research ideas and designed the experiments. Experimental data were recorded and reviewed by human researchers. We did not use AI for mathematical derivations or for organizing, analyzing, or interpreting experimental results. The authors reviewed all AI-assisted edits for accuracy and consistency with the intended meaning. All analyses, editorial decisions, and conclusions were made by the authors, who take full responsibility for the final content of this work.

ETHICS STATEMENT

All novice players, demonstration experts, and human annotators provided informed consent to participate in the study. Their participation involved gameplay evaluation, recovery demonstration recording, or annotation of gameplay clips. Formal institutional ethics approval was not sought.

REPRODUCIBILITY STATEMENT

The appendix provides the training procedure and pseudocode, model configurations, reward and replay rules, VLM prompts, data collection procedures, and evaluation protocols to support independent implementation of the method. We plan to release model weights following acceptance, subject to internal review and approval. Preparing the weights for public release may require additional time.

REFERENCES

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

Philip J Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. Efficient online reinforcement learning with offline data. In *International Conference on Machine Learning*, pp. 1577–1594. PMLR, 2023. 3

Aili Chen, Aonian Li, Bangwei Gong, Binyang Jiang, Bo Fei, Bo Yang, Boji Shan, Changqing Yu, Chao Wang, Cheng Zhu, et al. Minimax-m1: Scaling test-time compute efficiently with lightning attention. *arXiv preprint arXiv:2506.13585*, 2025a. 5

Peng Chen, Pi Bu, Jun Song, Yuan Gao, and Bo Zheng. Can vlms play action role-playing games? take black myth wukong as a study case. *arXiv preprint arXiv:2409.12889*, 2024. 3

Peng Chen, Pi Bu, Yingyao Wang, Xinyi Wang, Ziming Wang, Jie Guo, Yingxiu Zhao, Qi Zhu, Jun Song, Siran Yang, et al. Combatvla: An efficient vision-language-action model for combat tasks in 3d action role-playing games. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10919–10928. IEEE, 2025b. 3, 6, 8

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. 3

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, 2025. 3, 6

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023. 3

Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Ian Osband, et al. Deep q-learning from demonstrations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018. 3

Ryan Hoque, Ashwin Balakrishna, Ellen Novoseller, Albert Wilcox, Daniel S Brown, and Ken Goldberg. Thriftydagger: Budget-aware novelty and risk gating for interactive imitation learning. *arXiv preprint arXiv:2109.08273*, 2021. 2, 3

Michael Kelly, Chelsea Sidrane, Katherine Driggs-Campbell, and Mykel J Kochenderfer. Hg-dagger: Interactive imitation learning with human experts. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8077–8083. IEEE, 2019. 2, 3

Jelle Luijkx, Zlatan Ajanović, Laura Ferranti, and Jens Kober. Askdagger: active skill-level data aggregation for interactive imitation learning. *arXiv preprint arXiv:2508.05310*, 2025. 3

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

Ajay Mandlekar, Danfei Xu, Roberto Martín-Martín, Yuke Zhu, Li Fei-Fei, and Silvio Savarese. Human-in-the-loop imitation learning using remote teleoperation. *arXiv preprint arXiv:2012.06733*, 2020. 3

Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020. 3

Junhyuk Oh, Yijie Guo, Satinder Singh, and Honglak Lee. Self-imitation learning. In *International conference on machine learning*, pp. 3878–3887. PMLR, 2018. 3

Davide Paglieri, Bartłomiej Cupiał, Samuel Coward, Ulyana Piterbarg, Maciej Wołczyk, Akbir Khan, Eduardo Pignatelli, Łukasz Kuciński, Lerrel Pinto, Rob Fergus, et al. Balrog: Benchmarking agentic llm and vlm reasoning on games. In *International Conference on Learning Representations*, volume 2025, pp. 96666–96702, 2025. 3

Mingjie Pan, Siyuan Feng, Qinglin Zhang, Xinchen Li, Jianheng Song, Chendi Qu, Yi Wang, Chuankang Li, Ziyu Xiong, Zhi Chen, et al. Sop: A scalable online post-training system for vision-language-action models. *arXiv preprint arXiv:2601.03044*, 2026. 3

Rom Parnichkun, Matthew N Dailey, and Atsushi Yamashita. Reil: a framework for reinforced intervention-based imitation learning. *arXiv preprint arXiv:2203.15390*, 2022. 3

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5. 12

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. 1, 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. 3

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011. 2, 3

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015. 3

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

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

Elad Sharony, Tom Jurgenson, Orr Krupnik, Dotan Di Castro, and Shie Mannor. Vlm-guided experience replay. *arXiv preprint arXiv:2602.01915*, 2026. 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. 1, 3

Weihao Tan, Xiangyang Li, Yunhao Fang, Heyuan Yao, Shi Yan, Hao Luo, Tenglong Ao, Huihui Li, Hongbin Ren, Bairen Yi, et al. Lumine: An open recipe for building generalist agents in 3d open worlds. *arXiv preprint arXiv:2511.08892*, 2025. 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

Rui Yan, Zaitian Gongye, Lars Paulsen, Xuxin Cheng, and Xiaolong Wang. Human-robot copilot for data-efficient imitation learning. *arXiv preprint arXiv:2604.03613*, 2026. 3

Yifan Yang, Zhixiang Duan, Tianshi Xie, Fuyu Cao, Pinxi Shen, Peili Song, Chenyang Zhao, Piaopiao Jin, Guokang Sun, Shaoqing Xu, et al. Fpc-vla: A vision-language-action framework with a supervisor for failure prediction and correction. *Expert Systems with Applications*, pp. 131742, 2026. 3

Shaopeng Zhai, Qi Zhang, Tianyi Zhang, Haoran Zhang, Fuxian Huang, Zhanhui Lin, Zijun Xu, and Weinan Zhang. Help: Human-efficient large-scale robot post-training with rollout segmentation. *arXiv preprint arXiv:2607.09776*, 2026. 3

Weipu Zhang, Adam Jelley, Trevor McInroe, Amos Storkey, and Gang Wang. Object-centric world models from few-shot annotations for sample-efficient reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 91366–91394, 2026. 3

A Implementation Details and Additional Analysis

A.1 Actor, Actions, and Context

The compact actor is initialized from a native Qwen3.5-2B checkpoint (Qwen Team, 2026). Throughout SFT and RL, the vision encoder and multimodal projection layer remain frozen. We fine-tune all parameters of the language model, including its token embeddings, together with the output head.

Visual observations are resized to 512×512 and acquired every 200 ms. An action group has the form `DX DY KEY1 KEY2 ...`. Each supervised target serializes four groups with semicolons. At deployment, the actor streams groups as they are decoded and the client executes each available group at 50 ms intervals while the game is locked to 60 FPS. Thus, four groups span one 200 ms observation interval, but the inference parser does not require exactly four groups in a single decode. The enabled action vocabulary is `one`, `two`, `three`, `four`, `W`, `A`, `S`, `D`, `LB`, `RB`, `T`, `R`, `Space`, `Shift`, and `Ctrl`.

Demonstration encoding uses fixed non-uniform mouse-displacement quantizers with 47 horizontal and 33 vertical representative values. SFT preprocessing truncates numeric displacements toward zero before applying the mapping in Table 4; online human recording uses the same mapping on integer displacements. The signed representative values are serialized as decimal pixel values rather than bin indices. At deployment, the server parses generated XY fields directly as integers and transmits them without requantization or clipping to the representative-value range. Current-rollout and replay samples retain their stored action strings.

