TRACE-GRPO: Teacher-Refined Advantage Credit Estimation for Long-Horizon Agentic RL

Anonymous ACL submission

Abstract

Group-Relative Policy Optimization (GRPO) is widely used to post-train LLM agents, but its trajectory-level advantage is poorly matched to long-horizon credit assignment. In reward-homogeneous rollout groups, the vanishing variance collapses the advantage to zero, depriving the policy of any gradient update; conversely, when reward variance exists, a singular trajectory-level scalar uniformly scales all tokens, failing to isolate the specific decisions dictating the outcome. We propose **TRACE-GRPO** (**T**eacher-**R**efined **A**dvantage **C**redit **E**stimation for GRPO), which converts teacher-LLM feedback into hierarchical credit signals. At the *sample* level, teacher scores provide a fallback for zero-variance groups. At the *turn* level, trajectory advantages are reweighted according to teacher judgments of individual decisions. At the *token* level, turn-level credit is redistributed using the policy's own log-probability shift under a teacher critique, yielding a dense training-time signal without token-level annotations or learned token critics. We further introduce **CAPA** (Critique-Aware Paged Attention), which reuses trajectory KV caches and reduces token-level conditioning overhead from approximately $\sim 2\times$ to under 5% of a vanilla GRPO step. On ALFWorld and ScienceWorld with Qwen2.5-7B-Instruct, TRACE-GRPO achieves 96.3% and 73.4% success rates, respectively, outperforming strong agent RL baselines in success rate. Ablations show cumulative gains from the three credit levels, and comparison with supervised fine-tuning on teacher-generated successes confirms that critique-guided policy optimization provides benefits beyond imitation.

1 Introduction

Large language model (LLM) agents are increasingly deployed in interactive, long-horizon environments where success depends on a sequence of grounded decisions rather than a single response (Yao et al., 2023; Shridhar et al., 2021; Wang et al., 2022). While reinforcement learning is ideal for post-training such agents, credit assignment remains difficult due to sparse and delayed rewards. Group-Relative Policy Optimization (GRPO) (Shao et al., 2024) has emerged as an attractive alternative by normalizing rewards within sampled rollout groups, thereby bypassing an auxiliary value function. However, its trajectory-level scalar advantage is applied uniformly to every token, inducing two severe credit-assignment failures in long-horizon tasks. First, under reward homogeneity, group-normalized advantages collapse to zero, paralyzing the policy gradient precisely when local mistakes require guidance. Second, within reward-variant groups, a uniform scalar advantage is indiscriminately broadcasted across the entire response, updating erroneous action tokens with the same magnitude as punctuation, boilerplate, or formatting tokens. GRPO therefore misallocates gradient on incidental positions while under-crediting outcome-shaping decisions.

Prior work mitigates sparse rewards by introducing denser supervision. Process reward models and step-wise evaluators score intermediate reasoning (Uesato et al., 2022; Lightman et al., 2024), while LLM feedback methods leverage natural-language critiques to explain failures or guide revision (Bai et al., 2022; Madaan et al., 2023; Shinn et al., 2023; Wang et al., 2026; Song et al., 2026). Although these approaches demonstrate that teacher feedback captures subtle information missed by environment rewards, they ultimately compress this rich feedback into coarse trajectory- or turn-level scalars (Wang et al., 2026; Song et al., 2026). This compression discards critical localization, failing to exploit the specific object errors, missing preconditions, or invalid actions identified by the critique text. While direct token-level labels or learned token critics could preserve this

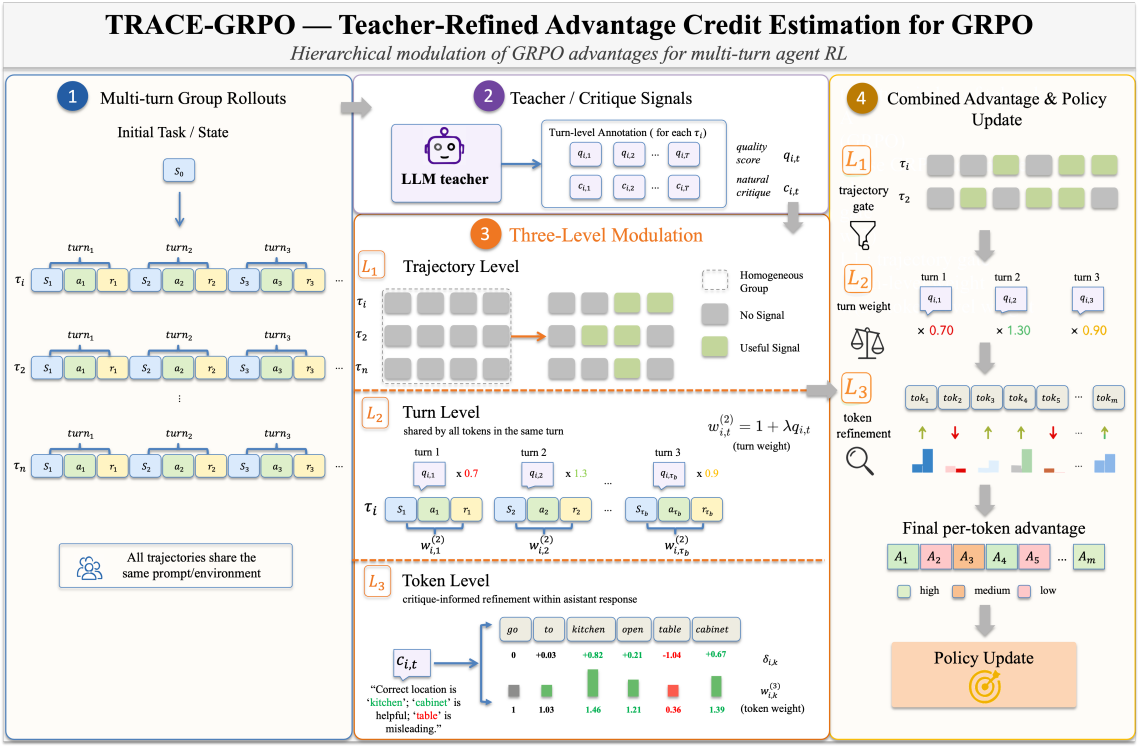

Figure 1: Overview of TRACE-GRPO. A teacher LLM provides turn-level quality scores and natural-language critiques. TRACE-GRPO converts this feedback into sample-level fallback advantages, turn-level reweighting, and token-level redistribution through the critique-induced log-probability shift δ.

structural granularity, they are costly to obtain, difficult to define for open-ended agent responses, and introduce additional modeling assumptions.

We propose TRACE-GRPO, a hierarchical extension that operationalizes natural-language teacher feedback into multi-granularity credit signals. Specifically, sample-level fallback scores rescue zero-variance groups, turn-level judgments reweight individual decisions, and token-level credit is dynamically assigned via critique-induced log-probability shifts. The latter yields dense reward signals without requiring token-level labels or learned token critics. To render token scoring efficient, we develop CAPA (Critique-Aware Paged Attention), which reuses trajectory prefix KV caches to reduce training overhead to under 5%. Evaluated with Qwen2.5-7B-Instruct, TRACE-GRPO achieves 96.3% success on ALFWorld (+5.5 pt over GiGPO; +18.7 pt over GRPO) and 73.4% on ScienceWorld (+24.3 pt over GRPO). Detailed analyses confirm that token-level updates precisely concentrate on decision-critical spans while reinforcing optimization stability. Our contributions are:

1. We propose **TRACE-GRPO**, a hierarchical advantage estimator that integrates teacher feedback at the sample, turn, and token levels. Its token-level signal uses critique-induced log-probability shifts to extract dense credit from natural-language critiques without token labels or a learned token critic.
2. We introduce **CAPA**, an efficient training-time implementation that reuses KV caches and paged attention to make critique-conditioned token scoring practical.
3. We demonstrate consistent improvements on ALFWorld and ScienceWorld, with analyses showing that token-level modulation aligns with teacher-identified errors and improves optimization stability.

2 Related Work

GRPO and sub-trajectory credit assignment. GRPO (Shao et al., 2024) replaces a learned value function with group-normalized rewards, simplifying large-scale RL for reasoning and agentic LLMs. Recent variants improve stability through normalization, filtering, or baseline estimation, including DR-GRPO (Liu et al., 2025) and StarPO-S (Wang et al., 2025), but still rely on trajectory-level advantages. Other methods refine credit assignment

within trajectories: GiGPO (Feng et al., 2026) uses shared-prefix rollout structure to derive per-state advantages, and ArCHer (Zhou et al., 2024) uses a hierarchical actor-critic formulation. In this context, TRACE-GRPO is complementary: it preserves GRPO's value-free structure while using teacher critiques and policy distribution shifts to refine credit at the sample, turn, and token levels.

Teacher feedback and token-level credit. RLHF (Ouyang et al., 2022) and RLAIF (Bai et al., 2022) obtain preference or reward signals from human or AI feedback, typically at the response level. Process-reward and textual-feedback methods provide denser supervision by scoring intermediate steps or turns (Wang et al., 2026), but often compress critiques into scalar rewards over coarse segments (Song et al., 2026). This loses localization information in the teacher's text, such as the wrong object, missing precondition, or invalid action. TRACE-GRPO keeps scalar teacher scores for sample- and turn-level modulation, while using critique-induced log-probability shifts to obtain token-level credit without token labels or a learned token critic.

3 Method

3.1 Problem Setup

We consider multi-turn agent RL. At turn t, the policy π_θ generates an action a_t, represented as a sequence of tokens, conditioned on the history

$$h_t = (o_1, a_1, o_2, \ldots, o_{t-1}, a_{t-1}, o_t)$$

where o_t is the environment observation. The environment returns the next observation o_{t+1} and a terminal reward R at the end of the episode. A trajectory τ is the resulting interleaved sequence of observations and generated actions.

