ATTNSR: ATTENTION STEP REWARD FOR LONG-HORIZON AGENTIC RL

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon LLM agents make multi-step decisions, yet terminal-reward-only reinforcement learning provides little direct supervision for individual steps. We introduce ATTNSR, a step-level reward augmentation method that derives reward allocations from policy attention extracted during training replay, without a separately trained large reward model or additional environment rollouts. ATTNSR traces backward through the policy model attention graph to estimate the relevance of earlier steps to the final response. The terminal step, measured relative to a running baseline, determines whether these rewards reinforce or penalize the selected steps. In a controlled Qwen2.5-7B-Instruct study, ATTNSR consistently outperforms baseline across all evaluated checkpoints on ALFWORLD, raising final held-out success from 91.2% to 97.1%. The step-reward ablation experiment shows that the benefit depends on how intermediate rewards are assigned, rather than simply on their presence. ATTNSR also improves SCIENCEWORLD at every checkpoint while cutting interaction by 29.9%. No statistically detectable performance improvement is observed on WebShop. These results support policy attention as a practical prior for step-level reward allocation, with benefits that may depend on task complexity and horizon.

1 INTRODUCTION

Reinforcement learning (RL) with verifiable rewards (Shao et al., 2024; DeepSeek-AI, 2025) is increasingly applied to agents that interleave reasoning and tool use (Putta et al., 2024; Wang et al., 2025b; Yao et al., 2023). In environments such as ALFWORLD, WEBSHOP, and SCIENCEWORLD, agents make sequences of decisions to complete a task (Shridhar et al., 2021; Yao et al., 2022; Wang et al., 2022). Vanilla PPO uses the outcome reward to estimate value targets and policy advantages (Schulman et al., 2017; 2016), but it provides limited information about intermediate action contributions. Assigning credit to earlier decisions therefore remains a central challenge over long horizon RL (Zhang, 2026).

Step-level rewards address this challenge by providing more localized training signals, but obtaining them can require step annotations, additional environment interactions, or a separately trained evaluator (Lightman et al., 2024; Setlur et al., 2024; Kazemnejad et al., 2025). This motivates using signals already available within the policy model to guide intermediate rewards, without incurring the substantial training and environment-interaction costs.

Attention patterns in the policy model may provide such a signal: they record how later decisions attend to earlier actions and observations. We hypothesize that these relationships provide a useful prior for allocating intermediate rewards, although attention alone does not establish causal contributions (Jain & Wallace, 2019; Abnar & Zuidema, 2020). The signal can be extracted during the policy model existing forward pass over collected trajectories. The method needs little computation, without requiring separate attribution backward pass or learned progress estimator.

Building on this hypothesis, we introduce Attention Step Reward (ATTNSR), which uses the policy model attention to identify relevant earlier steps and turns the task outcome into intermediate training signals. Starting from the final step, it traces attention links backward to estimate the relevance of earlier actions and observations. These estimates determine where helpful itermediate steps are for reward allocation, while weaker or less consistent attention reduces their overall magnitude. Outcomes above the average score in training yield positive intermediate rewards; those below it

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.

yield negative rewards. These signals can in principle be used directly by step-level RL algorithms or placed at assistant-step endpoints for token-level RL. We study the latter integration in PPO, where the added rewards change value targets and policy advantages while retaining the critic architecture and losses.

We evaluate ATTNSR on ALFWORLD, SCIENCEWORLD, and WEBSHOP using Qwen2.5-7B-Instruct LoRA with matched token-level PPO training. ALFWorld success rises from 91.2% to 97.1%, with larger gains on unseen tasks; a Linear Step Reward control reaches 92.3%. On ScienceWorld, ATTNSR yields higher official scores at all four reported checkpoints and 29.9% fewer mean environment steps. WebShop shows no statistically detectable performance improvement despite increased interactions. This contrast suggests the method has greater potential on complex, long-horizon tasks.

Our contributions can be summarized as follows:

- We hypothesize that policy attention provides a relevance prior over interaction steps by indicating how later decisions attend to earlier actions and observations. Based on this hypothesis, we use attention-derived dependencies to distribute outcome information across interaction steps for temporal credit assignment, treating attention as a heuristic relevance signal rather than evidence of causal contribution.
- We introduce ATTNSR, a step-level reward augmentation scheme that combines attention-based allocation, confidence gating, and outcome centering. The signals can be used directly as step-level rewards or aligned with assistant-step endpoints for token-level objectives; we study the latter integration with PPO. The scheme requires neither counterfactual rollouts nor a separately trained reward model.
- We evaluate ATTNSR across three agentic benchmarks with ablation and trajectory analysis to characterize its performance gains and potential boundary conditions across tasks with different interaction lengths. The results are consistent with a stronger benefit on complex, long-horizon tasks, while suggesting more limited gains when trajectories are short.

2 PROBLEM FORMULATION

We model each LLM-agent interaction trajectory as many turns.Turn i is a model conversation with query and K_i decision steps. Each step $S_{i,k}$ comprises an assistant message $A_{i,k}$ and its environment observation $O_{i,k}$ when present; the final message may have no observation. Let $\mathcal{I}_{i,k}$ be the assistant token indices at step k, and $e_{i,k} \in \mathcal{I}_{i,k}$ its last trainable token, the *step endpoint*. We supply PPO with a terminal task reward $R_i \in [0, 1]$. In the canonical runs, R_i is binary success for ALFWORLD, the normalized official score for WEBSHOP, and the normalized final score for SCIENCEWORLD; format-error trajectories receive $R_i = 0$. Thus the binary success metric for WEBSHOP and SCIENCEWORLD is an evaluation threshold on the official score, not a replacement for the training reward. Table 14 in Appendix G summarizes the notation.

Our terminal-only PPO baseline aligns rewards with trajectory tokens, indexed by u, placing the outcome at the final assistant endpoint:

$$r_{i,u}^{\mathrm{PPO}} = R_i \mathbb{1}[u = e_{i,K_i}], \tag{1}$$

where $\mathbb{1}[\cdot]$ is the indicator function. Observation tokens provide context but are masked from actor and critic losses. TD residuals and GAE compute value-function targets and policy advantages over valid assistant tokens, skipping masked positions (Schulman et al., 2016; 2017). The terminal reward evaluates the whole trajectory without specifying contributions of individual actions.

Let $s_{i,k}$ denote a signed shaping reward assigned to intermediate step $k < K_i$. ATTNSR augments the baseline reward sequence as

$$r_{i,u}^{\mathrm{AttnSR}} = r_{i,u}^{\mathrm{PPO}} + \sum_{k=1}^{K_i-1} s_{i,k} \mathbb{1}[u = e_{i,k}]. \tag{2}$$

This retains R_i at the final endpoint and adds $s_{i,k}$ at intermediate assistant endpoints. Algorithms that consume step-level rewards can use these signals directly in principle; Equation 2 specifies

their token-level placement for the PPO implementation studied here. Our goal is to derive $s_{i,k}$ from actor attention and terminal outcomes. Section 3 defines its allocation, confidence gate, and outcome-dependent sign.