Table 4: Mouse-displacement encoding for integer inputs, in pixels. Intervals include both endpoints. The mapping preserves the input sign and maps zero to zero, yielding 47 signed X values and 33 signed Y values.

Input $\lvert d_x \rvert$	Encoded $\lvert d_x \rvert$	Input $\lvert d_y \rvert$	Encoded $\lvert d_y \rvert$
0	0	0	0
1	1	1	1
2	2	2	2
3	3	3	3
4	4	4–5	4
5–6	5	6–7	6
7–8	7	8–9	8
9–10	9	10–13	11
11–13	12	14–17	15
14–17	15	18–23	20
18–21	19	24–31	27
22–27	24	32–42	36
28–34	30	43–56	48
35–43	38	57–77	65
44–54	48	78–107	90
55–68	60	108–137	125
69–86	76	≥ 138	150
87–108	96		
109–137	120		
138–177	155		
178–230	200		
231–300	260		
301–420	340		
≥ 421	500		

At inference time, the deployed actor limits its context to 50 image–action turns, corresponding to approximately 10 seconds at 5 Hz. Upon reaching this limit, it resets the context while retaining the five most recent image–action pairs. This bounds context length, inference cost, and control latency. The inference context window is distinct from the training-time trajectory-chunk limit described below, even though both are configured as 50 interaction turns. Deployment applies the same ignored-key and repeated-key suppression used during evaluation.

Inference prompt and dialogue format. The deployed actor uses the following system instruction, with `[GAME NAME]` populated from the game client.

Actor system prompt

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`.

At inference time, the context contains the system message, a sequence of historical image–action pairs, and the current raw game image. No textual HP, stamina, boss-phase, or other game-state fields are appended to the actor prompt. The actor directly decodes an action string and does not produce a textual rationale or `<think>` tokens. The box below shows the message layout at turn n, with H historical image–action pairs.

Inference dialogue format

System	Actor system prompt above.
User	*[historical image at turn $n-H$]*
Assistant	`DX DY KEY1 KEY2 ...; ...`
	⋮
User	*[historical image at turn $n-1$]*
Assistant	`DX DY KEY1 KEY2 ...; ...`
User	*[current image at turn n]*
Assistant	`DX DY KEY1 KEY2 ...; ...`

A.2 Recovery-Library Initialization

FIRE initializes its recovery library by using the fixed SFT actor's failures to determine what experts should demonstrate. The failure-type registry starts empty. During initial rollout review, the VLM registers situation-specific failure labels and definitions using the diagnostic guidance in the prompt. During subsequent collection rounds, diagnoses reuse a label when its definition matches and register a new mode otherwise.

Before online policy updating, the fixed SFT policy π_{θ_0} collects five complete rollout episodes for each of the seven training bosses, yielding 35 diagnostic rollouts and approximately 90 minutes of native-game interaction. The actor remains fixed throughout initialization: no reward, routing, RL, CE, or policy-gradient update is applied, and the rollouts are not inserted into curated replay. The VLM directly reviews every full rollout video to identify underperforming situations and missing recovery behaviors. These observations are aggregated into recording requests, and human experts record approximately two minutes of targeted recovery demonstrations per boss, totaling approximately 14 minutes of initial human recovery data. An accepted demonstration may address several failure modes and stores its associated mode-label set and contextual metadata. These multi-label demonstrations initialize D_{rec} before Round 1.

A.3 Full Post-Training Procedure

Algorithm 1 summarizes FIRE's feedback loop between policy experience and human recovery acquisition. After library initialization, each outer round comprises $U = 7$ training stages, each processing $B = 200$ mixed chunks at a target current-rollout:curated-replay:CE ratio of 1:1:1. The actor π_θ is optimized throughout the stage, while the reference policy π_{ref} remains fixed. After all B chunks have been processed, the actor's current weights are copied to the reference policy, giving seven refreshes per round. At the end of each of the first $K_{\text{rec}} = 5$ rounds, accumulated failures guide targeted recording for subsequent rounds. Training continues for $K = 10$ rounds. During

Rounds 6–10, diagnosis and retrieval continue with the fixed registry and recovery library; no new failure modes are registered and no further demonstrations are collected.

Algorithm 1 FIRE: Failure-Informed REcovery Post-Training

Input: actor π_{θ_0}, demonstrations D_0, bosses $\mathcal{G}$
  online rounds K, recovery horizon K_{rec}
  stages per round U, chunks per stage B
Initialize: $\theta, \theta_{\mathrm{ref}} \leftarrow \theta_0$
State: $\mathcal{B}, D_{\mathrm{rec}}, \mathcal{Z} \leftarrow \varnothing$ ▷ empty buffers and registry
Recovery-library initialization
  $\mathcal{T}_{\mathrm{init}} \leftarrow \mathrm{Rollout}(\pi_{\theta_0}, \mathcal{G}, 5)$ ▷ no update
  $(\mathcal{E}_{\mathrm{init}}, \mathcal{Z}) \leftarrow \mathrm{ReviewInitialRollouts}(\mathcal{T}_{\mathrm{init}}, \mathcal{Z})$ ▷ situations + initial registry
  $D_{\mathrm{rec}} \leftarrow \mathrm{AcquireRecovery}(\mathcal{E}_{\mathrm{init}})$ ▷ human recording + annotation
Online post-training
for $k = 1, \ldots, K$ **do**
  $\mathcal{E}_k \leftarrow \varnothing$ ▷ round-level failure events
  for $u = 1, \ldots, U$ **do** ▷ fixed reference within stage
    $\mathcal{C} \leftarrow \mathrm{Rollout}(\pi_\theta, \mathcal{G})$ ▷ fresh policy data
    $(R^{\mathrm{hyb}}, q) \leftarrow \mathrm{EvaluateAndRoute}(\mathcal{C})$ ▷ Eqs. (1), (2)
    $\mathcal{B} \leftarrow \mathcal{B} \cup \{c \in \mathcal{C} : q(c) \in \{\textit{good}, \textit{mistake}\}\}$ ▷ curated replay
    $\mathcal{C}_{\mathrm{mis}} \leftarrow \{c \in \mathcal{C} : q(c) = \textit{mistake}\}$ ▷ failure-informed recovery
    $(\mathcal{E}, \mathcal{Z}) \leftarrow \mathrm{Diagnose}(\mathcal{C}_{\mathrm{mis}}, \mathcal{Z}, k \le K_{\mathrm{rec}})$ ▷ register only through K_{rec}
    $\mathcal{E}_k \leftarrow \mathcal{E}_k \cup \mathcal{E}$
    $\mathcal{D}_{\mathrm{CE}} \leftarrow \mathrm{Retrieve}(D_0, D_{\mathrm{rec}}, \mathcal{C}, \mathcal{E})$ ▷ boss + mode priority
    $\mathcal{Q} \leftarrow \mathrm{MixChunks}(\mathcal{C}, \mathcal{B}, \mathcal{D}_{\mathrm{CE}}, B)$ ▷ 1:1:1; same-boss replay
    for each minibatch b **in** shuffled $\mathcal{Q}$ **do**
      $\mathcal{L} \leftarrow \mathrm{ComputeLoss}(b; \theta, \theta_{\mathrm{ref}})$ ▷ RL or CE by data source
      $\theta \leftarrow \mathrm{OptimizerStep}(\theta, \mathcal{L})$
    end for
    $\theta_{\mathrm{ref}} \leftarrow \theta$ ▷ refresh after B chunks
  end for
  if $k \le K_{\mathrm{rec}}$ **then**
    $D_{\mathrm{rec}} \leftarrow D_{\mathrm{rec}} \cup \mathrm{AcquireRecovery}(\mathcal{E}_k)$ ▷ human recording
  end if
end for; **return** π_θ

A.4 Reward and Routing Implementation

The post-training rule evaluator uses player HP, boss HP, and gourd state to provide training feedback. The enabled step rules assign -2 when the player dies and $+2$ when the boss is killed; player damage contributes a maximum-HP-normalized penalty and boss damage a maximum-HP-normalized reward.