GRPO (Shao et al., 2024) samples N trajectories for each prompt and normalizes rewards within the rollout group:

$$A_{\text{GRPO}}(\tau_i) = \frac{R(\tau_i) - \mu_G}{\sigma_G + \varepsilon_{\text{std}}} \tag{1}$$

for $i \in \{1, \ldots, N\}$, where μ_G and σ_G are the group mean and standard deviation, and $\varepsilon_{\text{std}} > 0$ is a numerical-stability constant. The resulting trajectory-level advantage is broadcast to every generated response token:

$$A(i,k) = A_{\text{GRPO}}(\tau_i)\, m(i,k)$$

where $m(i,k) \in \{0,1\}$ masks assistant-generated response tokens. This trajectory-level design induces two credit-assignment failures in long-horizon agents. First, under reward homogeneity, group-normalized advantages collapse to zero, paralyzing the policy gradient. Second, within reward-variant groups, a singular advantage scalar is indiscriminately applied to all constituent tokens, blinding the model to the causal locus of individual decisions. To bridge this gap, TRACE-GRPO retains GRPO's value-free architecture while refining advantages across three structural granularities: sample, turn, and token.

3.2 Teacher Annotation Protocol

After each rollout, a frozen teacher LLM T annotates the trajectory using the prompt in Appendix E. The teacher is instructed to label only turns whose decisions are evidence-grounded and materially affect the outcome. Routine or ambiguous turns are omitted.

For trajectory τ_b, the teacher returns a sparse set of labeled turns:

$$T(\tau_b) = \{(d_{b,t}, c_{b,t}, r_{b,t})\}_{t \in \mathcal{T}_b} \tag{2}$$

where $\mathcal{T}_b$ is the set of annotated turns, $d_{b,t} \in \{-1.0, -0.7, -0.3, 0.3, 0.7, 1.0\}$ is the decision-quality score, $c_{b,t} \in \{0.5, 0.75, 1.0\}$ is the teacher confidence, and $r_{b,t}$ is a concise rationale. Neutral turns are omitted rather than assigned a zero label, which keeps teacher output focused on outcome-relevant decisions. We define the signed turn signal

$$q_{b,t} = d_{b,t} c_{b,t} \in [-1, 1] \tag{3}$$

for $t \in \mathcal{T}_b$. For unlabeled turns, we set $q_{b,t} = 0$, corresponding to teacher abstention. The rationale $r_{b,t}$ is used only as a training-time conditioning signal for Level 3. The teacher also returns a trajectory-level success judgment for diagnostics, but this flag is not used in the policy-gradient objective.

3.3 Level 1: Zero-Variance Fallback

Level 1 addresses reward-homogeneous rollout groups. When the group reward variance is below the numerical threshold, GRPO provides no learning signal. We replace the zero trajectory advantage with a teacher-derived turn signal:

$$A^{(1)}(b,k) = \begin{cases} \lambda\, q_{b,\iota_b(k)}\, m(b,k) & \sigma_G < \varepsilon_{\text{std}} \\ A_{\text{GRPO}}(\tau_b)\, m(b,k) & \text{otherwise} \end{cases} \tag{4}$$

where $\iota_b(k)$ maps token k to its containing turn, and $\lambda > 0$ controls the fallback scale. If the teacher abstains on a turn, then $q_{b,\iota_b(k)} = 0$ and Level 1 assigns no advantage to tokens in that turn. Thus, Level 1 recovers a learning signal only where the teacher identifies outcome-relevant decisions.

3.4 Level 2: Turn-Level Critique Modulation

Level 2 reweights the update at turn granularity according to whether the teacher agrees with the GRPO direction. Let

$$s_b = \mathrm{sgn}(A_{\mathrm{GRPO}}(\tau_b)) \in \{-1, 0, +1\}$$
$$a_{b,k} = s_b q_{b,\iota_b(k)}$$

A positive $a_{b,k}$ means the teacher's assessment of the turn agrees with the trajectory-level update direction; a negative value means the teacher dissents. The turn-level weight and advantage are

$$w_{\mathrm{crit}}(b,k) = \mathrm{clip}(1 + \alpha a_{b,k},\ 0,\ 1 + \alpha) \tag{5}$$
$$A^{(2)}(b,k) = A^{(1)}(b,k)\, w_{\mathrm{crit}}(b,k) \tag{6}$$

where $\alpha \geq 0$ controls modulation strength. Agreement amplifies the update, while dissent dampens it. The lower clamp prevents the teacher signal from reversing the sign of the GRPO update. For zero-variance groups, $s_b = 0$, so $w_{\mathrm{crit}}(b,k) = 1$ and the Level 1 fallback passes through unchanged. Tokens associated with environment-detected format errors skip this reweighting and use $w_{\mathrm{crit}}(b,k) = 1$, preserving the environment penalty independent of teacher leniency.

3.5 Level 3: Critique-Conditioned Token-Level Reweight

3.5.1 Definition

Levels 1 and 2 operate at turn granularity. Level 3 further redistributes credit within each annotated response by measuring how the policy's own token probabilities change when conditioned on the teacher rationale. For turn t and response token $y_{b,t,j}$, we compute

$$\delta_{b,t,j} = \mathrm{sg}\Big[\log \pi_\theta\big(y_{b,t,j} \mid h_{b,t}, r_{b,t}, y_{b,t,<j}\big) - \log \pi_\theta\big(y_{b,t,j} \mid h_{b,t}, y_{b,t,<j}\big)\Big] \tag{7}$$

where $h_{b,t}$ is the turn history and $j = 1, \ldots, L_{\mathrm{resp}}(b,t)$. The operator $\mathrm{sg}[\cdot]$ denotes stop-gradient. In implementation, δ is computed once under the rollout policy and treated as a fixed credit-assignment signal during the policy update. For unlabeled turns, we set the Level 3 weight to one and do not compute δ.

The magnitude $|\delta_{b,t,j}|$ measures how strongly the rationale affects the likelihood of the realized token. Its sign indicates whether the rationale increases or decreases that likelihood. To align token reweighting with the current update direction, denote

$$u_{b,k} = \mathrm{sgn}(A^{(2)}(b,k))$$

For token k in turn $\iota_b(k)$ and within-turn position $j_b(k)$, the token weight is

$$w_{b,k} = 1 + \alpha_3 u_{b,k} |q_{b,\iota_b(k)}|\, \mathrm{clip}\big(\delta_{b,\iota_b(k),j_b(k)}, -\kappa, \kappa\big) \tag{8}$$

where $\alpha_3 \geq 0$ is the token-level modulation strength and $\kappa > 0$ clips the log-probability shift. We choose $\alpha_3 \kappa < 1$, which ensures $w_{b,k} > 0$ for all response tokens and prevents sign flips of the underlying advantage.

Assuming the trajectory contains at least one response token, the Level 3 advantage is

$$A^{(3)}(b,k) = A^{(2)}(b,k) \frac{w_{b,k}}{\bar{w}_\delta(b)} \tag{9}$$

where

$$\bar{w}_\delta(b) = \frac{\sum_k w_{b,k} m(b,k)}{\sum_k m(b,k)} \tag{10}$$

The row-mean normalization makes Level 3 a credit-redistribution operator: it changes where the update is concentrated within a trajectory, while limiting changes to the trajectory-level scale.

3.5.2 Properties

Level 3 is designed as a controlled token-level redistribution rather than an unconstrained rescaling of the trajectory advantage. Let $N_b = \sum_k m(b,k) > 0$, and let Cov_m and Var_m denote covariance and variance under the uniform distribution over masked response tokens. Writing $A_k = A^{(2)}(b,k)$ and $w_k = w_{b,k}$, the row-mean normalization in Eq. (9) gives

$$\sum_k A^{(3)}(b,k) m(b,k) = \sum_k A_k m(b,k) + \frac{N_b \,\mathrm{Cov}_m(A,w)}{\bar{w}_\delta(b)} \tag{11}$$

Thus, Level 3 exactly preserves the masked trajectory total whenever $\mathrm{Cov}_m(A,w) = 0$. This includes the case where $A^{(2)}(b,\cdot)$ is constant over

response tokens, which covers broadcast advantages and trajectories whose turn-level multiplier is uniform.

When the covariance is nonzero, the deviation is bounded. Let $a = \alpha_3 \kappa$. Since $|w_k - 1| \leq a$ and $\bar{w}_\delta(b) \geq 1 - a > 0$, we have

$$\left| \sum_k A^{(3)} m - \sum_k A^{(2)} m \right| \leq \frac{N_b a}{1-a} \cdot \sqrt{\mathrm{Var}_m(A^{(2)})} \tag{12}$$

The derivation is provided in Appendix A. In our experiments, $a = 0.3$, so Level 3 remains close to pure redistribution while allowing token-specific amplification and damping.

Soft fallback. If the teacher abstains on all turns, then $q_{b,t} = 0$, $w_{b,k} = 1$, and $A^{(3)} = A^{(2)}$. More generally, if $w_{b,k}$ is constant over the response-token mask, the normalization in Eq. (9) cancels exactly. Level 3 therefore degrades gracefully when the critique contains no token-specific information.

Ablation chain. Setting $\alpha_3 = 0$ recovers the L1+L2 variant. Setting the Level 2 coefficient to zero further recovers L1. Setting the Level 1 fallback coefficient to zero recovers vanilla GRPO, including its zero gradient on reward-homogeneous rollout groups. Each level can therefore be disabled independently.