3 ATTNSR: ACTOR ATTENTION FOR STEP CREDIT

ATTNSR constructs intermediate rewards by tracing how later decisions attend to earlier interaction steps. It uses these relationships to allocate rewards, with their magnitude and sign determined by the reliability of the attention signal and the trajectory outcome relative to a running baseline.

3.1 WHERE TO ASSIGN CREDIT: A STEP DEPENDENCY GRAPH

For each selected transformer layer l, we construct a directed graph over interaction steps. Its components $G^{(l)}_{i,\mathrm{AA}}[j,k]$ and $G^{(l)}_{i,\mathrm{AO}}[j,k]$ measure attention from the query tokens of assistant step j to the assistant-message and paired-observation key tokens of historical step k, respectively. We sum causal attention probabilities over each historical message key tokens, then average over step j query tokens and heads. Only $k < j$ edges are retained; missing observations contribute zero AO mass. The combined graph is $G^{(l)}_i = 0.5G^{(l)}_{i,\mathrm{AA}} + 0.5G^{(l)}_{i,\mathrm{AO}}$.

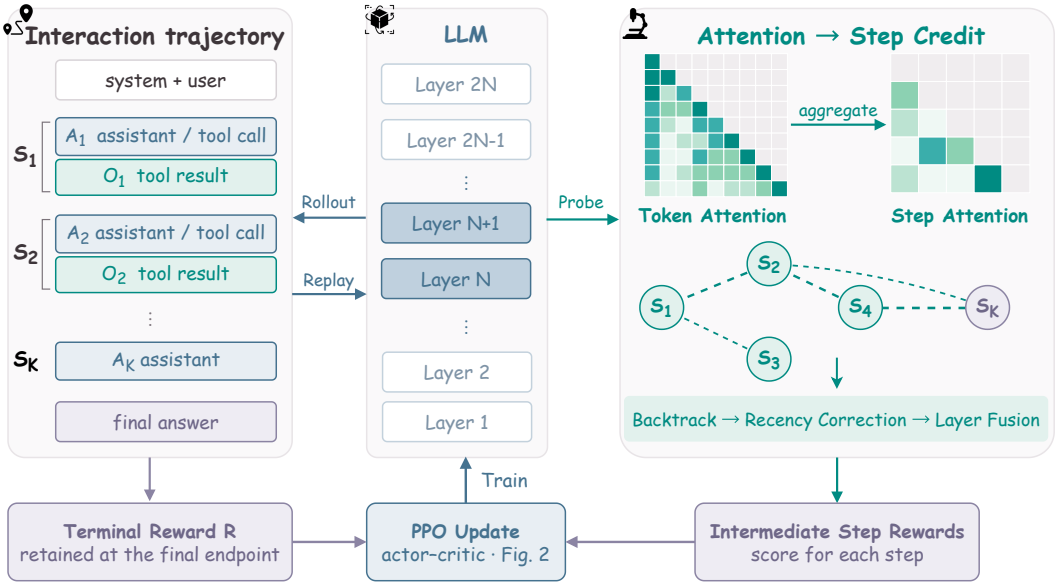

Figure 1: Overview of ATTNSR. Actor replay supplies attention for a step graph. Backtracking from the terminal step, recency correction, and layer fusion yield an allocation; confidence and the centered outcome scale it into signed intermediate rewards. The terminal reward is retained. Graph edges link later steps to earlier dependencies; episode indices are omitted.

During actor replay for old log-probabilities, we capture Q/K after rotary position embedding (Su et al., 2024) and before fused attention (Dao et al., 2022), reconstructing the packed tensor/context-parallel sequence. Query blocks use full causal-prefix normalization before message aggregation; the historical-step submatrix is not renormalized, preserving its mass relative to other context tokens. Figure 1 illustrates this conversion from token attention to a step graph and its subsequent use in reward construction.

Relevance then flows backward from the terminal step, independently within each layer. Writing $a^{(l)}_{i,k} \geq 0$ for the relevance layer l assigns to step k,

$$a^{(l)}_{i,K_i} = 1, \qquad a^{(l)}_{i,k} = \rho \sum_{j>k} a^{(l)}_{i,j} G^{(l)}_i[j,k], \tag{3}$$

where $j \leq K_i$. Each backward path contributes its edge-weight product with decay $\rho = 0.85$ per hop, capturing direct and indirect dependencies. This graph-hop decay differs from the temporal-distance correction below. Only $k < K_i$ scores enter reward allocation.

3.2 How much to trust attention: correction and consensus

To reduce recency preference, define terminal distance $\Delta_{i,k} = K_i - k$ and proximity $d_{i,k} = \exp(-(\Delta_{i,k} - 1)/\kappa)$. For $K_i > 1$, we standardize log relevance and proximity over intermediate steps:

$$z_{i,k}^{(l)} = \mathrm{norm}\Big(\log \max(a_{i,k}^{(l)}, \epsilon)\Big) + \eta\, \mathrm{norm}(d_{i,k}), \qquad p_{i,k}^{(l)} = \frac{\exp(z_{i,k}^{(l)}/T)}{\sum_{h<K_i} \exp(z_{i,h}^{(l)}/T)}. \tag{4}$$

Here $\epsilon = 10^{-12}$ and $\mathrm{norm}(x) = (x - \mathrm{mean}(x))/\,\mathrm{std}(x)$ uses population standard deviation, returning zero for a single element or standard deviation at most 10^{-6}. We use $T = 1$, $\kappa = 4$, and $\eta = -0.25$; the negative proximity term reduces allocation to recent steps without determining reward sign.

Because attribution varies across layers (Abnar & Zuidema, 2020), we fuse their independently computed allocations by a geometric mean:

$$q_{i,k} = \exp\left(\frac{1}{L}\sum_{l\in\mathcal{L}} \log \max\left(p_{i,k}^{(l)}, \epsilon\right)\right), \qquad p_{i,k} = \frac{q_{i,k}}{\sum_{h<K_i} q_{i,h}}, \tag{5}$$

where $L = |\mathcal{L}|$ is the number of selected layers. Geometric fusion favors consistent support across layers; normalization is over intermediate steps.