Within chunk c, index the actor's output turns by n and denote their raw rule scores by r_n^{rule}. Let G_n^{rule} be the discounted and clipped return, N_c the last output turn, and $n(t)$ the turn containing action step t. The return is computed as

$$G_n^{\mathrm{rule}} = \mathrm{clip}\big(r_n^{\mathrm{rule}} + 0.7 G_{n+1}^{\mathrm{rule}}, -4, 4\big), \quad G_{N_c+1}^{\mathrm{rule}} = 0. \tag{7}$$

Discounting occurs once per actor output turn within each chunk. The return is shared across the turn's four action groups as $R_t^{\mathrm{rule}} = G_{n(t)}^{\mathrm{rule}}$. Turns with zero immediate rule scores retain the discounted continuation.

Gemini 3.7 Flash performs trajectory scoring, failure diagnosis, and semantic labeling of recovery demonstrations. For trajectory scoring, it receives each chunk's 5 Hz gameplay video with real-time player HP and boss HP displayed in the upper-right overlay. It returns JSON fields `reason` and `score`; the planning score v_j lies in $[0, 1]$ and is recorded to one decimal place. The following is the trajectory scoring prompt.

Trajectory scoring prompt

Role and input

You are evaluating the video of one trajectory chunk from a real-time action game. The upper-right overlay displays real-time Player HP and Boss HP. Review the video to assess the player's behavior and relate health changes to the combat situations shown.

`[GAMEPLAY VIDEO]`

Use as supporting evidence

- Declining Boss HP indicates successful damage.
- Declining Player HP indicates damage received.
- Use the visible behavior to assess the timing and coordination of movement, attacks, dodges, and item use.
- Evaluate the overall quality of the player's control, not isolated frames.

Assess

Action continuity, combat effectiveness, defense and evasion, skill and item use, combat rhythm, and survival awareness.

Score scale

- **0.0–0.2:** Very poor control, including repeated unaddressed threats, unsafe low-HP offense, or death.
- **0.2–0.4:** Poor and inefficient control, frequent mistimed dodges, or unsafe low-HP decisions.
- **0.4–0.6:** Neutral play, ordinary movement, or non-combat behavior; assign 0.5 when there is no meaningful combat interaction.
- **0.6–0.8:** Coherent and effective play with limited damage taken.
- **0.8–1.0:** Precise, tactically strong control with efficient attacks and reliable evasions.

Judgment rules

Normal movement, running, repositioning, and approaching an enemy are legitimate behavior and must not be penalized. Judge behavior strictly only during meaningful combat. When visible stamina is depleted, repeated attacks are low-value; the player should commonly reposition or wait for recovery. At low HP, especially when healing remains available, survival should take priority over unsafe offense. Base the assessment on the gameplay video and its HP overlay.

Output

Return only one JSON object, without Markdown or any additional text:

```
{
  "reason":  "50--150 word overall control assessment",
  "score":  0.0
}
```

The `score` must be a one-decimal number in $[0, 1]$.

For chunk j, let d_j be its boss-HP reduction and $D_{\mathcal{S}}$ the reduction over its complete battle (session $\mathcal{S}$). The historical reference $\bar{D}_5$ is maintained separately for each boss: battle-level reductions are averaged within each training stage, then the stage means are averaged over the preceding five stages. With fewer than five available stages, all available stage means are used; with no history, $\bar{D}_5 = 0$. These HP-reduction quantities are normalized by total initial Boss HP and expressed as fractions in $[0, 1]$. The implemented trajectory score is

$$\tau_j = \min\big(1, v_j + d_j + D_{\mathcal{S}} - \bar{D}_5\big)\,. \tag{8}$$

Let $j(t)$ be the chunk containing action step t. Its score is broadcast to the covered action steps as

$$R_t^{\mathrm{vlm}} = 0.5\big(\tau_{j(t)} - 0.5\big), \qquad R_t^{\mathrm{hyb}} = R_t^{\mathrm{rule}} + R_t^{\mathrm{vlm}}. \tag{9}$$

This assignment carries the chunk's assessment of high-level tactical quality to its covered actions, while the rule term represents their visible combat outcomes. The trajectory score has no lower clipping bound. Terminal win/loss enters the immediate rule score at the turn containing the kill or death event, rather than being broadcast through the VLM term.

The reported online post-training protocol uses fixed routing values

$$\eta_{\mathrm{mistake}} = -0.5, \qquad \eta_{\mathrm{good}} = 0.5. \tag{10}$$

Positive- and negative-evidence detection both consider all action steps in a policy chunk. Chunks containing only negative evidence are labeled `bad`, while those containing both positive and negative evidence are labeled `good_bad`; both map to mistake chunks. Returns exactly equal to either threshold do not trigger the corresponding strict comparison, and chunks with neither trigger are normal and discarded from replay. All retained labels share one curated-replay store and are distinguished by metadata.

A.5 Chunking, Optimization, and Replay

A trajectory chunk is one training sample and contains at most 50 interaction turns, including any preceding context. This cap applies to original SFT demonstrations, recovery demonstrations, current rollouts, and curated replay. The replay loader rejects chunks exceeding the cap and leaves shorter chunks unpadded. Each training stage processes 200 mixed chunks, with current-rollout, replay, and CE candidates assembled at a target ratio of 1:1:1 and shuffled before training. For each replay request, chunks are sampled uniformly with replacement from the corresponding boss's pool. Good and mistake labels determine routing eligibility for replay; sampling within the boss pool is uniform. An empty replay pool supplies no chunks, so source availability and stage termination can affect the realized mixture.

We train the shared SFT initialization on approximately 75 hours of original demonstrations with a learning rate of 10^{-6}. Post-training uses AdamW with the same learning rate, weight decay 0.01, and gradient-norm clipping at 1.0. The actor is updated on each training minibatch. Entropy and KL coefficients are zero. The reference policy is refreshed after every 200 mixed chunks.

The fixed CISPO clipping constants are

$$\ell = -20, \qquad u = 20, \qquad \bar{\rho} = 1.28. \tag{11}$$

The token-level CISPO-style objective in Section 3.5 clamps the log-ratio to $[\ell, u]$ before exponentiation and applies the detached upper ratio-weight cap $\bar{\rho}$. It has no lower ratio cap and performs no return or advantage normalization. RL gradients are masked to valid keyboard or mouse-button target tokens with nonzero final training signals, excluding mouse-displacement tokens. Boss encounters use target lock, under which mouse x/y movements do not change the camera view. Mouse-displacement targets remain part of supervised learning.

