Anticipatory Planning for Multimodal AI Agents

Supplementary Material

A. Implementation Details

A.1. Training Hyperparameters

Table 4 summarizes the GRPO and optimization hyperparameters used in our experiments. Both Stage 1 (trajectory-level optimization) and Stage 2 (grounded fine-tuning) share identical training configurations; the only difference lies in their reward definitions.

A.2. Reward Formulation Details

Based on the provided implementation, the reward function in Stage 1 aims to align the predicted trajectory skeleton (action types and status) with the ground truth, while strictly enforcing output formatting. The total reward R for a predicted sample is a weighted sum of the accuracy reward R_{acc} and the format reward R_{fmt}:

$$R = (1 - \lambda_{\text{fmt}}) \cdot R_{\text{acc}} + \lambda_{\text{fmt}} \cdot R_{\text{fmt}} \tag{6}$$

where we set $\lambda_{\text{fmt}} = 0.1$.

Format Reward (R_{fmt}). To ensure the model generates parseable actions, we check for the presence of specific XML tags (e.g., `<think>`, `<answer>`) and JSON keys. For a generated response containing N steps, the format reward is the ratio of valid steps:

$$R_{\text{fmt}} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}[\text{Valid}(\hat{s}_i)] \tag{7}$$

A step $\hat{s}_i$ is considered valid only if it strictly contains the required keys: `"screenshot_abstraction"`, `"action"` (with `"action_type"`), and `"status"`.

Trajectory Alignment Accuracy (R_{acc}). Unlike standard exact matching, our trajectory-level reward focuses on the correctness of the *plan sequence* (Action Types). Let $\mathcal{A} = [\hat{a}_1, \ldots, \hat{a}_N]$ be the predicted action sequence and $\mathcal{A}^* = [a_1^*, \ldots, a_M^*]$ be the ground truth. We parse only the `action_type` and `status` fields for alignment.
(1) Greedy Alignment with Position Penalty. We compute the best alignment between $\mathcal{A}$ and $\mathcal{A}^*$. If lengths differ, we employ a greedy matching strategy. A predicted action $\hat{a}_i$ matches a ground truth action a_j^* if:

$$\text{sim}(\hat{a}_i, a_j^*) = \mathbb{1}[\hat{a}_i.\text{type} = a_j^*.\text{type}] \tag{8}$$

To encourage temporal consistency, we apply a position penalty $P_{\text{pos}} = |i - j| \times 0.1$. A match is accepted only if the adjusted score $(\text{sim} - P_{\text{pos}}) > 0.5$.

(2) Discounted Score. For the set of aligned pairs $\mathcal{M}$, the base alignment score is calculated using a discount factor $\gamma = 0.8$:

$$S_{\text{align}} = \sum_{(i,j) \in \mathcal{M}} \gamma^i \cdot \text{sim}(\hat{a}_i, a_j^*) \tag{9}$$

We subtract a coverage penalty of 0.15 for every unmatched action in both prediction and ground truth. The score is normalized by the maximum possible discounted return of the reference trajectory.
(3) Repetition Penalty. Finally, we penalize stuck loops (e.g., repeatedly predicting "click" without state change). If three consecutive actions have the same type, it counts as a repetition.

$$R_{\text{acc}} = \text{Clip}_{[0,1]} \left(\bar{S}_{\text{align}} - 0.1 \times N_{\text{repetitions}} \right) \tag{10}$$

where $\bar{S}_{\text{align}}$ is the normalized alignment score. This design encourages the agent to follow the correct high-level plan before refining parameters in Stage 2.

Category	Hyperparameter	Value
Actor Optimization	Learning rate	1×10^{-6}
	Optimizer	AdamW (bf16)
	Weight decay	0.01
	Warmup ratio	0
	Training steps	143
	Max grad norm	1.0
	Global batch size	128
GRPO / RL Parameters	Advantage estimator	GRPO
	Discount (γ)	0.8
	GAE λ	1.0
	KL type	fixed
	KL target	0.1
	KL coef	0.01
	KL penalty	low_var_kl
	Clip ratio (low)	0.2
	Clip ratio (high)	0.3
	Clip ratio (dual)	3.0
Rollout Generation	Engine	vLLM
	Number of rollouts (n)	5
	Rollout batch size	512
	Temperature	1.0
	Top-p	0.99
	Tensor parallel size	2
	Max batched tokens	8192
	Response length	2048
Reward Parameters	λ_{align}	0.8
	λ_{rep}	0.1
	λ_{format}	0.1

Table 4. **GRPO training hyperparameters used for both Stage 1 and Stage 2.** All reinforcement learning stages share the same GRPO and optimization settings.