Confidence controls the strength of the attention-derived reward. For layer l, define the terminal covered mass as

$$m_i^{(l)} = \sum_{k<K_i} \Big(G_{i,\mathrm{AA}}^{(l)}[K_i, k] + G_{i,\mathrm{AO}}^{(l)}[K_i, k]\Big) \in [0, 1]. \tag{6}$$

This uses the summed AA and AO masses before the equal-weight graph mixture. For $K_i > 2$, define the normalized entropy

$$H_i^{(l)} = -\frac{\sum_{k<K_i} p_{i,k}^{(l)} \log \max(p_{i,k}^{(l)}, \epsilon)}{\log(K_i - 1)}. \tag{7}$$

For $K_i = 2$, the unique intermediate step has allocation one and we set $H_i^{(l)} = 0$. The confidence components are

$$C_{i,\mathrm{mass}}^{(l)} = \mathrm{clip}(m_i^{(l)}/0.20, 0, 1), \tag{8}$$

$$C_{i,\mathrm{focus}}^{(l)} = \mathrm{clip}((1 - H_i^{(l)})/0.15, 0, 1), \tag{9}$$

$$C_{i,\mathrm{layer}} = 1 - \mathrm{mean}_l\, \mathrm{TV}(p_i^{(l)}, p_i), \tag{10}$$

$$C_i = \mathrm{mean}_l\Big(C_{i,\mathrm{mass}}^{(l)} C_{i,\mathrm{focus}}^{(l)}\Big)\, C_{i,\mathrm{layer}} \in [0, 1], \tag{11}$$

where $\mathrm{clip}(x, 0, 1)$ saturates to $[0, 1]$ and $\mathrm{TV}(p, q) = \frac{1}{2}\sum_k |p_k - q_k|$. Covered mass, concentration, and layer agreement measure historical attention, selectivity, and consistency, respectively; 0.20 and 0.15 are saturation thresholds. The resulting confidence is a shaping gate, not a calibrated probability of correct attribution.

Zero total intermediate relevance sets that layer focus confidence to zero. For $K_i = 1$, we skip allocation, set $C_i = 0$, and retain terminal-only rewards.

3.3 Whether to reinforce a step: a centered budget

Attention indicates computational relevance, not whether a step contributed positively to success. We determine reward sign relative to a running outcome baseline. For collected training batch t with mean terminal reward M_t, we initialize or update the baseline before shaping:

$$\bar{R}_t = \begin{cases} M_1, & t = 1, \\ 0.99\bar{R}_{t-1} + 0.01 M_t, & t > 1, \end{cases} \qquad B_i = \beta\, C_i(R_i - \bar{R}_t), \tag{12}$$

Algorithm 1 ATTNSR reward construction for one batch

Require: completed trajectory with endpoints $e_{i,k}$ and rewards R_i; selected layers $\mathcal{L}$; previous baseline if initialized
Ensure: token rewards $r_{i,u}$; updated baseline $\bar{R}_t$
capture detached post-RoPE Q, K during actor replay for old log-probabilities
initialize $r_{i,u} \leftarrow R_i \mathbb{1}[u = e_{i,K_i}]$ and $C_i \leftarrow 0$ for each turn
compute batch mean M_t; initialize or update $\bar{R}_t$ by Equation 12
for each episode i with $K_i > 1$ **do**
    for each layer $l \in \mathcal{L}$ **do**
        aggregate causal attention into AA/AO step graphs; set $G_i^{(l)} \leftarrow \frac{1}{2}(G_{i,\mathrm{AA}}^{(l)} + G_{i,\mathrm{AO}}^{(l)})$
        backtrack relevance and apply recency correction to obtain $p_i^{(l)}$ ▷ Eqs. 3–4
        compute mass and focus confidence; zero focus if intermediate relevance is zero
    fuse layer allocations into p_i and compute C_i ▷ Eqs. 5–11
    add $\beta C_i(R_i - \bar{R}_t)p_{i,k}$ at $u = e_{i,k}$ for $k < K_i$ ▷ retain terminal reward
return $r_{i,u}$, $\bar{R}_t$

where $\beta = 0.25$. The detached baseline is shared across the batch: outcomes above it yield positive budgets, and those below it yield negative budgets. Thus sign reflects relative outcome quality, not binary success. For $k < K_i$, we assign

$$s_{i,k} = B_i\, p_{i,k} = \beta\, C_i(R_i - \bar{R}_t)\, p_{i,k}, \tag{13}$$

at endpoint $e_{i,k}$, retaining R_i at e_{i,K_i}. For $K_i > 1$, intermediate rewards sum to B_i; zero confidence yields no shaping. Format-error trajectories retain environment reward $R_i = 0$ and participate in the EMA update. Algorithm 1 summarizes the complete reward construction for a collected batch.

Since $C_i, R_i, \bar{R}_t \in [0, 1]$, $\sum_{k<K_i} |s_{i,k}| = |B_i| \leq \beta$: the upper bound is length-independent, while the realized budget depends on confidence and outcome. The undiscounted shaped return is $R_i + B_i$, so policy invariance is not guaranteed. This is a training heuristic distinct from return-preserving redistribution (Arjona-Medina et al., 2019); evaluation uses the original task metrics.

3.4 COMPLEXITY AND INTEGRATION

Blockwise aggregation into step buckets avoids storing full token attention matrices but retains quadratic query–key arithmetic and selected-layer buffers. It reuses the actor forward pass without additional rollouts or a separately trained reward model. Wall-clock and peak-memory overheads remain unmeasured. Rewards enter PPO before advantage estimation, retaining its critic architecture and losses.

Actor attention is detached when constructing rewards; policy probabilities remain differentiable during optimization. Given pre-update critic values V_u^{old} and unnormalized GAE advantages $\hat{A}_u$, the canonical PPO training signals are

$$\widetilde{A}_u = \mathrm{sg}\Big(\mathrm{Norm}(\hat{A})_u\Big), \qquad y_u = \mathrm{sg}\Big(\hat{A}_u + V_u^{\mathrm{old}}\Big), \tag{14}$$

where Norm uses the response mask and sg denotes stop-gradient. The actor uses $\widetilde{A}_u$ in its clipped surrogate; the critic fits returns y_u, constructed *before* advantage normalization, using the existing clipped value loss with reference V^{old}. Both signals remain fixed within the batch; neither loss backpropagates through GAE or reward construction.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Agentic Tasks. We evaluate on ALFWORLD (274 held-out tasks: 140 seen and 134 unseen), WEBSHOP (500 test tasks), and SCIENCEWORLD (1,819 test tasks) (Shridhar et al., 2021; Yao et al., 2022; Wang et al., 2022), covering household interaction, online shopping, and scientific procedures.

ALFWorld official score equals task success rate. WebShop and ScienceWorld additionally provide graded scores for partial progress, reported separately from binary success. Their evaluators mark success at normalized scores of at least 0.99 and 0.999, respectively.