For both current-rollout and curated-replay chunks, each training forward pass computes target-action log-probabilities under the current actor and an independent frozen reference model. Historical chunks are rescored under the current reference snapshot whenever they are reused. Thus, ρ_i measures the actor's likelihood relative to the stage's reference snapshot, rather than to the policy that originally generated the chunk. The reference weights are initialized from π_{θ_0} and refreshed from the actor at each stage boundary. The same reference forward pass supplies probabilities for the replay loss filter described below.

The fixed CE component weights in the reported implementation are

$$\lambda_{\mathrm{key}} = 0.5, \qquad \lambda_{\mathrm{mouse}} = 0.5. \tag{12}$$

Curated replay stores selected policy chunks with their processed turn scores and routing metadata. Stored scores include collection-time reward processing; for encounters with total boss HP reduction $D > 0.9$, the collection routine adds a progress bonus $D - 0.5$. During reuse, scores strictly above 0.5 or below -0.5 retain their original magnitudes, and all others are set to zero. Returns are

recomputed backward within the chunk with discount 0.7 and clipping to $[-4, 4]$ at each turn. The recursion ends at the chunk boundary and is reset at context turns; zero-score policy turns retain discounted continuation. Replay uses these recomputed returns without adding the separately stored trajectory-score contribution. Reference log-probabilities are recomputed on each use. Replay has no configured capacity, age, or per-chunk reuse limit.

Reference-based replay filtering. The replay loss filter operates on the ordered list of supervised policy turns, excluding context and turns without four valid action groups or with zero recomputed return. For each turn, it takes the arithmetic mean of reference probabilities over keyboard and mouse-button target tokens, falling back to all supervised tokens when no such key tokens are present. A significant turn has both a nonzero thresholded stored score and a nonzero recomputed return. For each significant turn, the filter multiplies the mean probabilities over that turn and up to three preceding supervised turns, excluding the preceding significant turn and all turns before it. This window is indexed by supervised turns rather than elapsed game time.

A significant turn retains its signal when the product is below 0.7 for a positive recomputed return or above 0.3 for a negative return. Otherwise, its own signal is set to zero. Other supervised turns retain their propagated signals only when they lie within three preceding entries of a significant turn identified before probability filtering, whether or not that significant turn passes the probability gate. Filtering leaves the previously computed returns at these retained turns unchanged and does not rerun discounting. Thus, a rejected significant turn can still contribute through returns already propagated to preceding turns. The chunk remains in the replay pool for later sampling.

The resulting signal is denoted $\widetilde{R}_t$ and shared by all four action groups in its turn. The RL mask is constructed after replay filtering: a token is retained when its target label is valid, its token group is keyboard or mouse-button, and its final signal is nonzero. Both the loss sum and its denominator use this retained set $\widetilde{\mathcal{M}}$. The denominator is $\max(1, |\widetilde{\mathcal{M}}|)$, giving zero RL loss when no tokens remain. Current-rollout chunks use $\widetilde{R}_t = R_t^{\mathrm{hyb}}$ with the same token eligibility rule.

A.6 Failure-Informed Recovery Acquisition and Retrieval

Failure-informed recovery uses diagnosed mistakes to determine what experts record and which demonstrations supervise later updates. Each mistake chunk is treated as one event and yields one JSON object in the format below. The event can carry multiple semantic failure-mode labels $\mathcal{L}_e$, together with a visual trigger, a failure explanation, the desired response, and a human recording request. A complete rollout can contain several events across its mistake chunks.

At each collection round, failure events are aggregated by boss and semantic mode to prioritize missing recovery behaviors. Modes with no corresponding human recovery data receive first priority. The remaining modes are ranked in descending order of failure frequency divided by the duration of existing human recovery demonstrations for the same boss and mode. These priorities determine the recording queue. Experts create the requested combat conditions through gameplay and record the recovery behaviors. New failure-mode registration is enabled during initialization and Rounds 1–5; subsequent diagnosis uses the fixed registry for retrieval, as specified in Algorithm 1.

Human agreement with failure diagnosis. The final failure-type registry contains 29 labels across the seven training bosses, each covered by at least one annotated human recovery segment. We randomly sample 200 chunks from the mistake-chunk pool to assess VLM diagnosis. Two human annotators independently review each chunk and select all applicable labels from the corresponding boss's full set of registered failure labels. Annotators can access the label definitions but are blinded to the VLM's selections. Let V_i denote the VLM-selected label set for chunk i and $H_{i,a}$ the set selected by annotator a. We compute mean Jaccard agreement over the 400 human–VLM comparisons:

$$\text{Agreement} = \frac{1}{200 \times 2} \sum_{i=1}^{200} \sum_{a=1}^{2} \frac{|V_i \cap H_{i,a}|}{|V_i \cup H_{i,a}|} \times 100\%.$$

The mean agreement is 92.10%.

Failure diagnosis prompt

Role and context

You are a high-level *Black Myth: Wukong* player and action-game coach reviewing a poor gameplay segment produced by an AI policy. The first seconds may contain the setup that caused a later error; inspect this lead-in carefully. `[BOSS CONTEXT] [REGISTERED FAILURE TYPES]`

Diagnosis procedure

- Explain why the failure occurred and what the correct response should have been. Use over-commitment, low offensive pressure, failed evasion, and mistimed healing as diagnostic guidance, and identify the specific combat situation in which the error occurs.
- Identify the visible enemy cue—for example a wind-up, weapon posture, body movement, visual effect, or distance change—then identify the policy's actual action and why it was unsafe or ineffective.
- Prescribe a response with controls and timing when the clip supports them. Consider low-HP survival, healing safety, and stamina recovery.
- Do not invent exact timing, attack properties, audio cues, or hidden state.

Failure-type registry

Read the specific definition of each relevant registered label, rather than matching its name alone. Reuse its English key exactly when the definition covers the observed combat situation and response failure. If no registered definition covers that situation, register a new label even if the error shares a broad diagnostic category with an existing mode. Provide a concise lower-snake-case key and a definition in `new_failure_types`, identifying the relevant boss mechanic, visible cue, and response failure. Reuse labels for repeated instances of the same defined mode.

Registration example: Guangzhi

If the policy fails to evade Guangzhi's aerial charged attack and no registered definition covers the situation, the label-related fields of the output could be:

```
{
"failure_types": ["guangzhi_aerial_evasion_failure"],
"new_failure_types": [
                {
                  "key": "guangzhi_aerial_evasion_failure",
                  "definition": "Fails to evade Guangzhi's aerial
              charged attack after the visible aerial wind-up,
              resulting in a hit."
                }
              ]
```

} Reuse this key for later events covered by its definition, leaving `new_failure_types` empty when no additional mode is identified.

Output

Return only one JSON object, without Markdown:

```
{
  "visual_trigger": "30--60 word visible cue",
  "failure_types": ["existing_or_new_lower_snake_case", "..."],
  "root_cause": "40--80 word explanation",
  "correct_response": "40--80 word corrective behavior",
  "coaching_request": "30--60 word demonstration request",
  "severity_agrees": true,
  "caption": "short failure caption",
  "new_failure_types": [
```

```
    {"key": "new_lower_snake_case", "definition": "concise
definition"}
  ]
}
```

If no clear policy error is visible, set `severity_agrees` to false, `failure_types` to `["no_clear_failure"]`, `coaching_request` to an empty string, and `new_failure_types` to `[]`; do not register `no_clear_failure`.