Policy update. All teacher-derived quantities, including q, δ, and the resulting advantages, are detached during optimization. We replace the GRPO advantage in the standard clipped policy-gradient loss with $A^{(3)}$:

$$\mathcal{L}_{\mathrm{PG}} = -\mathbb{E}_{b,k}\Big[m(b,k) \min\big(\rho_{b,k} A^{(3)}(b,k), \rho^{\epsilon}_{b,k} A^{(3)}(b,k)\big)\Big] \tag{13}$$

where $\rho_{b,k}$ is the policy ratio, $\pi_\theta(y_{b,k} \mid h_{b,k}) / \pi_{\mathrm{old}}(y_{b,k} \mid h_{b,k})$, and $\rho^{\epsilon}_{b,k}$ denotes its clipped value, $\mathrm{clip}(\rho_{b,k}, 1-\epsilon, 1+\epsilon)$. The KL penalty and optimizer details follow Appendix B.

3.6 CAPA: Efficient Critique-Conditioned Forwards

Naively computing the critique-conditioned term in Eq. (7) requires one extra full forward per annotated turn. CAPA (Critique-Aware Paged Attention) reduces this cost by reusing the trajectory prefix KV cache from the standard log-probability pass. Since annotated turns share nested prefixes, CAPA treats the trajectory cache as a paged store (Kwon et al., 2023): shared prefix pages are reused, while each turn appends a private stub containing its rationale and response tokens. A single batched FlashAttention-2 paged call (Dao, 2024) then evaluates all critique-conditioned continuations using per-turn `block_table` and `cu_seqlens` indices (Figure 2).

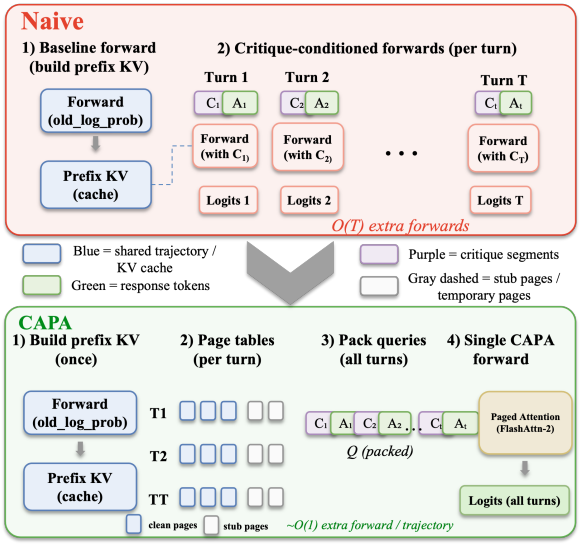

Figure 2: CAPA architecture. The trajectory KV cache is partitioned into shared prefix pages. Each annotated turn appends a private stub containing its rationale and response tokens. Per-turn `block_table` indices concatenate shared pages with the stub page, allowing all critique-conditioned continuations to be evaluated in one paged-attention call. Due to space constraints, a simplified illustration is shown here; the full, detailed architecture diagram is provided in Appendix D.

CAPA matches the naive conditioned forward within an absolute tolerance ($\mathrm{atol} = 10^{-5}$) in FP32. With BF16 and FlashAttention-2, the maximum observed numerical drift is below $\mathrm{atol} = 4 \times 10^{-3}$, well below typical critique-induced shifts. On Qwen2.5-7B-Instruct with 8 GPUs, CAPA reduces Level-3 forward cost by 12.7× and keeps the full training step within 5% overhead relative to vanilla GRPO. Implementation details and FSDP integration are in Appendix D.

4 Experiments

4.1 Setup

We evaluate TRACE-GRPO using Qwen2.5-7B-Instruct on two long-horizon agent benchmarks, ALFWorld (Shridhar et al., 2021) and ScienceWorld (Wang et al., 2022), strictly following the GiGPO (Feng et al., 2026) protocol. Train-

ing is conducted on 8×H100 GPUs with a frozen DeepSeek-V3.1 teacher guiding the turn-level critique derivation. Full implementation details (including hyperparameters, rollout settings, and dataset construction) are deferred to Appendix B. We compare TRACE-GRPO against leading closed-source models, prompting agents, and strong reinforcement learning baselines (including PPO (Schulman et al., 2017), RLOO (Ahmadian et al., 2024), vanilla GRPO (Shao et al., 2024), and GiGPO (Feng et al., 2026)). To isolate the gains of RL optimization from mere teacher imitation, we also evaluate **Student-SFT** and **Teacher-Prompt** controls. Detailed descriptions of all baselines are deferred to Appendix B.

4.2 Main Results

As shown in Table 1, TRACE-GRPO achieves **96.3**% overall success on ALFWorld, improving over GiGPO+std (90.8%) by +5.5 points and over vanilla GRPO by +18.7 points. The largest gains over GiGPO+std occur on decision-intensive subtasks, including Look (+17.3 points), Pick2 (+14.9 points), Heat (+12.0 points), and Cool (+10.7 points). On ScienceWorld, TRACE-GRPO obtains a score of **70.6** and a success rate of **73.4**%, exceeding vanilla GRPO by +8.8 score points and +24.3 success points, and exceeding GiGPO+std by +1.4 score points and +20.0 success points. Overall, the results indicate that converting teacher feedback into hierarchical credit signals improves long-horizon agent training beyond standard group-relative reward normalization.

SFT vs. RL. Teacher-Prompt obtains 57.1% ALFWorld success, and Student-SFT obtains 61.4%, both far below TRACE-GRPO at 96.3%. The same pattern appears on ScienceWorld, where Teacher-Prompt and Student-SFT remain substantially below RL-trained agents. These results show that successful teacher demonstrations alone do not explain the gain. The student must also explore its own failure modes and receive critique-guided correction, which is provided by RL but not by success-only imitation.

4.3 Teacher Feedback Analysis

Figure 4 analyzes DeepSeek-V3.1 feedback during ALFWorld training. The teacher's trajectory judgment agrees with the environment outcome above 95%, indicating reliable supervision. Its feedback also adapts as the student improves: negative labels dominate early training, become majority positive by iteration 100, and the dominant aspect shifts from format errors to action choice. Thus, the teacher provides an adaptive process signal rather than a static outcome label.

4.4 Ablation Study

The bottom rows of Table 1 isolate the three levels of TRACE-GRPO. On ALFWorld, the gains over vanilla GRPO increase from +1.0 point with L1 to +12.7 points with L1+L2 and +18.7 points with L1+L2+L3, showing cumulative benefits under sparse binary rewards. On ScienceWorld, L1 has little effect on success (49.5% vs. 49.1% for GRPO), as dense task scores reduce zero-variance groups. L2 raises success to 80.5%, while L3 increases the dense task score from 50.8 to 70.6 and remains far above GRPO in success (73.4% vs. 49.1%). This suggests that turn-level feedback improves coarse decision selection, whereas token-level reweighting improves fine-grained task progress and policy efficiency.

Where L3 helps. To locate L3's benefit, we stratify the 128-task split by GRPO per-template score (Figure 6a). On the hardest tertile, where GRPO averages 12, L3 reaches **78**, a +59 point gain over L1+L2. On the easiest tertile, where GRPO already scores 70, the variants are close. Among tasks solved by both L1+L2 and L1+L2+L3 (Figure 6b), L3 achieves a higher goal score (87.1 vs. 56.8) with fewer turns (8.2 vs. 15.8). Over the full validation pool, L3 also shortens episodes to 9.2 turns, compared with 15.2 for L1+L2 (Figure 6). These results indicate that L3 improves difficult-task progress and efficiency rather than relying on longer interaction sequences.

4.5 Training Dynamics

Figure 3 compares TRACE-GRPO with our reproduced GRPO over 150 ALFWorld iterations. TRACE-GRPO reaches 96.3% validation success, compared with 69.5% for GRPO. At convergence, it reduces gradient norm by 76% (2.93 to 0.72), yields 13.6× lower KL divergence from the reference policy (0.028 vs. 0.381), and shortens mean episode length by 64% (29.4 to 10.6 steps). These trends show that hierarchical credit assignment improves success while keeping the policy closer to the reference and producing shorter trajectories.

Type	Method	ALFWorld							ScienceWorld	
		Pick	Look	Clean	Heat	Cool	Pick2	All	Score	Succ
Closed-source prompting baselines										
Closed-src	GPT-4o (ReAct) (Hurst et al., 2024)	75.3	60.8	31.2	56.7	21.6	49.8	48.0	54.3	45.4
Closed-src	Gemini-2.5-Pro (ReAct) (Comanici et al., 2025)	92.8	63.3	62.1	69.0	26.6	58.7	60.3	42.5	35.9
Teacher-only control										
Teacher	DeepSeek-V3.1 Prompt	82.86	69.23	51.85	31.25	36.00	58.33	57.14	13.05	21.88
Base model: Qwen2.5-7B-Instruct										
Prompt	Direct Prompt (Yang et al., 2024)	33.4	21.6	19.3	6.9	2.8	3.2	14.8	11.4	4.2
Prompt	ReAct (Yao et al., 2023)	48.5	35.4	34.3	13.2	18.2	17.6	31.2	17.4	7.8
Prompt	Reflexion (Shinn et al., 2023)	62.0	41.6	44.9	30.9	36.3	23.8	42.7	23.4	11.7
SFT	Student-SFT (Qwen2.5-7B-Instruct ← DeepSeek-V3.1)	62.86	53.85	59.26	75.00	48.00	70.83	61.43	19.09	14.06
RL	PPO (critic) (Schulman et al., 2017)	92.3	64.0	92.5	89.5	80.3	68.8	80.4	37.1	24.0
RL	RLOO (Ahmadian et al., 2024)	87.6	78.2	87.3	81.3	71.9	48.9	75.5	–	–
RL	GRPO (Shao et al., 2024)	90.8	66.1	89.3	74.7	72.5	64.7	77.6	61.8	49.1
RL	GiGPO+std (Feng et al., 2026)	**97.7**	82.7	**98.8**	83.7	89.3	79.2	90.8	69.2	53.4
RL	GiGPO-std (Feng et al., 2026)	91.8	88.6	95.9	90.2	86.5	85.2	90.2	–	–
TRACE-GRPO ablations										
Ours	TRACE-GRPO (L1 only)	91.5	67.8	89.7	76.0	74.1	66.5	78.6	62.0	49.5
Ours	TRACE-GRPO (L1+L2)	94.5	91.7	92.0	87.5	92.4	83.6	90.3	50.8	**80.5**
Ours	**TRACE-GRPO (L1+L2+L3)**	95.8$_{+5.0}$	**100.0**$_{+33.9}$	93.6$_{+4.3}$	**95.7**$_{+21.0}$	**100.0**$_{+27.5}$	**94.1**$_{+29.4}$	**96.3**$_{+18.7}$	**70.6**$_{+8.8}$	73.4$_{+24.3}$

Table 1: Main results on ALFWorld and ScienceWorld using Qwen2.5-7B-Instruct as the student backbone. ALFWorld reports per-subtask and overall success (%); ScienceWorld reports task score and success (%). Bold indicates the best result; gains in the final row are absolute improvements over vanilla GRPO; N/A denotes unavailable matched-protocol results.