RL Training. We use Qwen2.5-7B-Instruct (Qwen Team, 2024) with a frozen base model and rank-64 LoRA adapters (Hu et al., 2022) for actor and critic. Both learning rates are 5×10^{-6}, the global batch size is 8, and the reference KL coefficient is 0.02. PPO uses one rollout per prompt and one optimization epoch per batch. Separate policies train on each benchmark training split, excluding held-out tasks; Appendix Table 7 reports data sizes and update counts. Figure 2 shows how detached actor attention changes policy advantages and value targets through intermediate rewards, retaining the PPO losses. We evaluate after each of four data epochs. Training uses VERL (Sheng et al., 2025), Megatron (Shoeybi et al., 2019), and vLLM (Kwon et al., 2023) on 8×H200 GPUs (140 GB each), with a 32,768-token context and no additional per-turn token or turn-count cap. Attention extraction uses $L = 6$ layers and 128-token query blocks; Appendix C gives implementation settings and hyperparameters.

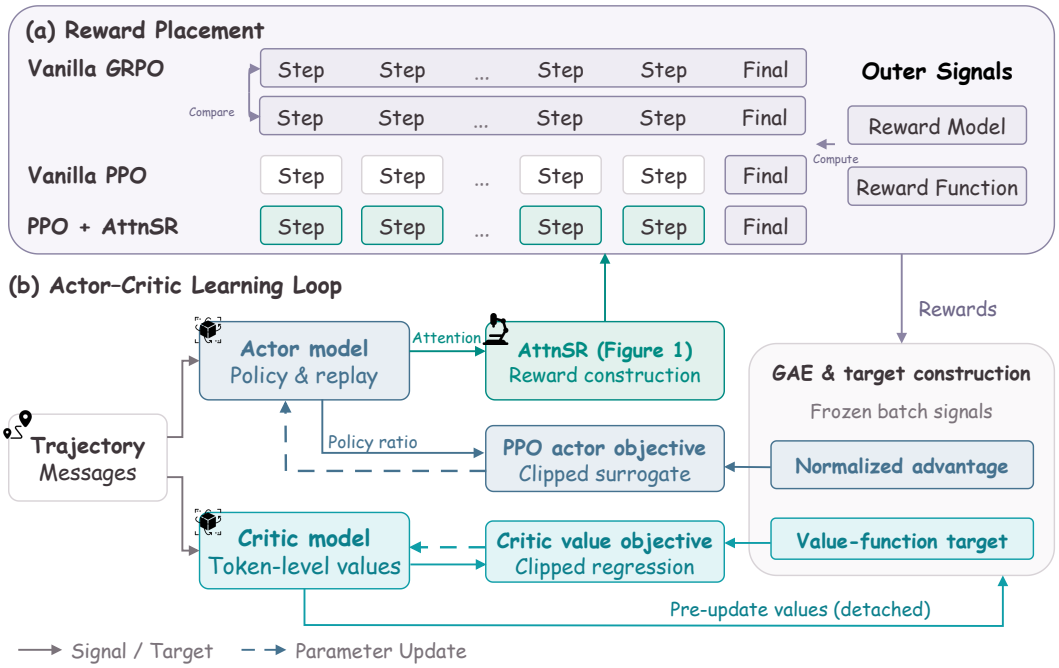

Figure 2: AttnSR integrated in Actor–critic RL Algorithm. (a) Unlike the terminal-only PPO and GRPO baselines, ATTNSR adds intermediate rewards while retaining the terminal reward. (b) Detached actor attention supplies rewards; pre-update critic values support GAE and target construction. Normalized advantages guide the actor, while estimated returns supervise the critic (Equation 14). Targets remain fixed during optimization. Solid arrows carry signals or targets; dashed arrows denote parameter updates.

Baselines and Controls. Terminal-only PPO matches ATTNSR in model, rollouts, optimizer, and evaluation. Linear Step Reward controls for positional shaping on ALFWorld (Section 4.3). GRPO uses four rollouts per prompt for one data epoch, serving as an exploratory reference rather than a controlled comparison.

Evaluation Protocol. We decode greedily once per held-out task and report full-set metrics; WebShop valid is reserved for training-time validation. The main comparison uses epoch 4 without selecting the best test-set checkpoint. Format errors violate the required tool-call protocol and receive terminal reward zero without an additional training penalty. For statistical comparisons, exact-input matching retains all 274 ALFWorld and 500 WebShop tasks and 1,488 of 1,819 ScienceWorld tasks (Appendix E). A two-sided exact McNemar test (McNemar, 1947) assesses binary success

differences; a paired bootstrap (Efron, 1979) with 10,000 resamples estimates 95% percentile intervals for success-rate and official-score differences. These intervals quantify sampling uncertainty over the matched held-out tasks, conditional on the trained policies.

Table 1: Method comparison across the three benchmarks. All values are shown as percentages(%).

	ALFWorld			WebShop		ScienceWorld	
Method	Score	Seen	Unseen	Score	Success	Score	Success
GRPO†	69.71	75.00	64.18	67.21	37.80	30.19	16.66
PPO	91.24	92.86	89.55	**70.74**	40.20	44.53	28.59
PPO + AttnSR	**97.08**	**96.43**	**97.76**	69.71	**41.00**	**46.35**	**30.90**

†GRPO is evaluated at epoch 1 with rollout $n = 4$; PPO and AttnSR are evaluated at epoch 4 with rollout $n = 1$. GRPO is therefore not a controlled comparison. ALFWorld's official score equals success rate; WebShop and ScienceWorld report official score and binary success separately.

4.2 Main Results

At epoch 4, AttnSR shows the clearest benefit on ALFWorld, smaller numerical gains on ScienceWorld, and mixed outcomes on WebShop (Table 1). ALFWorld success rises from 91.24% to 97.08% (266/274), reducing failures from 24 to 8. ScienceWorld official score increases from 44.53% to 46.35%, and success from 28.59% to 30.90%. WebShop success rises from 40.20% to 41.00%, but official score falls from 70.74% to 69.71%.

Statistical Evidence. Tests use exact-input-matched tasks rather than the full-set estimates in Table 1. On ALFWorld, 20 tasks are solved only by AttnSR and 4 only by PPO ($p = 0.002$; 95% gain interval $[+2.55, +9.49]$ points). Only ALFWorld remains significant after Bonferroni correction across the three success tests. ScienceWorld gains $+2.49$ points on 1,488 matched tasks, versus $+2.31$ on the full set; its unadjusted $p = 0.040$ does not survive correction. Its score interval and both WebShop intervals include zero (Appendix D).

4.3 Ablation, Control, and Transfer Analysis

Positional Reward Control. Linear Step Reward tests whether a simple temporal allocation can reproduce the ALFWorld gain. It retains the terminal reward, outcome centering, and budget scale 0.25, but replaces attention allocation with increasing temporal weights and removes confidence gating. For $K_i > 1$,

$$w_{i,k}^{\text{lin}} = \frac{k}{\sum_{j=1}^{K_i-1} j}, \qquad s_{i,k}^{\text{lin}} = 0.25(R_i - \bar{R}_t^{\text{lin}})w_{i,k}^{\text{lin}}. \tag{15}$$

Each run maintains its own outcome baseline. The control reaches 92.34% success, gaining 1.10 points over PPO versus AttnSR 5.84 points (Table 2), recovering less than one fifth of the gain.