Human recovery demonstration annotation prompt

Role and input

You are reviewing a human-recorded recovery demonstration for *Black Myth: Wukong*. The segment was collected in response to known recovery needs. Identify the registered failure modes whose recovery behavior the human actually demonstrates, and record the visible context needed to retrieve this segment for a similar mistake. `[BOSS CONTEXT]` `[RECORDING REQUESTS]` `[REGISTERED FAILURE TYPES]`

Annotation procedure

- Verify that the demonstrated encounter matches the supplied boss context.
- Inspect the visible enemy cue, the player's position and resources, and the sequence of recovery actions. A segment may address multiple failure modes.
- Treat the supplied recording requests as contextual references, not as a closed label set. Read the definitions in the full registered failure-type registry and select every applicable label. Match a label only when the demonstrated behavior addresses the combat situation and recovery need specified by its definition.
- Record combat phase, player-HP bucket, boss-HP bucket, and distance bucket only when supported by the video. Use `unknown` rather than inferring hidden state.
- Describe the demonstrated recovery in observable action and timing terms. Do not diagnose a new failure, create a new failure type, or request additional recording.

Output

Return only one JSON object, without Markdown:

```
{
  "boss": "boss_name",
  "eligible": true,
  "failure_types": ["registered_lower_snake_case", "..."],
  "event_context": {
    "combat_phase": "observed_phase_or_unknown",
    "player_hp_bucket": "observed_bucket_or_unknown",
    "boss_hp_bucket": "observed_bucket_or_unknown",
    "distance_bucket": "observed_bucket_or_unknown"
  },
  "visual_trigger": "20--50 word visible cue",
  "demonstrated_recovery": "40--80 word corrective behavior"
}
```

If the segment does not match the boss context or does not visibly demonstrate recovery behavior addressing any registered failure mode, set `eligible` to `false`, `failure_types` to `[]`, and `demonstrated_recovery` to an empty string.

Shared failure-mode labels link the acquired demonstrations to the policy situations they address. Each accepted human recovery segment stores one or more such labels and the corresponding con-

textual metadata. Before entering the retrieval pool, segments are automatically screened for boss compatibility, contextual relevance, and segment quality.

For a diagnosed mistake event e, the priority pool contains accepted recovery segments whose boss matches that of e and whose mode-label set shares at least one label with $\mathcal{L}_e$. We sample uniformly from this pool. If no event-specific priority pool is available, including for a rollout without a diagnosed mistake event, we sample uniformly from accepted recovery segments associated with the rollout boss. Supervised updates use each demonstration's own observation–action sequence. Targeted recovery demonstrations occupy up to 70% of the CE stream. The non-targeted 30% stream consists of original SFT demonstrations, which also fill any targeted-data shortfall.

A.7 Baseline Implementation Details

The SFT baseline is the shared supervised initialization π_{θ_0}. Outcome-reward RL, Rule reward only, and Hybrid-reward RL retain FIRE's current-rollout and replay allocations without CE updates, using approximately two-thirds of its total training chunks. Outcome-reward RL assigns $+2$ for a win and -2 for a loss, broadcasting this value to all action steps in the encounter. Rule reward only uses rule-based HP-change and terminal feedback without VLM planning feedback. Hybrid-reward RL uses FIRE's hybrid feedback, combining rule returns with calibrated VLM planning and damage-progress contributions for current rollouts.

The iRe-VLA-inspired baseline interleaves rollout RL using FIRE's hybrid feedback with supervised updates at an RL:SL chunk ratio of 1:2, matching FIRE's total training-chunk budget. Its supervision combines original demonstrations and terminal-success online trajectories. It uses neither curated replay nor failure-guided recovery acquisition.

The w/o Human Recovery Library control replaces targeted recovery demonstrations with approximately 84 minutes of additional successful boss-fight recordings across the seven training bosses. Collection follows FIRE's initialization and Rounds 1–5 schedule without failure-guided recording requests. Hybrid feedback, curated replay, RL, and CE updates are retained.

Static recovery retains FIRE's label-based retrieval, hybrid feedback, curated replay, RL, and CE updates. Human recording targets are restricted to failure labels diagnosed from the initial SFT rollouts and are not updated from subsequent policy failures.

The direct VLM baselines use GPT-5.6 Luna and Gemini-3.7 Flash with the same actor system prompt and image–action dialogue template as the compact actor. To accommodate API latency, each request predicts ten action groups, which the client consumes at 100 ms intervals. The client refreshes a latest-frame buffer every 200 ms and serializes API requests. After a request completes, the next request uses the most recent buffered frame. If the action queue is empty, the client continues executing the preceding action group.

A.8 Training and Collection Protocol

Original SFT demonstrations comprise approximately 75 hours. The fixed SFT policy π_{θ_0} collects five diagnostic episodes per training boss, yielding 35 complete rollout videos (approximately 90 minutes) before any updates. Failure diagnosis of these videos determines initial recording requests; the rollouts are excluded from replay. Regular post-training collects approximately 11 minutes per boss and round, or 77 minutes per round. Each FIRE round comprises seven 200-chunk stages, giving 1,400 mixed chunks and seven reference-policy refreshes. All RL-based methods run ten rounds, totaling approximately 770 minutes of regular rollout. Including initialization, FIRE uses approximately 860 minutes of policy-generated native-game interaction.

FIRE and both recovery controls receive approximately two minutes of additional demonstrations per boss before the first online update and two more after each of Rounds 1–5. This schedule provides 14 minutes initially and 70 minutes during training, totaling 12 minutes per boss and 84 minutes across seven bosses. FIRE redirects recording toward accumulated policy failures, whereas Static recovery allocates recording time uniformly across each boss's initial SFT failure labels. Rounds 6–10 reuse the collected data without further recording. FIRE registers no new failure modes after Round 5.

Human effort for failure-mode preparation and recording is estimated at approximately 250 minutes for FIRE, versus 170 minutes for the equal-video-budget successful-demonstration control. Both recovery variants allocate up to 70% of the CE sampling stream to recovery demonstrations. Original SFT demonstrations supply the remaining 30% and fill any recovery-data shortfall.

A.9 Recovery-Curve Readings

Round 0 is the shared supervised initialization. Table 5 provides the exact HP Reduction values underlying Figure 3, and Table 6 reports the corresponding Win Rates for all three methods. FIRE has higher mean HP Reduction than the successful-demonstration control at all ten post-training checkpoints on each boss, and than Static recovery from Round 2 onward.

At Round 10, FIRE's HP Reduction gains over Static recovery are 1.54, 7.68, and 6.19 percentage points on Guangzhi, Black Wind King, and Black Bear Guai, respectively; its Win Rate gains are 5, 10, and 10 points. Relative to the successful-demonstration control, FIRE gains 1.75, 9.89, and 7.54 points in HP Reduction, while Static recovery gains 0.21, 2.21, and 1.35 points.

On Black Wind King and Black Bear Guai, FIRE first surpasses Static recovery's highest observed HP Reduction across all ten post-training checkpoints at Rounds 4 and 7, respectively. Against the successful-demonstration control, the corresponding rounds are 4 and 5. These comparisons show earlier attainment of higher performance, with larger final gains on the harder encounters.