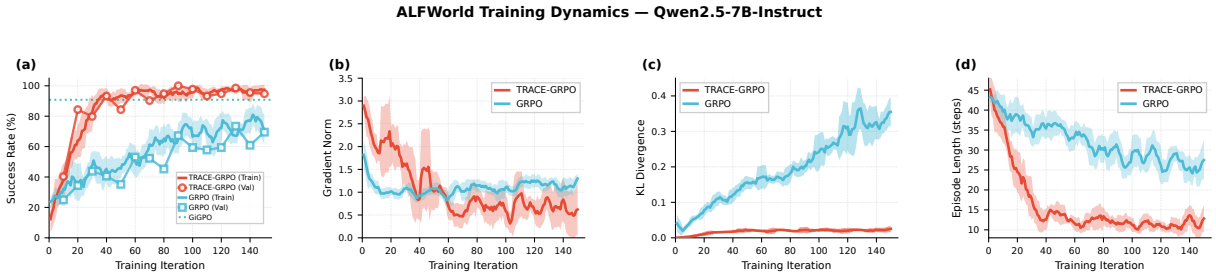

Figure 3: ALFWorld training dynamics with Qwen2.5-7B-Instruct on 8×H100 over 150 iterations. **(a)** Train and validation success rates. The dashed line marks the GiGPO reported performance of 90.8%. **(b)** Gradient norm. **(c)** KL divergence from the reference policy. **(d)** Mean episode length. Shaded regions denote $\pm 1\sigma$ within a sliding window.

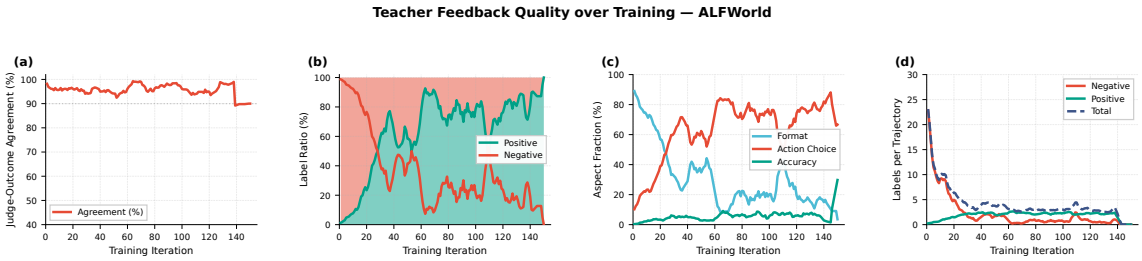

Figure 4: DeepSeek-V3.1 feedback over ALFWorld training. **(a)** Teacher outcome agreement. **(b)** Positive and negative label ratio. **(c)** Dominant feedback aspect. **(d)** Mean labels per trajectory.

4.6 Interpretability of Token-Level Shifts

To examine what L3 modulates, we analyze the step-150 checkpoint on 20 ALFWorld valid-unseen trajectories, covering 302 action turns and 2,289 response tokens. For each token, we compute $\delta = \log \pi(a \mid h, c) - \log \pi(a \mid h)$ and group tokens by semantic category. Figure 5 shows that δ concentrates on decision-critical positions. Format boundary tokens have the largest mean $|\delta|$ (0.76), reflecting whether the critique changes the model's confidence in producing an action response. Among content tokens, object nouns receive the strongest modulation (mean $|\delta| = 0.084$, P95 $= 0.44$), followed by action verbs (mean $|\delta| = 0.028$, P95 $= 0.18$). In contrast, prepositions and whitespace have near-zero shifts ($< 10^{-4}$), indicating that L3

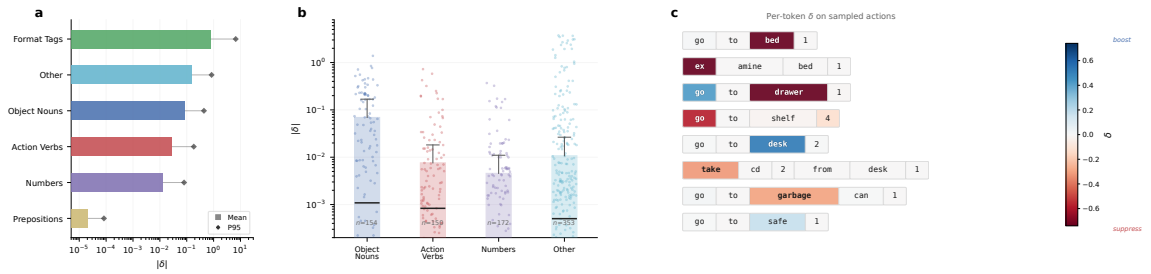

Figure 5: L3 token-level δ modulation analysis on ALFWorld (step 150, $n = 2{,}289$ tokens). **(a)** Mean and P95 of $|\delta|$ by semantic token category. **(b)** Per-token $|\delta|$ distribution for content categories. **(c)** Qualitative examples with token-level δ overlaid on sampled actions. Red indicates suppression and blue indicates boosting.

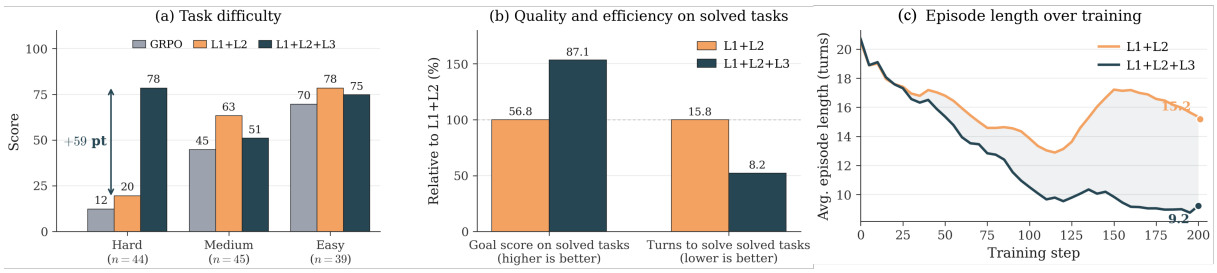

Figure 6: ScienceWorld L3 ablation and efficiency. **(a–b)** Task score, goal score, and trajectory length under L3 ablations. **(c)** Mean validation episode length; L1+L2+L3 converges to 9.2 turns, compared with 15.2 for L1+L2.

does not uniformly rescale all tokens. Qualitative examples further show that location nouns such as *bed*, *drawer*, and *desk* are strongly suppressed when implicated by the critique, while function words remain largely unchanged.

4.7 Efficiency

The critique-conditioned shift δ in Eq. (7) requires an additional forward computation. Figure 7 shows that CAPA makes this step practical by reusing trajectory-prefix KV caches and batching critique-conditioned continuations with paged attention. On Qwen2.5-7B-Instruct with 8-GPU FSDP, CAPA reduces the Level-3 forward overhead from 178 seconds per PPO step under the naive implementation to 14 seconds, a 12.7× speedup. The resulting TRACE-GRPO step remains within 5% of vanilla GRPO wall time, showing that token-level critique conditioning can be added with modest training overhead.

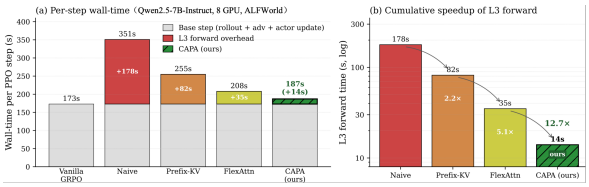

Figure 7: Level-3 critique-conditioned forward cost. **(a)** Per-step wall time, including the base GRPO step and the Level-3 forward overhead. **(b)** Cumulative Level-3 speedup on a log scale.

5 Conclusion

We have presented TRACE-GRPO, a value-free hierarchical advantage estimator that operationalizes natural-language teacher critiques into multi-granularity credit signals across the sample, turn, and token levels. By measuring critique-induced log-probability shifts, TRACE-GRPO extracts dense, on-policy token credit without token-level labels or learned critics, while our CAPA mechanism constrains the training overhead to under 5%. Empirical evaluations on ALFWorld and ScienceWorld demonstrate that TRACE-GRPO substantially outperforms standard GRPO and state-of-the-art agent RL baselines, with progressive ablations validating the independent efficacy of each hierarchical level. Furthermore, the performance gap over SFT highlights the unique value of critique-guided exploration over imitation. Future directions include implementing self-critique mechanisms, adapting the framework to mathematical and programming domains, and extending token-level modulation to actor-critic architectures.