Linear Step Reward leads at epoch 1 but trails both methods at epochs 2–3 and AttnSR at epoch 4 (Appendix Table 5). Joint changes to allocation, gating, and realized budgets prevent isolating attention contribution or ruling out other dense rewards. Differences are descriptive because per-task outputs were not retained for paired testing.

Supplementary Training Configuration. Under single-trajectory asynchronous Qwen2.5-7B-Instruct training, epoch-4 ALFWorld success rises from 59.32% to 63.99% ($+4.67$ points). Changed collection, importance weighting, and critic updates limit comparison to this setting (Appendix F), precluding component attribution or direct comparison with canonical PPO.

Cross-backbone Check. With Qwen3-4B-Instruct-2507 and the main PPO/shaping hyperparameters, epoch-4 success rises from 87.96% to 91.24% ($+3.28$ points; nine additional successes). This extends the descriptive evidence to another backbone without isolating reward components or model-size effects (Appendix F).

Table 2: ALFWorld score. Blocks show the main control and two supplementary comparisons; Δ is the gain over each block baseline in percentage points.

Method	Succ. (%)	Δ (pp)
PPO	91.24	–
+Linear Step Reward	92.34	+1.10
+AttnSR	**97.08**	**+5.84**
PPO *Asynchronous*	59.32	–
+AttnSR	**63.99**	**+4.67**
Qwen3-4B-Instruct-2507		
PPO	87.96	–
+AttnSR	**91.24**	**+3.28**

Table 3: Trajectories on full held-out sets. Steps are environment interactions; output includes tool observations and is measured in thousands of characters.

		Environment steps			Out.
Task	Method	Mean	Med.	P90	(k)
ALFWorld	PPO	19.28	9.0	38.7	16.91
	AttnSR	15.51	8.0	31.0	13.32
WebShop	PPO	5.93	5.0	9.0	8.31
	AttnSR	8.03	7.0	12.0	11.08
ScienceWorld	PPO	30.68	15.0	80.0	45.42
	AttnSR	21.51	10.0	74.0	31.99

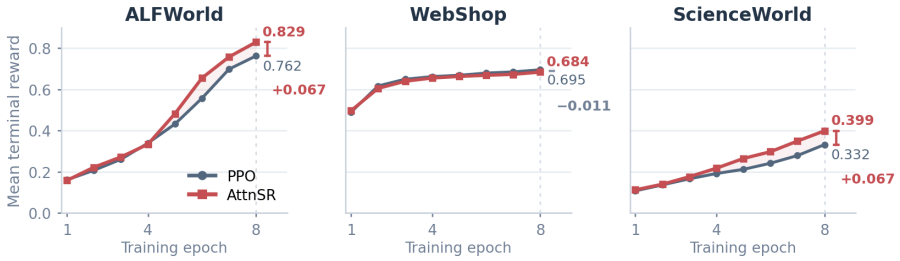

Figure 3: Mean terminal environment reward during training, excluding intermediate shaping rewards. These eight-epoch traces are supplementary to the four-epoch held-out comparison.

4.4 Learning Dynamics and Task-Level Analysis

Training and Checkpoints. AttnSR scores higher at all checkpoints on ALFWorld and ScienceWorld, with gaps peaking at epochs 4 and 3, respectively (Appendix A). Positive differences on WebShop at epochs 1–3 reverse slightly at epoch 4. The eight-epoch training traces with smaller learning rate 1e-6 (Figure 3) supplement these four-epoch held-out results; neither establishes an asymptotic advantage.

Generalization and Task Categories. ALFWorld success rises from 92.86% to 96.43% on seen tasks and from 89.55% to 97.76% on unseen tasks, with the larger unseen gain consistent with improved generalization. *Clean*, *Heat*, *Cool*, and *Pick2* gain 6.89–10.87 points, while *Pick* and *Look* remain unchanged (Appendix Table 6). The improved families require object processing or handling two objects, consistent with a benefit on dependent action sequences. However, each gains only three to five successes, and PPO is already at 100%/96.77% on Pick/Look; these differences do not isolate complexity from available headroom.

4.5 Inference-time Interaction Analysis

We analyze 5,186 epoch-4 trajectories (Table 3). Steps count environment interactions, not internal reasoning; output length includes observations and does not measure generated tokens or compute (Appendix E).

Overall Interaction Cost. Mean steps fall from 19.28 to 15.51 on ALFWorld (-19.56%) and from 30.68 to 21.51 on ScienceWorld (-29.88%), alongside higher success. Serialized output decreases by 21.22% and 29.56%, respectively. WebShop instead increases from 5.93 to 8.03 steps without a clear performance gain.

What Explains Shorter Trajectories? On 268 ScienceWorld tasks solved by both policies, mean steps fall from 11.06 to 7.40. ALFWorld 246 joint successes instead increase from 11.19 to 11.99 steps, and WebShop 157 from 5.11 to 6.19 (Appendix Table 12). ALFWorld aggregate saving

accompanies fewer prolonged failures: 24 PPO failures average 103.83 steps, versus 8 AttnSR failures averaging 109.12. Earlier unsuccessful stopping may also reduce aggregate length (Appendix E). These outcome-selected comparisons distinguish shorter solutions from fewer failures but do not establish a causal credit-assignment mechanism.

Scope of the Benefit. WebShop has shorter PPO trajectories (5.93 mean steps versus 19.28 on ALFWorld and 30.68 on ScienceWorld) and relatively close PPO/GRPO scores (70.74%/67.21%). Its lack of a detectable AttnSR gain suggests less benefit when shorter trajectories reduce delayed-credit difficulties. However, GRPO uses four rollouts per prompt for one epoch, versus one rollout for four epochs in PPO; the scores establish neither equivalence nor convergence. Policy-dependent lengths also confound horizon and complexity. Greater benefit on complex, long-horizon tasks thus remains an AttnSR hypothesis, not evidence that step rewards generally fail on short tasks.

5 Related Work

Agentic RL. Language agents interleave reasoning with actions (Yao et al., 2023; Putta et al., 2024), each spanning multiple tokens. This creates a distinction between token-level optimization and step-level credit assignment. POAD decomposes Bellman backups within and across actions (Wen et al., 2024); ArCHer pairs a turn-level critic with a token-level actor (Zhou et al., 2024); GiGPO groups steps with matching anchor states to estimate critic-free relative advantages (Feng et al., 2025). These methods address the granularity of value or advantage estimation. AttnSR intervenes at the reward interface: it assigns outcome-conditioned rewards to interaction steps, which then enter the target computation of RL algorithms. In our token-level PPO implementation, these rewards are placed at assistant-step endpoints and propagated through GAE (Schulman et al., 2017; 2016), retaining the critic architecture and optimization losses. This interface permits step-level or token-level integration in principle, but our experiments establish its effectiveness only within PPO.