Table 5: Mean Boss HP Reduction (%) for Figure 3, with Player Attack $= 50$, Player Defense $= 600$ and 20 episodes per point. Static denotes Static recovery; w/o HRL denotes w/o Human Recovery Library. Round 0 is the shared SFT initialization. Bold marks the best result per boss and round, including ties.

	Guangzhi			Black Wind King			Black Bear Guai		
Round	**Ours**	Static	w/o HRL	**Ours**	Static	w/o HRL	**Ours**	Static	w/o HRL
0	**87.73**	**87.73**	**87.73**	**57.76**	**57.76**	**57.76**	**57.31**	**57.31**	**57.31**
1	86.60	**86.77**	85.84	**56.11**	55.28	53.25	60.05	**60.09**	57.53
2	**87.95**	86.32	84.73	**59.26**	58.18	57.56	**59.67**	58.83	58.02
3	**89.38**	87.94	86.59	**60.79**	58.44	57.84	**63.86**	59.96	61.35
4	**90.85**	89.12	89.73	**63.46**	59.15	55.63	**66.19**	61.40	61.27
5	**90.55**	90.49	90.53	**64.92**	59.74	57.31	**68.98**	62.61	63.12
6	**93.67**	91.98	92.32	**67.59**	61.11	59.26	**69.05**	64.39	62.43
7	**95.13**	92.28	91.28	**69.67**	60.96	60.01	**73.86**	66.21	65.06
8	**95.62**	93.02	92.91	**71.01**	61.44	60.23	**72.39**	67.58	65.43
9	**95.45**	93.76	93.49	**68.87**	62.79	61.98	**74.51**	69.16	67.95
10	**95.53**	93.99	93.78	**70.95**	63.27	61.06	**75.07**	68.88	67.53

Table 6: Win Rate (%) at checkpoints in Figure 3, with Player Attack $= 50$, Player Defense $= 600$ and 20 episodes per point. Method abbreviations follow Table 5. Round 0 is the shared SFT initialization. Bold marks the best result per boss and round, including ties.

	Guangzhi			Black Wind King			Black Bear Guai		
Round	**Ours**	Static	w/o HRL	**Ours**	Static	w/o HRL	**Ours**	Static	w/o HRL
0	**55**	**55**	**55**	**15**	**15**	**15**	**20**	**20**	**20**
1	50	50	**55**	**10**	**10**	**10**	**25**	20	20
2	**55**	**55**	**55**	**15**	**15**	**15**	**20**	**20**	**20**
3	**60**	55	50	**15**	**15**	**15**	**30**	25	25
4	**65**	60	60	**20**	15	10	**30**	25	20
5	**60**	**60**	**60**	**15**	**15**	**15**	**35**	25	25
6	**70**	65	65	**25**	20	20	**40**	30	20
7	**80**	70	60	**30**	25	20	**30**	**30**	**30**
8	**75**	70	70	**35**	20	15	**35**	30	25
9	**80**	75	75	**25**	20	15	**40**	35	30
10	**80**	75	75	**35**	25	20	**40**	30	30

A.10 Evaluation Configuration

Every method–task combination targets 20 valid episodes. Let $h_{j,0}^{\text{boss}}$ and $h_{j,T}^{\text{boss}}$ denote the total initial and remaining Boss HP across encounter phases in episode j. Normalized Boss HP Reduction is

$$\text{HPRed} = \frac{1}{20}\sum_{j=1}^{20}\frac{h_{j,0}^{\text{boss}} - h_{j,T}^{\text{boss}}}{h_{j,0}^{\text{boss}}} \times 100\%. \tag{13}$$

White-Clad Noble and Non-Able each have two phases. For each boss, the denominator is the sum of the initial HP of both phases, and HP reduction accumulates damage across the two phases. An undefeated phase retains its remaining HP, including the full HP of a phase not yet reached. Win Rate records whether the final boss phase is defeated, while HP Reduction measures continuous combat progress. We compare methods boss by boss rather than pooling win rates.

Evaluation uses sampling temperature 1.0 and the same action post-processing as deployment. No code-level episode timeout is imposed. Training and evaluation are started during periods of relatively stable network connectivity between the model server and game client. For non-API models, episodes are retained when their average end-to-end delay is at most 220 ms. This delay includes model inference to generate actions and transmission of those actions to the game client. Under these operating conditions, approximately 5% of non-API training and evaluation episodes are excluded by this threshold. Evaluation continues until 20 valid episodes have been obtained. The direct VLM API baselines are evaluated over 20 episodes without this latency filter. All post-training results use Round 10 without selecting checkpoints on test-task performance.

Human evaluation protocol. Five participants who had not previously played *Black Myth: Wukong* provide the novice-human reference. Each participant completes four episodes per evaluated setting, giving 20 episodes in total. Before testing each boss, participants watch a corresponding boss-fight video, for both the in-domain and held-out evaluations. The fixed boss order is Guangzhi, Wandering Wight, Lingxuzi, Guangmou, White-Clad Noble, Black Wind King, Black Bear Guai, First Prince of the Flowing Sands, Gore-Eye Daoist, Non-Able, and Captain Wise-Voice. Participants gain gameplay experience throughout this sequence, so novice status refers to their experience at the start of the study.

Table 7 specifies the player configuration for the native Chapter-1, controlled, and held-out evaluations. The native Chapter-1 state uses no external attribute modifier: its attributes and skills are attainable through standard Chapter-1 gameplay. The controlled $50/600$ evaluations modify only Player Attack and Player Defense; the boss-specific native HP, available skills, and equipment effects remain unchanged. At test time, any action selecting an unavailable skill is ignored and cannot trigger that ability.

Table 7: Player configurations used for evaluation. I: Immobilize; R: Red Tides (Guangzhi transformation); WWS: Wandering Wight spirit; CS: Cloud Step; FM: Fireproof Mantle (vessel). All four Opulence armor pieces are equipped for the first five encounters listed. Its two-piece bonus reduces jumping stamina costs; its headpiece increases movement speed and reduces action stamina costs after a successful three- or four-Focus-Point attack. The White-Clad Scholar set provides slow health regeneration.

Evaluation tasks	HP	Attack	Defense	Abilities	Armor set
Wandering Wight	300	36	33	I, R	Opulence set
Guangzhi	300	36	33	I	Opulence set
Lingxuzi	300	36	53	I, R, WWS	Opulence set
Guangmou	360	36	53	I, R, WWS	Opulence set
White-Clad Noble	360	36	73	I, R, WWS	Opulence set
Black Wind King	360	40	96	I, R, WWS	White-Clad Scholar set
Black Bear Guai	360	40	96	I, R, WWS, CS, FM	White-Clad Scholar set
Controlled $50/600$ variants	native	50	600	native	native
Held-out later-chapter bosses	875	68	486	all skills	White-Clad Scholar set

For the comparison with reported CombatVLA results, Wandering Wight and Guangzhi use Player Attack $= 100$ and Player Defense $= 600$ to align their attributes with the reported setting. The remaining practical tasks retain the respective evaluation configurations. Held-out bosses are excluded from training, recovery collection, and checkpoint selection.

The *Sekiro* tasks use the same *Black Myth: Wukong*-trained checkpoint, with no *Sekiro* data used in SFT or online post-training. Following CombatVLA's control mapping, we bind deflection to `Space` in place of the right mouse button.