Limitations

While TRACE-GRPO significantly improves fine-grained credit assignment for long-horizon LLM agents, we identify a few natural avenues for future exploration:

Teacher Reliance and Domain Scope. Our hierarchical modulation utilizes natural-language critiques from an external teacher LLM. While our sparse annotation protocol ensures high-quality conditioning signals, future work could explore self-critique mechanisms to eliminate reliance on external oracles and further reduce annotation costs. Additionally, while we demonstrate strong efficacy in semantic, text-based environments (ALFWorld and ScienceWorld), systematically extending this critique-induced framework to multi-modal domains (e.g., Vision-Language-Action models) or strictly algorithmic tasks presents an exciting direction for future research.

Architectural and Efficiency Extensions. Algorithmically, TRACE-GRPO leverages the critic-free GRPO framework to maintain high memory efficiency during post-training. In environments with extreme stochasticity, integrating our token-level redistribution with standard actor-critic architectures (e.g., PPO) could provide complementary variance reduction. Finally, while our CAPA mechanism successfully restricts the critique-conditioning time overhead to under 5%, it introduces a marginal GPU memory footprint for managing paged block tables, which could be further optimized in future distributed training deployments.

References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12248–12267.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, and 32 others. 2022. Constitutional AI: Harmlessness from AI feedback. *Preprint*, arXiv:2212.08073.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Tri Dao. 2024. Flashattention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations*, volume 2024, pages 35549–35562.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. 2026. Group-in-group policy optimization for llm agent training. *Advances in Neural Information Processing Systems*, 38:46375–46408.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pages 39578–39601.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Understanding r1-zero-like training: A critical perspective. In *Second Conference on Language Modeling*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan

Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2021. ALFWorld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*.

Yuda Song, Lili Chen, Fahim Tajwar, Rémi Munos, Deepak Pathak, Drew Bagnell, Aarti Singh, and Andrea Zanette. 2026. Expanding the capabilities of reinforcement learning via text feedback. In *The 1st Workshop on Scaling Post-training for LLMs*.

Jonathan Uesato, Nate Kushman, Ramana Kumar, H Francis Song, Noah Yamamoto Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process-based and outcome-based feedback.

Hanyang Wang, Lu Wang, Chaoyun Zhang, Tianjun Mao, Si Qin, Qingwei Lin, Saravan Rajmohan, and Dongmei Zhang. 2026. Text2grad: Reinforcement learning from natural language feedback. In *The Fourteenth International Conference on Learning Representations*.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. 2022. Scienceworld: Is your agent smarter than a 5th grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11279–11298.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, and 1 others. 2025. Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*.

Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, and 25 others. 2024. Qwen2.5 technical report. *ArXiv*, abs/2412.15115.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*.

Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. 2024. Archer: Training language model agents via hierarchical multi-turn RL. In *Forty-first International Conference on Machine Learning*.

A Proof of Level-3 Redistribution Bound

Let $A_k = A^{(2)}(b,k)$, $w_k = w_{b,k}$, and let expectations, variances, and covariances be taken uniformly over masked response tokens. Since $\bar{w}_\delta = \mathbb{E}_m[w]$,

$$\sum_k A^{(3)}(b,k)m(b,k) = \frac{N_b \mathbb{E}_m[Aw]}{\mathbb{E}_m[w]}$$
$$= \sum_k A_k m(b,k) + \frac{N_b \operatorname{Cov}_m(A,w)}{\bar{w}_\delta}$$

If $\operatorname{Cov}_m(A,w) = 0$, the masked total is preserved exactly. Otherwise, with $a = \alpha_3\kappa$, the clipped weight satisfies $|w_k - 1| \le a$ and $\bar{w}_\delta \ge 1 - a > 0$. Therefore,

$$\left|\sum_k A^{(3)}m - \sum_k A^{(2)}m\right|$$
$$\le \frac{N_b}{\bar{w}_\delta}\sqrt{\operatorname{Var}_m(A)}\sqrt{\operatorname{Var}_m(w)}$$
$$\le \frac{N_b a}{1-a}\sqrt{\operatorname{Var}_m(A^{(2)})}$$

B Implementation Details

Hardware and software. All training runs use 8×NVIDIA H100 (80 GB) on a single node, with FSDP for the actor and reference, vLLM (FlashAttention-2 backend) for rollout, and Ray for orchestration. We use `verl-agent` (GiGPO (Feng et al., 2026) fork) as the trainer scaffold and Qwen2.5-7B-Instruct as the base policy. The teacher (DeepSeek-V3.1) is queried over an OpenAI-style HTTP API with up to 64 concurrent workers per training step.

Hyperparameters. Table 2 lists the hyperparameters used for both benchmarks. The evaluation settings for both ALFWorld and ScienceWorld strictly follow the GiGPO (Feng et al., 2026) protocol where applicable. TRACE-GRPO-specific hyperparameters (λ, α, α_3, κ, σ_ϵ) were fixed early in development on a small ALFWorld sweep and reused on ScienceWorld without further tuning.

Teacher annotation. At each training step, the DeepSeek-V3.1 teacher is queried at temperature 0 with up to 4096 judge tokens per call and a 240 s timeout. The schema returns, for each turn, a decision quality $\in \{-1, 0, +1\}$, a confidence $\in [0,1]$, a categorical aspect, and a free-form rationale; we set q_t to their product (signed confidence). Format-error turns flagged by the environment are pinned to $q_t = 0$ and bypass L2/L3 modulation (§3.4).

	ALFWorld	ScienceWorld
Optimization		
Learning rate	1×10^{-6}	1×10^{-6}
KL loss coefficient	0.01	0.01
KL penalty type	low-var KL	low-var KL
Entropy coefficient	0	0.001
PPO mini-batch size	16	16
PPO micro-batch / GPU	2	2−4
Invalid-action penalty	0.1	0.1
Rollout		
Group size N	8	8
Train batch size	16	16
Val batch size	128	128
Max env steps	50	30
Max prompt length	8192	7000
Max response length	24576	16384
Train rollout temp.	1.0	1.0
Eval rollout temp.	0.4	0.4
TRACE-GRPO		
λ (L1 fallback)	0.2	0.2
σ_ϵ (L1 trigger)	10^{-3}	10^{-3}
α (L2 turn-level)	0.5	0.5
α_3 (L3 token-level)	0.3	0.3
κ (L3 clamp)	1.0	1.0
Trainer		
Total iterations	150	200
Eval frequency	every 5 iters	every 5 iters
Checkpoint frequency	every 20 iters	every 20 iters

Table 2: Hyperparameters for TRACE-GRPO on ALFWorld and ScienceWorld.

Evaluation protocol. ALFWorld evaluation uses the GiGPO protocol: 1 trajectory per task on the 134-game valid-unseen split, eval rollout temperature 0.4, max 50 environment steps, success counted as the binary `won` flag returned by the textworld engine. ScienceWorld evaluation uses the protocol following : 128 tasks stratified across the 29 task types in the held-out pool, 1 trajectory per task, max 30 steps, task score is defined as the mean of $\max(0, \text{score})$ over the 128 tasks where score $\in [-100, 100]$ is the engine's progress signal.

C SFT and Teacher-Prompt Baselines

SFT data construction. For each benchmark we collect successful trajectories produced by the DeepSeek-V3.1 teacher under the same prompt template used during RL rollout. We sample $K = 8$ rollouts per training task, retain only trajectories with `won`$= 1$ (ALFWorld) or final `score`≥ 50 (ScienceWorld), and dedupe by trajectory hash. The two pools are merged into a multi-turn SFT parquet keyed by `messages`, yielding roughly 5× the training-pool size of the RL set after deduplication.

SFT training. We fine-tune Qwen2.5-7B-Instruct with FSDP for 3 epochs at learning rate 1×10^{-4} (LoRA rank 16, $\alpha = 32$), per-device batch 1 with gradient accumulation 16, max sequence length 2048. Training uses the same `verl` multi-turn SFT trainer as the RL stack so the chat template and tokenization are identical to inference time. Evaluation reuses the same protocol as the RL runs (Appendix B).

Teacher-Prompt baseline. Teacher-Prompt evaluates DeepSeek-V3.1 directly on each benchmark's held-out split with no fine-tuning, using the same system prompt, action template, and environment as the student. Generation is deterministic ($T = 0$) with a 4096-token budget per turn and the same step limits as RL evaluation.

D CAPA Implementation Details

This appendix gives the full implementation of CAPA, the optimization from §3.6 that brings the critique-conditioned $\delta_{b,t,j}$ in Eq. (7) from one extra full forward per annotated turn down to a single packed forward per trajectory. The key observation is that the standard PPO log-probability pass already runs a full forward over the trajectory x; CAPA re-uses that pass's KV cache as the read-only *shared prefix* for all N annotated turns and only adds a packed forward over the new critique + response tokens. We instantiate this idea on top of vLLM's paged-KV abstraction (Kwon et al., 2023) and FlashAttention-2's varlen paged kernel (Dao, 2024).

D.1 Notation and turn layout

A trajectory is a token sequence $x = [s_0, a_1, s_1, \ldots, s_T, a_T]$ of length L_τ. For each annotated turn $t \in \{1, \ldots, N\}$ we tokenize the teacher's rationale to $\mathbf{c}_t = (c_{t,1}, \ldots, c_{t,|c_t|})$. The conditioned input that the reference (slow) implementation feeds the model for turn t is

$$\Big[\, x_{0:\mathrm{ctx_end}_t},\ \langle\text{teacher_note}\rangle\ \mathbf{c}_t\ \langle/\text{teacher_note}\rangle,\ x_{\mathrm{resp_start}_t:\mathrm{resp_end}_t} \,\Big],$$

where $\mathrm{ctx_end}_t = \mathrm{resp_start}_t$ is the position right after the last environment observation visible to turn t. We write $s_t = |c_t| + |r_t|$ for the number of new tokens turn t contributes (critique + response). Per-token log-probabilities of r_t under this conditioned input are recovered from the model's output logits at the response slice of the packed forward.