Credit assignment. Long horizons and partial observability complicate credit assignment from terminal outcomes (Zhang, 2026). VinePPO estimates intermediate values through Monte Carlo rollouts (Kazemnejad et al., 2025); AgentPRM learns step-level promise and progress (Xi et al., 2025); SWEET-RL trains an evaluator with privileged information (Zhou et al., 2025). Process supervision also uses step annotations (Lightman et al., 2024), estimated progress (Setlur et al., 2024), or trajectory preferences (Liu et al., 2025). Other approaches use counterfactual baselines (Foerster et al., 2018), hindsight estimators (Tan et al., 2026), learned progress in SPA-RL (Wang et al., 2025a), or gradient saliency in ACPO (Yin et al., 2025). AttnSR draws its allocation prior from actor attention, avoiding additional environment rollouts, a separately trained large reward model, or an attribution backward pass. This signal expresses relevance rather than verified causal importance (Jain & Wallace, 2019; Abnar & Zuidema, 2020); confidence gating moderates its strength, while the centered outcome determines its sign. Unlike return-preserving redistribution in RUDDER (Arjona-Medina et al., 2019), AttnSR retains the terminal reward and adds intermediate rewards. It therefore changes the training return without guaranteeing policy invariance, and its utility is assessed using the original task metrics.

6 Conclusion

AttnSR turns relationships between interaction steps, extracted from actor attention, into outcome-conditioned intermediate rewards. Confidence gating and outcome centering regulate their strength and sign without a separately trained large reward model or additional rollouts. The resulting step-level augmentation changes value targets and policy advantages while retaining the standard PPO losses, offering a practical way to reuse actor signals for training supervision.

Integrated with token-level PPO, AttnSR raises ALFWorld success from 91.2% to 97.1%, exceeding the Linear Step Reward control, and yields higher ScienceWorld scores with fewer interactions. WebShop shows no statistically detectable benefit despite increased interactions. This contrast suggests greater potential for complex, long-horizon tasks, although it does not isolate horizon or complexity as the cause. The results support attention as a useful allocation prior in the tested settings; controlled horizon comparisons and integration with other RL algorithms remain necessary to establish its broader scope.

Ethics Statement

This work studies training methods for simulated text-based agents. The experiments use benchmark environments and do not involve human subjects or collection of personal data. As with other agent-training methods, improved planning and tool-use capability could be transferred to applications with unintended consequences; deployment should therefore include task-specific safety controls and monitoring.

Data Availability Statement

We use publicly available ALFWorld, WebShop, and ScienceWorld benchmark environments and do not introduce a new dataset. The evaluation splits, checkpoint-level summaries, and derived tables used in this paper are included with the supplementary artifacts.

Funding and Conflict of Interest

This work received no external funding. The authors declare no conflict of interest.

Reproducibility Statement

The training configuration, reward equations, evaluation splits, and checkpoint-level results are specified in this paper and its appendix. An anonymized repository will provide the reward extraction and allocation implementation, launch configurations, evaluation summaries, experiment record, significance analysis, and cached per-episode inputs. The main experiments use Qwen2.5-7B-Instruct, LoRA rank 64, VERL/Megatron training, vLLM rollout, and the fixed held-out sets described in Section 4.1.

AI Use Statement

Generative AI tools were used to assist with the organization and language editing of this manuscript, as well as with the development, debugging, and refinement of experimental code. The authors reviewed and verified all AI-assisted content, including technical claims, equations, experimental results, implementation details, and code, against the repository artifacts and experimental records. The authors remain fully responsible for the correctness, integrity, and reproducibility of the submitted work.

References

Samira Abnar and Willem H. Zuidema. Quantifying attention flow in transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4190–4197, 2020. doi: 10.18653/v1/2020.acl-main.385.

Jose A. Arjona-Medina, Michael Gillhofer, Michael Widrich, Thomas Unterthiner, Johannes Brandstetter, and Sepp Hochreiter. Rudder: Return decomposition for delayed rewards. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Bradley Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1): 1–26, 1979.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. *Advances in Neural Information Processing Systems*, 2025.

Jakob N. Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, volume 32, 2018. doi: 10.1609/aaai.v32i1.11794.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Sarthak Jain and Byron C. Wallace. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3543–3556, 2019.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. Vineppo: Refining credit assignment in rl training of llms. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, 2023.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024.

Xiaoqian Liu, Ke Wang, Yuchuan Wu, Fei Huang, Yongbin Li, Junge Zhang, and Jianbin Jiao. Online process reward learning for agentic reinforcement learning. *arXiv preprint arXiv:2509.19199*, 2025.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

Pranav Putta, Edmund Mills, Naman Garg, Sumeet Motwani, Chelsea Finn, Divyansh Garg, and Rafael Rafailov. Agent q: Advanced reasoning and learning for autonomous ai agents. *arXiv preprint arXiv:2408.07199*, 2024.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2016.

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

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding progress: Scaling automated process verifiers for llm reasoning. *arXiv preprint arXiv:2410.08146*, 2024.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Li, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

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

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

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

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Hui-Ze Tan, Xiao-Wen Yang, Hao Chen, Jie-Jing Shao, Yi Wen, Yuteng Shen, Weihong Luo, Xiku Du, Lan-Zhe Guo, and Yu-Feng Li. Hindsight credit assignment for long-horizon llm agents. *arXiv preprint arXiv:2603.08754*, 2026.

Hanlin Wang, Chak Tou Leong, Jiashuo Wang, Jian Wang, and Wenjie Li. Spa-rl: Reinforcing llm agents via stepwise progress attribution. *arXiv preprint arXiv:2505.20732*, 2025a.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. Scienceworld: Is your agent smarter than a 5th grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11279–11298, 2022. doi: 10.18653/v1/2022.emnlp-main.775.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, et al. Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*, 2025b.

Muning Wen, Ziyu Wan, Weinan Zhang, Jun Wang, and Ying Wen. Reinforcing language agents via policy optimization with action decomposition. *arXiv preprint arXiv:2405.15821*, 2024.

Zhiheng Xi, Chenyang Liao, Guanyu Li, Yajie Yang, et al. Agentprm: Process reward models for llm agents via step-wise promise and progress. *arXiv preprint arXiv:2511.08325*, 2025.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. In *Advances in Neural Information Processing Systems*, volume 35, pp. 20744–20757, 2022.

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

Junxi Yin, Haisen Luo, Zhenyu Li, Yihua Liu, Dan Liu, Zequn Li, and Xiaohang Xu. Pinpointing crucial steps: Attribution-based credit assignment for verifiable reinforcement learning. *arXiv preprint arXiv:2510.08899*, 2025.

Chenchen Zhang. From reasoning to agentic: Credit assignment in reinforcement learning for large language models. *arXiv preprint arXiv:2604.09459*, 2026.

Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. Archer: Training language model agents via hierarchical multi-turn rl. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Yifei Zhou, Song Jiang, Yuandong Tian, Jason Weston, Sergey Levine, Sainbayar Sukhbaatar, and Xian Li. Sweet-rl: Training multi-turn llm agents on collaborative reasoning tasks. *arXiv preprint arXiv:2503.15478*, 2025.

A Full checkpoint results

Table 4: Official scores (%) at four checkpoints from the canonical held-out evaluations. ALFWorld official score equals binary success.

Task	Metric	Epoch 1	Epoch 2	Epoch 3	Epoch 4
ALFWorld PPO	score (%)	36.50	69.34	89.42	91.24
ALFWorld AttnSR	score (%)	37.59	70.44	91.24	97.08
WebShop PPO	score (%)	63.72	67.53	66.71	70.74
WebShop AttnSR	score (%)	65.74	68.35	69.58	69.71
ScienceWorld PPO	score (%)	22.89	31.49	34.59	44.53
ScienceWorld AttnSR	score (%)	27.12	35.57	41.15	46.35

Table 5: ALFWorld success (%) across checkpoints for the canonical positional control. All checkpoints use the full 274-task held-out set.

Method	Epoch 1	Epoch 2	Epoch 3	Epoch 4
PPO	36.50	69.34	89.42	91.24
Linear Step Reward	43.43	59.85	79.20	92.34
AttnSR	37.59	70.44	91.24	97.08

Table 6: Fine-grained ALFWorld task-family success at epoch 4 on the full held-out set. Columns use the compact family names *Pick* (pick and place simple), *Look* (look at object in light), *Clean*, *Heat*, *Cool*, and *Pick2* (pick two objects and place). Counts sum to 274 tasks.

	Pick	Look	Clean	Heat	Cool	Pick2	Overall
n	59	31	58	39	46	41	274
PPO (%)	100.00	96.77	86.21	89.74	86.96	87.80	91.24
AttnSR (%)	100.00	96.77	93.10	100.00	97.83	95.12	97.08
Δ (pp)	0.00	0.00	+6.89	+10.26	+10.87	+7.32	+5.84

B ALFWORLD CATEGORY-LEVEL RESULTS

C IMPLEMENTATION DETAILS

The selected layers are 8, 10, 12, 14, 16, and 18. We use full causal-prefix softmax attention and do not renormalize within the step submatrix. AA mass targets historical assistant tokens, while AO mass targets the observation paired with the historical step. Step rewards are placed at assistant-step end tokens; tool observations are excluded from the actor loss. The main implementation uses query blocks of 128 tokens and reconstructs tensor/context-parallel shards before graph aggregation. Format errors are assigned the ordinary terminal reward of zero rather than an extra penalty.

Each benchmark is trained independently from the instruction-tuned checkpoint. Table 7 reports the training manifests and collected-batch updates; epoch lengths use complete batches of eight, dropping the incomplete final batch when present. Training reward is binary success on ALFWorld, the official score in $[0, 1]$ on WebShop, and the final ScienceWorld score clipped to $[0, 100]$ and divided by 100. The native ScienceWorld step-reward increments are not supplied to PPO. Benchmark-specific expert-action and search bonuses are disabled.

Table 7: Canonical training-set sizes and update budgets, per method. One update uses one collected batch of eight trajectories.

Benchmark	Training tasks	Updates/epoch	Updates at epoch 4
ALFWorld	3,553	444	1,776
WebShop	10,587	1,323	5,292
ScienceWorld	3,592	449	1,796

D PAIRED SIGNIFICANCE ANALYSIS

Table 10 reports the full uncertainty analysis summarized in Section 4.2. Both policies decode greedily on identical held-out tasks, so the final-checkpoint comparison is paired at the task level over the exact-input-matched episodes defined in Appendix E. Wins and losses count tasks solved by exactly one policy, and the McNemar test is applied to those discordant pairs only. Bootstrap intervals resample the paired per-task differences 10,000 times.

E INFERENCE-TRAJECTORY DIAGNOSTICS

Source and measurement. We re-read the six final-checkpoint `generations/*_openai.jsonl` files referenced by the canonical evaluation CSVs.

Table 8: Framework and model configuration for the canonical runs.

Component	Configuration
Base model	Qwen2.5-7B-Instruct
Training framework	VERL with Megatron-based training
Rollout engine	vLLM
Hardware	8×H200 GPUs; 140 GB memory per GPU
Fine-tuning	LoRA rank 64; frozen base model
Trainable modules	Actor and critic LoRA adapters
Message format	Qwen standard chat format with JSON tool calls
Maximum context	32,768 tokens
Additional length limits	No per-turn token cap or turn-count cap
Training parallelism	Megatron TP=4, CP=2
Rollout parallelism	Tensor parallelism 4

Table 9: PPO, evaluation, and AttnSR hyperparameters used for the main results.

Component	Configuration
Batch size	8
PPO update epochs	1
Rollouts per prompt	1
Advantage estimator	GAE with trajectory-level agent advantage
Actor / critic learning rate	5×10^{-6} / 5×10^{-6}
Reference-policy KL coefficient	0.02
Checkpoint frequency	Once per data epoch
Decoding	Greedy; one rollout per held-out task
Evaluation sets	ALFWorld 274; WebShop 500; ScienceWorld 1,819
Attention layers	8, 10, 12, 14, 16, 18
Attention aggregation	Full causal-prefix softmax; no step-submatrix renormalization; AA/AO = 0.5/0.5
Query block size	128 tokens
Step-reward budget scale	0.25
Global reward mean	First-batch mean initialization; stop-gradient EMA coefficient 0.99
Hop decay	0.85
Temporal-bias correction	Weight -0.25; distance scale $\kappa = 4$
Target attention mass / focus	0.20 / 0.15
Reward placement	Final token of each assistant step; terminal reward on final step

The analysis includes 274 ALFWorld, 500 WebShop, and 1,819 ScienceWorld episodes per method. Recomputed means for official score, success, environment steps, parsed calls, serialized output length, and trace-parser events agree with the existing evaluation summaries. Per-episode numeric extracts and source-file hashes are retained with the analysis script; no training runs or new policy evaluations are introduced.

Environment steps use the evaluator `env_steps` field. Tool calls are counted from `openai_tool_calls`: they are parseable calls, not necessarily valid or successful environment actions. In these exports, parsed assistant tool-call turns equal the call count and are not reported as an independent measure of reasoning depth. Output length counts characters in the serialized `output`, including tool-call syntax and observations. We do not interpret it as model-generated token count, internal reasoning length, or inference FLOPs. Medians and P90 values are computed from the episode-level observations; P90 uses linear interpolation.