A.11 Fixed-Checkpoint Episode Variability

Figure 5 reports episode-level outcomes for one fixed Black Bear Guai checkpoint under native Chapter-1 attributes. The same checkpoint is evaluated in 20 independent episodes, with no model update between episodes. Mean HP Reduction is 59.76%, and four episodes reach 100% HP Reduction and successfully defeat the boss. The connecting line indicates episode order only and must not be interpreted as a learning curve.

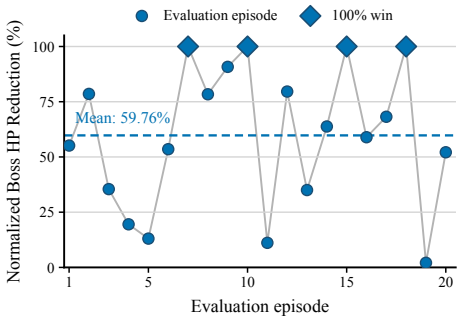


Figure 5: Episode-level variability for one fixed Black Bear Guai checkpoint under native Chapter-1 attributes over 20 independent evaluations. No model update occurs between episodes. Mean HP Reduction is 59.76%, and four episodes reach 100% and defeat the boss. The connecting line denotes episode order rather than learning progress.

A.12 Human-Rated Difficulty and Performance

Table 8 reports the human-rated difficulty and the corresponding HP Reduction for all 18 tasks: 14 in-domain boss–attribute configurations and four held-out bosses. Eight players, all of whom had completed a first playthrough of *Black Myth: Wukong*, rated each encounter. Ratings were averaged and normalized to a 1–10 scale.

Table 8: Human-rated task difficulty and normalized Boss HP Reduction (%). "Native" denotes the unmodified Chapter-1 progression state; "50/600" denotes the controlled Player Attack/Defense setting. Δ is Ours $-$ SFT in percentage points. Best results per task are bold.

Domain	Task / configuration	Difficulty	SFT	**Ours**	Δ
Held-out	Gore-Eye Daoist	3.0	**93.10**	92.62	−0.48
Held-out	First Prince of the Flowing Sands	4.0	84.16	**91.24**	+7.08
Held-out	Captain Wise-Voice	5.0	57.51	**76.20**	+18.69
Held-out	Non-Able	5.0	69.59	**88.63**	+19.04
In-domain	Guangmou (50/600)	3.5	81.48	**92.17**	+10.69
In-domain	Guangzhi (50/600)	4.0	87.73	**95.53**	+7.80
In-domain	Lingxuzi (50/600)	4.0	93.19	**96.62**	+3.43
In-domain	Guangmou (native)	5.0	64.34	**73.85**	+9.51
In-domain	Lingxuzi (native)	5.0	72.73	**73.10**	+0.37
In-domain	Guangzhi (native)	6.0	65.88	**71.01**	+5.13
In-domain	White-Clad Noble (50/600)	7.5	74.24	**87.05**	+12.81
In-domain	Wandering Wight (50/600)	8.0	68.54	**80.68**	+12.14
In-domain	Black Bear Guai (50/600)	8.5	57.31	**75.07**	+17.76
In-domain	Black Wind King (50/600)	9.0	57.76	**70.95**	+13.19
In-domain	White-Clad Noble (native)	9.5	29.50	**69.43**	+39.93
In-domain	Black Bear Guai (native)	10.0	31.83	**59.76**	+27.93
In-domain	Black Wind King (native)	10.0	39.10	**58.57**	+19.47
In-domain	Wandering Wight (native)	10.0	31.71	**56.59**	+24.88

A.13 Recovery Behavior Evaluation

Table 3 evaluates SFT and the final Round-10 checkpoints of w/o Human Recovery Library, Static recovery, and FIRE under Player Attack $= 50$, Player Defense $= 600$. For each method and boss, 20 episodes are examined. Only the first occurrence of the relevant situation in each episode is scored. Episodes that do not reach the situation or have the required skill unavailable at that occurrence are excluded from the denominator. A later occurrence is not substituted, so each episode contributes at most one trial.

The three criteria are as follows:

1. Guangmou: activate Red Tides, the Guangzhi transformation, during the snake-summoning phase. The window covers the entire phase, from snake summoning until the snakes disappear.
2. Black Wind King: move to the arena edge during wind form, wait for the boss to land, and release either a charged heavy-staff attack or a heavy attack in the Red Tides form. Releasing the attack is sufficient, regardless of whether it hits.
3. Black Bear Guai: activate the Fireproof Mantle during wind form and remain alive until that same phase ends. This criterion jointly requires activation and phase survival, not a win or survival through the rest of the encounter.

Each table cell reports k/n and $100k/n$ as a percentage, where k counts episodes meeting the case-specific criterion and n counts eligible episodes among the 20 evaluated. The Guangmou and Black Wind King responses were deliberately recorded only for FIRE. Both Static recovery and FIRE deliberately recorded Fireproof Mantle use.

A.14 Generalization to Unseen Later-Chapter Bosses

Table 9 evaluates the final policy on four later-chapter bosses excluded from original demonstrations, online rollouts, recovery data, and checkpoint selection. No further training or recovery collection is performed.

Table 9: Evaluation on held-out bosses from later chapters of Black Myth: Wukong. Cells report Normalized Boss HP Reduction / Win Rate (%). Bold marks the best model result per metric, including ties; humans provide a reference.

Method	First Prince of the Flowing Sands	Gore-Eye Daoist	Non-Able	Captain Wise-Voice
Novice human players	96.17 / 85	100.00 / 100	82.93 / 50	87.31 / 60
SFT	84.16 / 55	93.10 / **70**	69.59 / 35	57.51 / 20
w/o Human Recovery Library	89.64 / **65**	**93.57 / 70**	87.46 / 50	**79.27 / 45**
Ours	**91.24 / 65**	92.62 / **70**	**88.63 / 55**	76.20 / 40

FIRE improves both metrics over SFT on three of four held-out bosses. Win Rate increases by 10 percentage points on First Prince of the Flowing Sands and by 20 points on both Non-Able and Captain Wise-Voice. These gains without additional training support stronger generalization to unseen bosses than SFT.

A.15 Qualitative Gameplay Sequences

Figures 6–16 present gameplay sequences for the seven in-domain and four held-out bosses. Each sequence contains six frames in chronological order, read from left to right and then top to bottom.



(a) (b) (c)

(d) (e) (f)

Figure 6: Gameplay sequence for Guangzhi.



(a) (b) (c)

(d) (e) (f)

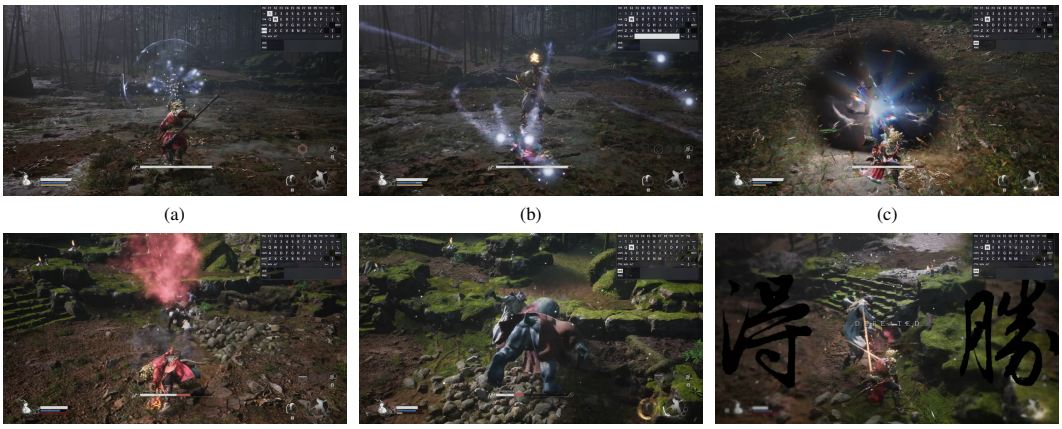
Figure 7: Gameplay sequence for Wandering Wight.