Algorithm 1 KVToPages: layer-wise KV → paged tensor.

Require: $K, V \in \mathbb{R}^{1\times h_{\mathrm{kv}}\times L_\tau\times d_{\mathrm{head}}}$, page size P
$\Delta \leftarrow (P - L_\tau \bmod P) \bmod P$; $L' \leftarrow L_\tau + \Delta$
Right-pad K, V with zeros along axis 2 to length L'
$n_\tau \leftarrow L'/P$
$K' \leftarrow K.\mathtt{view}(1, h_{\mathrm{kv}}, n_\tau, P, d_{\mathrm{head}})$
$K_{\mathrm{pages}} \leftarrow \mathtt{permute}(K', (2, 3, 1, 4))$
Apply the same view and permutation to obtain V_{pages}
return $(K_{\mathrm{pages}}, V_{\mathrm{pages}}, L_\tau)$

D.2 Page table layout

CAPA pages each layer's trajectory KV cache into fixed P=256-token pages (the alignment FA2's paged kernel expects). Let $n_\tau = \lceil L_\tau/P \rceil$ be the number of trajectory pages. Algorithm 1 (KVToPages) reshapes one layer's cache from $(1, h_{\mathrm{kv}}, L_\tau, d_{\mathrm{head}})$ into $(n_\tau, P, h_{\mathrm{kv}}, d_{\mathrm{head}})$.

For each annotated turn t the reads partition into *shared prefix pages* (referred to as "clean pages" below) and *stub pages*:

- **Clean pages.** The first $n_t^c = \lfloor \mathrm{ctx_end}_t/P \rfloor$ trajectory pages are fully occupied by tokens that turn t is allowed to attend to (every token in them lies strictly before $\mathrm{ctx_end}_t$). They are pointed to directly via the page table, with no copying.
- **Stub pages.** The trajectory tail $\tau_t = \mathrm{ctx_end}_t \bmod P$ tokens cannot be read from the clean cache: the same physical page also contains tokens belonging to *later* turns, and FA2's causal mask is computed at query position granularity, not page granularity, so reading the partial page directly would leak future-turn keys. Stub pages give turn t a private copy of the tail history followed by its own $K_{\mathrm{new}}/V_{\mathrm{new}}$.

Turn t allocates $n_t^s = \lceil (\tau_t + s_t)/P \rceil$ stub pages. Algorithm 2 (BuildPageTables) builds the per-turn block table and ragged Q/K length vectors. The block table for turn t lists, in order, the n_t^c clean page indices followed by n_t^s stub page indices. Stub pages live in a single tensor of length $n_s = \sum_{t=1}^{N} n_t^s$ appended after the trajectory pages, so a turn's absolute page index is $n_\tau + s_t^{\mathrm{start}} + p$ for the p-th stub page of turn t.

D.3 Packed forward and FlashAttention-2 dispatch

CAPA reuses the per-layer KV that the PPO log-probability pass has already produced. Concretely, the standard log-prob pass calls $\pi_\theta(x, \mathtt{use_cache=True})$ once on the trajectory;

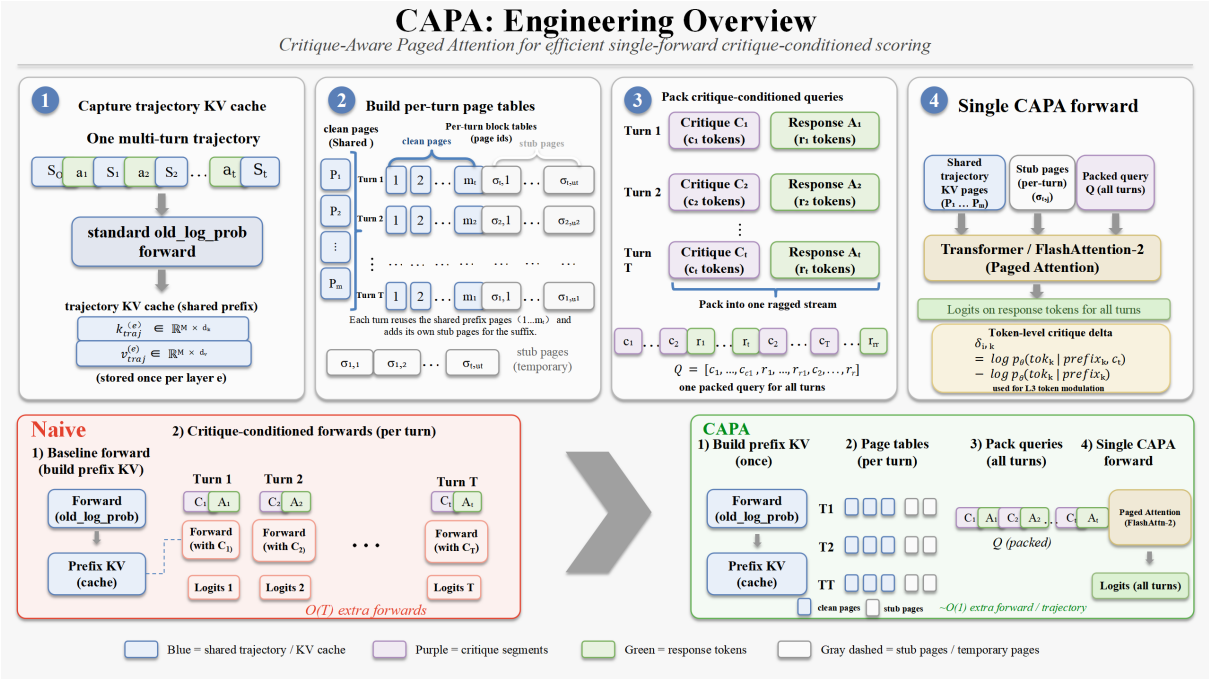

Figure 8: CAPA whole architecture. The trajectory KV cache is partitioned into shared prefix pages. Each annotated turn appends a private stub containing its rationale and response tokens. Per-turn `block_table` indices concatenate shared pages with the stub page, allowing all critique-conditioned continuations to be evaluated in one paged-attention call.

Algorithm 2 BUILDPAGETABLES: per-turn FA2 metadata.

Require: turn layouts $\{(\text{ctx_end}_t, s_t, |c_t|, |r_t|)\}_{t=1}^{N}, n_\tau, P$
stub_cursor $\leftarrow 0$
for $t = 1$ **to** N **do**
  $n_t^c \leftarrow \lfloor \text{ctx_end}_t / P \rfloor$; $\tau_t \leftarrow \text{ctx_end}_t \bmod P$
  $n_t^s \leftarrow \lceil (\tau_t + s_t)/P \rceil$
  Record $(n_t^c,\ \tau_t,\ n_t^s,\ s_t^{\text{start}}{=}\text{stub_cursor})$
  stub_cursor $\leftarrow$ stub_cursor $+ n_t^s$
  cache_seqlens$[t] \leftarrow \text{ctx_end}_t + s_t$
end for
$W \leftarrow \max_t(n_t^c + n_t^s)$
Allocate block_table $\in \mathbb{Z}^{N \times W}$ (zero-init)
for $t = 1$ **to** N **do**
  block_table$[t,\ 0 : n_t^c] \leftarrow [0, 1, \ldots, n_t^c - 1]$
  $a \leftarrow n_\tau + s_t^{\text{start}}$
  block_table$[t,\ n_t^c : n_t^c + n_t^s] \leftarrow [a,\ a+1,\ \ldots,\ a + n_t^s - 1]$
end for
cu_seqlens $\leftarrow [0,\ \text{cumsum}(s_t)_{t=1}^{N}]$
return (block_table, cache_seqlens, cu_seqlens, $\{(n_t^c, \tau_t, n_t^s, s_t^{\text{start}})\}$)

Algorithm 3 SCATTERINTOSTUBPAGES: in-place fill of private stubs.

Require: $K_{\text{stub}}, V_{\text{stub}} \in \mathbb{R}^{n_s \times P \times h_{\text{kv}} \times d_{\text{head}}}$; flat $K_{\text{traj}}, V_{\text{traj}} \in \mathbb{R}^{L_\tau \times h_{\text{kv}} \times d_{\text{head}}}$; new $K_{\text{new}}, V_{\text{new}} \in \mathbb{R}^{(\sum_t s_t) \times h_{\text{kv}} \times d_{\text{head}}}$; turn metadata; cu_seqlens
for $t = 1$ **to** N **do**
  $K_{\text{tail}} \leftarrow K_{\text{traj}}[\text{ctx_end}_t - \tau_t : \text{ctx_end}_t]$ (analog. V_{tail})
  $K_{\text{new},t} \leftarrow K_{\text{new}}[\text{cu_seqlens}[t] : \text{cu_seqlens}[t{+}1]]$ (analog. $V_{\text{new},t}$)
  $K_{\text{seg}} \leftarrow [\,K_{\text{tail}};\ K_{\text{new},t}\,]$ (length $\tau_t + s_t$)
  for $p = 0$ **to** $n_t^s - 1$ **do**
    $\ell_{\text{lo}} \leftarrow pP$; $\ell_{\text{hi}} \leftarrow \min((p{+}1)P,\ \tau_t + s_t)$
    $K_{\text{stub}}[s_t^{\text{start}} + p,\ 0 : \ell_{\text{hi}} - \ell_{\text{lo}}] \leftarrow K_{\text{seg}}[\ell_{\text{lo}} : \ell_{\text{hi}}]$
    (analog. for V_{stub}); trailing slots stay zero
  end for
end for

we detach and clone the returned $\{(K^\ell, V^\ell)\}_{\ell=1}^{L}$ on the local device (the `detach().clone()` is required to break FSDP's autograd graph and prevent re-sharding during the subsequent packed call). No additional full forward is paid by CAPA itself.

The conditioned forward then proceeds in one pass (Algorithm 4). Each layer's cache is paged via Algorithm 1; Algorithm 2 runs once and the page table is shared across layers. The packed query is the concatenation $\mathbf{q}_{\text{packed}} = [c_1\, r_1\, c_2\, r_2\, \cdots\, c_N\, r_N]$ and per-turn position IDs are $\text{pos}_t = (\text{ctx_end}_t,\ \ldots,\ \text{ctx_end}_t + s_t - 1)$, so RoPE applied during the packed forward matches the absolute trajectory frame the trajectory KV was captured in. We register a custom attention backend `ALL_ATTENTION_FUNCTIONS["capa"]` and set the model's `_attn_implementation` to `"capa"` for the duration of the packed forward, restoring the previous setting in a `try/finally` block.

Algorithm 4 CAPA-FORWARD: end-to-end critique-conditioned forward.

Require: Trained model π_θ, trajectory x of length L_τ, prompt length L_p, annotated turns $\{(\text{resp_start}_t, \text{resp_end}_t, \mathbf{c}_t)\}_{t=1}^{N}$, page size P

(Reuse) retrieve $\{(K^\ell, V^\ell)\}_{\ell=1}^{L}$ from the standard PPO log-prob pass; `detach().clone()`
Build per-turn layout $\{(\text{ctx_end}_t, s_t, |c_t|, |r_t|)\}_{t=1}^{N}$; build packed query $\mathbf{q}_{\text{packed}}$ and per-turn position IDs
(block_table, cache_seqlens, cu_seqlens, stubs) $\leftarrow$ BUILDPAGETABLES$(\cdot)$
for $\ell = 1, \ldots, L$ **do**
  $(K^\ell_{\text{pages}}, V^\ell_{\text{pages}}, \cdot) \leftarrow$ KVTOPAGES(K^ℓ, V^ℓ, P)
  Stash flat K^ℓ, V^ℓ for stub-page tail copies
end for
Set π_θ's attention implementation to `capa`; populate `capa_payload`
$\text{out}_{\text{cond}} \leftarrow \pi_\theta(\mathbf{q}_{\text{packed}}, \text{position_ids}, \texttt{capa_payload})$
Restore prior attention implementation
for $t = 1, \ldots, N$ **do**
  Locate response slice $[\text{resp_lo}_t, \text{resp_hi}_t)$ in out_{cond}`.logits`
  $\text{cond_logp}_t \leftarrow \text{logsoftmax}(\text{out}_{\text{cond}}\texttt{.logits}[\text{resp_lo}_t - 1 : \text{resp_hi}_t - 1])\,[\,\mathbf{q}_{\text{packed}}[\text{resp_lo}_t : \text{resp_hi}_t]\,]$
end for
return $\{\text{cond_logp}_t\}_{t=1}^{N}$

Inside the backend each layer receives $(Q, K_{\text{new}}, V_{\text{new}})$ of the new packed sequence plus that layer's $(K^\ell_{\text{traj}}, V^\ell_{\text{traj}})$ pages via the `capa_payload` dictionary. The backend allocates $(K^\ell_{\text{stub}}, V^\ell_{\text{stub}})$ and runs Algorithm 3. On CUDA fp16/bf16 it dispatches to `flash_attn_varlen_func` (Dao, 2024) with the page table:

```
flash_attn_varlen_func(
  q=q_packed, k=K_concat, v=V_concat,
  cu_seqlens_q=cu_seqlens,
cu_seqlens_k=cu_k,
  max_seqlen_q=max_q,
max_seqlen_k=max_k,
  causal=True, block_table=block_table)
```

where $K_{\text{concat}} = [\,K^\ell_{\text{traj,pages}};\ K^\ell_{\text{stub}}\,]$ along the page axis, and $\text{cu_k} = [0, \text{cumsum}(\text{cache_seqlens})]$. CPU/fp32 falls back to an eager reference (Algorithm 5) used in unit tests for indexing-equivalence checking. Because positional encoding is baked into K^ℓ_{traj} via the original log-prob pass and into K^ℓ_{new} via the packed forward's RoPE, no further position bookkeeping is required inside the kernel: FA2's causal mask suffices since each query token's absolute trajectory position is strictly greater than the keys it is allowed to see.

D.4 FSDP integration

The actor is wrapped in PyTorch `FullyShardedDataParallel` (FSDP) for sharded parameters and gradients. CAPA integrates with FSDP at three points.

Module unwrap. The custom backend needs to inspect the underlying causal-LM (`model.model.layers`, `model.lm_head`). FSDP and DeepSpeed both wrap the module; we unwrap with

```
root = getattr(model, "module",
    getattr(model,
"_fsdp_wrapped_module",
        model))
```

inside `CapaCritiqueForward._inner_model`, then access `root.model.layers` for per-layer KV capture.

KV capture under sharding. The PPO log-prob pass runs under FSDP's normal all-gather/forward path, so the returned `past_key_values` are full-precision, unsharded layer tensors on the local device. We `detach().clone()` each (K^ℓ, V^ℓ) to break the autograd graph and prevent FSDP from re-sharding the captured cache during the subsequent packed forward. Pages stay on the same device and dtype as the layer outputs; no cross-device transfer is required.

Packed forward. The packed forward triggers FSDP's all-gather identically to a normal forward; the only difference is that the input length is $\sum_t s_t$ (typically $\leq L_\tau$ in the regime where CAPA is profitable) and the attention backend is the registered `capa` function. We disable HF caching (`use_cache=False`) on the packed forward and pass `attention_mask={"full_attention": None, "sliding_attention": None}` so HF skips its default mask construction; CAPA builds its own mask via the page table plus FA2's `causal=True`. To keep gradient flow correct under FSDP we wrap the entire CAPA pass in

Algorithm 5 CAPA-EAGER: reference implementation (CPU/fp32 tests).

Require: Packed Q, paged K_{concat}, V_{concat}, block_table, cache_seqlens, cu_seqlens, turn layouts; softmax scale $\sigma = 1/\sqrt{d_{\text{head}}}$

for $t = 1$ **to** N **do**
  $T_t \leftarrow \text{cache_seqlens}[t]$; gather first T_t keys/values by walking $\text{block_table}[t]$ across pages; expand $h_{\text{kv}} \rightarrow n_{\text{heads}}$ for GQA
  $S \leftarrow \sigma\, Q[\text{cu_seqlens}[t] : \text{cu_seqlens}[t{+}1]] \cdot K^\top$
  Apply absolute-position causal mask: $S_{ij} \leftarrow -\infty$ if $j > \text{ctx_end}_t + i$
  $\text{out}[t] \leftarrow \text{softmax}(S) \cdot V$
end for
return packed out

`torch.no_grad()`: the log-probability shift $\delta_{b,t,j}$ enters Eq. (8) as a stop-gradient signal — it modulates the advantage but does not contribute its own gradient.

Mixed precision. On bf16/fp16 FSDP (the production setting), CAPA dispatches to FA2's varlen paged kernel with the layer's native dtype. The eager fallback is gated on `query.is_cuda & query.dtype in (fp16, bf16)`; unit tests run on CPU in fp32, exercising the same indexing logic through Algorithm 5.

D.5 Correctness, complexity, and fallbacks

Numerical correctness. We validate CAPA against the per-turn reference forward in two regimes. (i) On CPU/FP32 the maximum absolute difference across the response-token log-probabilities is below $\mathrm{atol} = 10^{-5}$, limited by softmax order of operations. (ii) On CUDA/BF16 with FA2's paged kernel the maximum observed drift is below $\mathrm{atol} = 4 \times 10^{-3}$, well below typical critique-induced shifts ($\mathrm{std}(\delta) \approx 0.1$ in our runs). An end-to-end smoke test runs a single PPO mini-batch with CAPA enabled and disabled and confirms the resulting $A^{(3)}$ tensors agree within the same BF16 tolerance.

Complexity. Let L_τ be the trajectory length and $S = \sum_t s_t$ the total new-token count across annotated turns ($S \leq N \cdot \max_t s_t$). The naive baseline runs N extra forwards each of length $\Theta(L_\tau + s_t)$, with attention cost $\Theta\big(\sum_t (L_\tau + s_t)^2\big)$. CAPA pays no additional trajectory forward (the trajectory KV is reused from the PPO log-prob pass) and adds only one packed forward whose attention cost is $\Theta\big(\sum_t s_t\,(\mathrm{ctx_end}_t + s_t)\big) \leq \Theta(S \cdot L_\tau)$. In our 7B / 32k-token regime this brings the per-step Level 3 overhead from 178 s to 14 s, a 12.7× speedup that holds the total TRACE-GRPO step within 5% of vanilla GRPO (Figure 7).

E Teacher Annotation Prompt

The box below shows the complete system prompt used to instruct the teacher LLM for trajectory annotation. The prompt is designed to elicit sparse, evidence-grounded labels: the teacher omits routine or neutral turns (no $d = 0$), commits to strong scores when evidence is clear, and grounds every judgment in observable environment feedback.

System Prompt for Teacher Annotation

You are an expert RL credit-assignment judge for long-horizon agent tasks. Output exactly one JSON object — no markdown, no code fences, no narration.

 # What you are doing

The trajectory has already finished. Your job is to identify turns where the agent's decision **materially changed the outcome** (positively or negatively), and assign a graded score to each. Use full context for credit assignment: task goal, prior state, executed action, environment feedback, trajectory context, and final outcome. Use the final outcome as a consistency reference, but not as a hard rule — a successful trajectory can still contain bad turns and a failed one can still contain strong decisions. The downstream RL trainer uses these scores as token-level advantage modulators — the strength and polarity of your scores directly shape gradient updates.