(a) (b) (c)

(d) (e) (f)

Figure 8: Gameplay sequence for Lingxuzi.



(a) (b) (c)

(d) (e) (f)

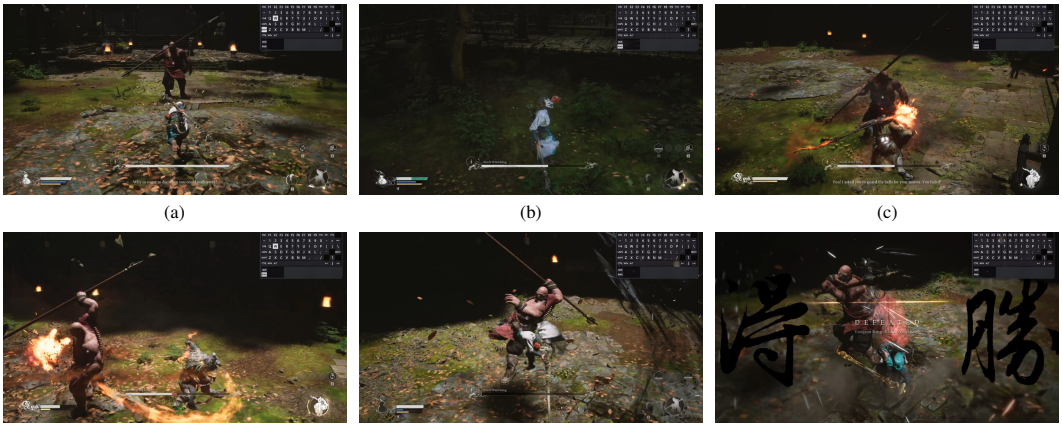
Figure 9: Gameplay sequence for Guangmou.



(a) (b) (c)

(d) (e) (f)

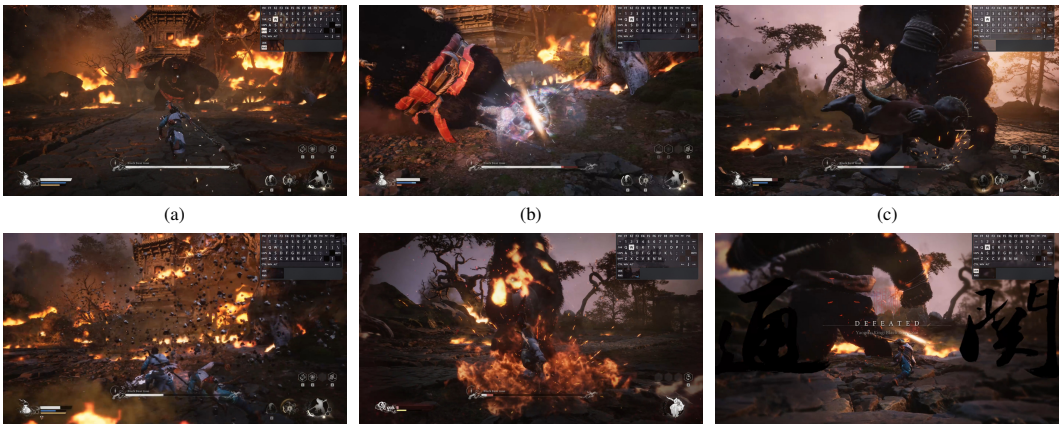
Figure 10: Gameplay sequence for White-Clad Noble.



(a) (b) (c)

(d) (e) (f)

Figure 11: Gameplay sequence for Black Wind King.



(a) (b) (c)

(d) (e) (f)

Figure 12: Gameplay sequence for Black Bear Guai.



(a) (b) (c)

(d) (e) (f)

Figure 13: Gameplay sequence for First Prince of the Flowing Sands.



(a) (b) (c)

(d) (e) (f)

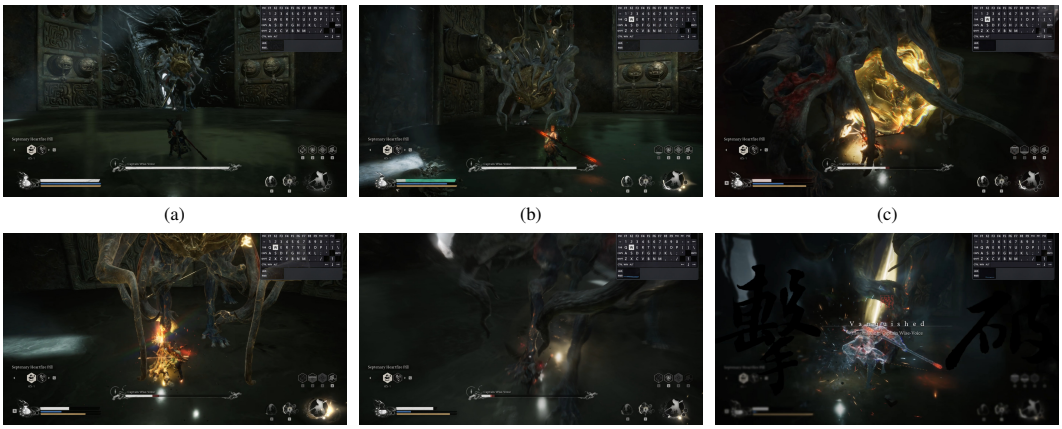
Figure 14: Gameplay sequence for Gore-Eye Daoist.



(a) (b) (c)

(d) (e) (f)

Figure 15: Gameplay sequence for Non-Able.



(a) (b) (c)

(d) (e) (f)

Figure 16: Gameplay sequence for Captain Wise-Voice.

A.16 Additional Qualitative Recovery Cases

Guangmou: transformation during snake summoning. Figure 17 contrasts responses to Guangmou's snake-summoning phase. The mistake rollout remains exposed to the snakes' attacks. After FIRE post-training, the actor activates Red Tides, the Guangzhi transformation, at the summoning cue and uses the transformed state to get through this dangerous interval. The recovery behavior lies in linking transformation to the boss phase in which the policy previously failed.

Mistake rollout



Post-trained actor (FIRE)



Figure 17: Guangmou recovery case. The FIRE actor activates Red Tides at the snake-summoning cue. Frames in each row are ordered chronologically from left to right.

Black Bear Guai: protection during wind form. In Figure 18, the mistake rollout takes repeated hits during Black Bear Guai's wind-form phase. The FIRE actor activates the Fireproof Mantle after the boss enters wind form, reducing incoming damage and surviving until the phase ends. Together, these cases illustrate the state-dependent recovery behaviors FIRE targets through human supervision in policy-induced states undercovered by open-loop demonstrations.

Mistake rollout



Post-trained actor (FIRE)



Figure 18: Black Bear Guai recovery case. The FIRE actor uses the Fireproof Mantle to withstand the wind-form phase. Frames in each row are ordered chronologically from left to right.