 # Judgment criteria

- Ground every judgment in observable evidence: environment feedback, score changes, error messages, task state transitions, or logical inference from the trajectory context, etc.
- **Positive turns**: advanced the goal, completed a subgoal, set up necessary preconditions, recovered from an earlier mistake, gathered critical information, etc.
- **Negative turns**: hindered progress, triggered errors, repeated failed actions, moved away from the goal, wasted steps on irrelevant actions, violated environment protocol, etc.
- You may infer poor strategy even without an explicit env error — e.g. repeating an unsuccessful approach, ignoring relevant observations, choosing an obviously suboptimal path, etc.
- Skip routine/filler turns that had no meaningful impact. Only label turns that genuinely shifted the trajectory's outcome.

 # Scoring guidance

- Commit to strong scores when the evidence is clear. ± 0.7 or ± 1.0 produce meaningful gradient signal; weak scores (± 0.3) are near-noise.
- Use ± 0.3 only when the step's impact is genuinely ambiguous.
- For confidence: `1.0` when env feedback directly confirms the result; `0.75` when the judgment relies on inference; `0.5` when truly uncertain.
- There is no required balance between positive and negative labels. Label what the evidence supports — follow the evidence, not a quota.

 # Output schema (every field required for emitted turns)

- `trajectory_outcome_recap`: object with `agent_succeeded`.
- `turn_judgments`: list of labeled turns.
- Each labeled turn contains `turn_index`, `decision_quality`, `confidence`, `judgment_tags`, and `rationale`.
- `decision_quality`: one of `-1.0`, `-0.7`, `-0.3`, `0.3`, `0.7`, `1.0`.
- `confidence`: one of `0.5`, `0.75`, `1.0`.

 # Rules

- Do NOT emit `decision_quality = 0.0`; if a turn is neutral, omit it.
- Do NOT emit `confidence < 0.5`; below that threshold, omit the turn.
- Keep `rationale` concise — cite specific env feedback or state changes.
- One tag per turn. Allowed tags: `goal_progress`, `accuracy`, `action_choice`, `observation_use`,

planning, exploration, recovery, efficiency, tool_or_env_protocol, formatting, safety_or_constraint, other.

- agent_succeeded must reflect the actual final outcome (lying voids the entire labeling).

 # Final reward formula (FYI)

q = decision_quality * confidence per labeled turn, applied as a token-level advantage modulator. Strong verdicts ($|q| \geq 0.7$) move the policy; weak verdicts ($|q| \leq 0.225$) are near-noise.

F SFT Baseline Construction

To test whether supervised fine-tuning on teacher-generated successes can match RL-based training, we construct SFT baselines using teacher rollouts. The teacher model, DeepSeek-V3.1, generates 3 trajectories per training task with temperature 0.4 under the standard environment prompts. The maximum episode length is 50 steps for ALFWorld and 30 steps for ScienceWorld, with a maximum of 512 tokens per turn. Each response follows the ⟨think⟩ . . . ⟨/think⟩⟨action⟩ . . . ⟨/action⟩ format.

Only successful trajectories (won=true) are retained and converted to multi-turn message format. Illegal actions are resolved by normalized matching with a task-specific fallback action: look for ALFWorld and look around for ScienceWorld. We fine-tune Qwen2.5-7B-Instruct on this success-only dataset to obtain **Student-SFT**. We additionally evaluate DeepSeek-V3.1 directly under prompting, denoted **Teacher-Prompt**. Both baselines are evaluated on the same held-out splits and under the same decoding settings as the RL methods.

G Teacher Annotation Case Study

This appendix grounds the abstract teacher-annotation pipeline of §3 in a concrete trajectory. We first state the exact JSON schema returned by the DeepSeek-V3.1 teacher, then walk through one ScienceWorld trajectory turn by turn, showing how the teacher's per-turn rationale and signed confidence q_t surface the *decision-level* information that trajectory-level reward cannot.

G.1 Annotation schema

For a trajectory of N turns the teacher returns a list of N records, one per turn:

```
{
  "turn_index": int,
  "response_start": int,
"response_end": int,
  "label": "positive" | "negative" |
"neutral",
  "q_t": float in [-1.0, 1.0],
  "has_format_error": bool,
  "critique_text":
    "<teacher_note>...</teacher_note>"
}
```

The fields play the following roles in the L1–L3 pipeline. response_start / response_end are flat-trajectory token offsets pointing at the assistant's response for this turn; they are used by the token-to-turn map $\iota_b(k)$ in §3.5. label is the teacher's discrete decision-quality verdict in $\{-1, 0, +1\}$; q_t multiplies that label by a confidence $\in [0, 1]$ to give the signed confidence used by L2's w_{crit} (Eq. 5) and L3's w_δ (Eq. 8). has_format_error mirrors the environment's format-error flag: when true we override $q_t \leftarrow 0$ and bypass L2/L3 modulation, so a lenient teacher cannot weaken the format penalty (§3.4). critique_text is the free-form rationale that becomes $r_{b,t}$ in Eq. (7); we wrap it with explicit <teacher_note>...</teacher_note> delimiters so the critique-conditioned forward (Appendix D.3) can splice it after the turn's prefix without ambiguity.

G.2 Worked example: "Find a non-living thing and move it to the blue box."

Table 3 displays a complete 7-turn trajectory from the ScienceWorld training pool (training step 43). The student makes the right high-level moves (turns 0–2), gets stuck for three consecutive turns on a parser ambiguity (turns 3–5), and recovers in turn 6. We chose this trajectory as illustrative because (i) the goal — "find a non-living thing, then move it to the blue box in the kitchen" — is short and unambiguous; (ii) the teacher emits both polarities and a reasoning chain that diagnoses the parser failure; and (iii) the trajectory-level reward signal (+92 raw progress, won=1) is identical to any other solved trajectory yet hides the mid-episode failure mode that L2/L3 need to penalise.

The annotation pattern in Table 3 is precisely the signal that L2/L3 amplify and that GRPO destroys by averaging. Turns 3–5 each receive $A_{\text{GRPO}} > 0$ (the trajectory ultimately wins), but the teacher's $q_t \in \{-0.7, -1.0, -0.7\}$ flips w_{crit} to < 1 and downweights the bad action tokens, while turn 6's $q_t = +1.0$ amplifies the recovery action. At the token level, splicing the turn-3 rationale — "Failed to specify inventory cup in move command, triggering ambiguity error" — shifts the policy's distribution over the response tokens that *would have* produced the disambiguating "move cup in inventory to

blue box" phrasing; those are exactly the tokens $\delta_{b,t,j}$ in Eq. (7) marks for L3 reweighting.

G.3 Why this layout works for L1–L3

Three structural properties of the schema are exploited by the algorithm. (i) *Sign agreement.* $\mathrm{sgn}(q_t)$ disagrees with $\mathrm{sgn}(A_{\mathrm{GRPO}})$ on turns 3–5, so L2's w_{crit} shrinks below 1 (Eq. 5); on turns 0–2 and 6 it agrees, so $w_{\mathrm{crit}} > 1$. The trajectory's total advantage is preserved up to a covariance term (Eq. 11), but the credit is reallocated toward the productive turns. (ii) *Confidence* $|q_t|$. The teacher's monotonic increase in disapproval across turns $3 \to 4$ ($-0.7 \to -1.0 \to -0.7$) reflects the deteriorating evidence; L3 carries this confidence through to the token-level scaling rather than collapsing it to a single bit. (iii) *Targeted rationale.* The free-form text on turn 3 names the precise failure ("failed to specify inventory cup") instead of restating the outcome. When this rationale is spliced into the critique-conditioned forward (Appendix D.3), the resulting per-token δ concentrates on the response tokens that produced the underspecified action — the tokens L3 must suppress to avoid the failure on the next rollout.

Table 3: One ScienceWorld trajectory annotated by the DeepSeek-V3.1 teacher (training step 43, task: *find a non-living thing and move it to the blue box in the kitchen*). Score is the environment's running progress in $[0, 100]$; Δ is the score change for that turn. The trajectory completes successfully (`won`$=1$, final score 100), but turns 3–5 stall on a parser ambiguity that trajectory-level reward cannot localise. Teacher rationales are reproduced verbatim with the `<teacher_note>` wrappers stripped.

Turn	Score	Δ	Action	q_t	Label	Teacher rationale
0	$8 \to 58$	$+50$	`focus on cup`	$+1.0$	positive	Correctly identified the non-living thing (cup) to focus on, confirmed by $+50$ score increase from environment.
1	$58 \to 67$	$+9$	`pick up cup`	$+1.0$	positive	Successfully picked up the cup, moving it to inventory as required for transportation, confirmed by $+9$ score increase.
2	$67 \to 83$	$+16$	`go kitchen`	$+1.0$	positive	Moved to the kitchen where the blue box is located, essential for task completion, confirmed by $+16$ score increase.
3	$83 \to 83$	0	`move cup to blue box`	-0.7	negative	Failed to specify inventory cup in move command, triggering ambiguity error and wasting a turn.
4	$83 \to 83$	0	`move cup to blue box`	-1.0	negative	Repeated same ambiguous command format after explicit environment feedback, resulting in unknown action error.
5	$83 \to 83$	0	`move cup to blue box in kitchen`	-0.7	negative	Added redundant location specification instead of addressing the inventory ambiguity, repeating the same error.
6	$83 \to 100$	$+17$	`0`	$+1.0$	positive	Correctly resolved the ambiguity by selecting option 0 (inventory cup), successfully completing the task with $+17$ score increase.