Outcome-conditioned and paired comparisons. Table 11 separates successful and unsuccessful episodes using the evaluator binary success flag. These two groups have different task membership under the two policies. The paired analysis in Table 12 additionally requires the same evaluation row, an exact input-text hash match, and an input unique within each benchmark set. This retains all 274 ALFWorld and 500 WebShop pairs. For ScienceWorld, 331 episodes are excluded because inputs differ or are non-unique, leaving 1,488 eligible pairs and 268 joint successes. No matching is inferred from row position alone.

On ALFWorld, the verified pairs include 20 tasks solved only by AttnSR, 4 only by PPO, 246 by both, and 4 by neither. This yields the observed net gain of 16 successes without claiming that every previously solved task remains solved. The corresponding WebShop counts are 48, 44, 157, and 251. Within ScienceWorld restricted eligible set, they are 173, 136, 268, and 911; these counts should not replace the full-test success totals. Successful-subset comparisons are descriptive and post-selected on outcomes of both policies, not estimates of a causal treatment effect.

Table 10: Paired evaluation-set uncertainty for the final-checkpoint comparison, computed over exact-input-matched episodes. Wins and losses count tasks solved by exactly one policy; p is the unadjusted two-sided exact McNemar test on those discordant pairs. Intervals are 10,000-resample percentile bootstrap intervals for the mean paired difference (AttnSR − PPO) in percentage points. ALFWorld official score equals binary success. Intervals quantify sampling uncertainty over the matched held-out tasks.

Task	Pairs	Wins	Losses	McNemar p	Δ Success (pp)	Δ Score (pp)
ALFWorld	274	20	4	0.002	+5.84 [+2.55, +9.49]	+5.84 [+2.55, +9.49]
WebShop	500	48	44	0.755	+0.80 [-3.00, +4.40]	-1.04 [-3.36, +1.20]
ScienceWorld	1488	173	136	0.040	+2.49 [+0.20, +4.70]	+1.85 [-0.37, +4.06]

Table 11: Full-set trajectory lengths of Benchmarks. Success and failure groups differ across policies and are not matched task subsets; Table 12 provides the restricted paired comparison.

		Successful		Unsuccessful	
Task	Method	n	Mean steps	n	Mean steps
ALFWorld	PPO	250	11.16	24	103.83
	AttnSR	266	12.69	8	109.12
WebShop	PPO	201	5.22	299	6.41
	AttnSR	205	6.62	295	9.00
ScienceWorld	PPO	520	11.32	1299	38.43
	AttnSR	562	8.00	1257	27.55

Tool use and termination. Table 13 distinguishes exported trace-parser events from the evaluator terminal format-error flag. Most parser events are unmatched call/response pairs, not malformed JSON. Specifically, ALFWorld PPO has 23 missing-response events; AttnSR has 8 missing-response and 4 orphan-response events. ScienceWorld PPO has 426 missing-response and 10 call-parsing events, while AttnSR has 245 missing-response events and no call-parsing events. WebShop has zero events under PPO and 3 missing-response events under AttnSR. An episode may contain multiple events; consequently the event count per episode differs from the fraction of affected episodes. We do not equate these exported trace diagnostics with runtime tool-execution failures.

Natural termination is also not a success indicator. On ScienceWorld its rate increases from 12.15% to 18.42%, so earlier unsuccessful stopping may contribute to the full-set length reduction; this is why we separately report jointly successful trajectories. Exact adjacent-call repetition does not uniformly decrease either: ScienceWorld pooled repetition fraction increases from 15.35% to 16.30%. Repeated commands can be necessary in stateful environments and are not labeled invalid or redundant without inspecting the intervening state. On WebShop, the parsed-call totals are 824 search and 2,143 click calls for PPO, versus 1,088 search and 2,925 click calls for AttnSR. This confirms that the higher interaction count involves both search and navigation rather than a change confined to one tool.

F Scope of supplementary runs

Table 1 includes a one-epoch GRPO reference with four rollouts per prompt. Table 2 separately reports the canonical positional control, an asynchronous optimization comparison, and a Qwen3 backbone comparison. The supplementary results are interpreted within their respective settings and are not pooled with the primary comparison for component attribution.

Asynchronous configuration. Both variants use Qwen2.5-7B-Instruct, rank-64 actor and critic LoRA, batch size 8, actor/critic learning rates of 5×10^{-6}, a 32,768-token context, and four data epochs on ALFWorld. This configuration uses asynchronous collection with a replay queue and rollout-policy token log-probabilities. Direct token importance sampling masks ratios outside $[0.7, 6.0]$. Observation tokens are skipped in GAE, with actor $\lambda = 1 - 1/(1.5n_i)$ for n_i trainable assistant tokens and critic $\lambda = 1$. Two critic passes are performed per actor pass; critic LoRA is restricted to MLP projections, with a trainable value head. The critic is initialized from the instruction-

Table 12: Environment steps on exact-input-matched episodes where both policies succeed. Eligible pairs require the same evaluation row and identical input text, unique within each benchmark set. All ALFWorld and WebShop episodes qualify; ScienceWorld retains 1,488 of 1,819 episodes, excluding 331 with different or non-unique input text. Results are descriptive and restricted to the jointly successful subset.

Task	Eligible	Both succeed	PPO	AttnSR	Δ steps
ALFWorld	274	246	11.19	11.99	+0.80
WebShop	500	157	5.11	6.19	+1.08
ScienceWorld	1488	268	11.06	7.40	-3.66

Table 13: Tool-trace and termination diagnostics at epoch 4. Parse/ep. counts exported trace-parser events, and Parse% is the fraction of episodes with any such event. Format% and Natural% are the evaluator terminal flags. Repeat% is the pooled fraction of adjacent parsed-call pairs with identical function names and arguments; repetition need not imply an invalid or redundant action.

Task	Method	Parse/ep.	Parse%	Format%	Natural%	Repeat%
ALFWorld	PPO	0.084	8.39	0.00	0.00	0.04
	AttnSR	0.044	3.28	0.00	0.00	0.00
WebShop	PPO	0.000	0.00	0.00	0.00	0.04
	AttnSR	0.006	0.60	0.00	0.00	0.03
ScienceWorld	PPO	0.240	23.97	0.60	12.15	15.35
	AttnSR	0.135	13.47	0.22	18.42	16.30

tuned model without additional value pretraining. The AttnSR variant retains the canonical attention layers and shaping hyperparameters. Thus the comparison changes more than collection timing relative to canonical PPO.

Qwen3 configuration. The paired launch configuration uses Qwen3-4B-Instruct-2507 with the same main hyperparameters as the canonical study: rank-64 LoRA, actor/critic learning rates of 5×10^{-6}, batch and PPO minibatch size 8, one rollout per prompt, one PPO epoch per batch, reference KL coefficient 0.02, and a 32,768-token context. Training uses TP=4 and CP=2; rollout TP=4. AttnSR retains the selected layers and reward settings. Four ALFWorld data epochs give 1,776 updates. Greedy evaluation of the epoch-4 policies on all 274 held-out tasks yields 241 successes for PPO and 250 for AttnSR. This is supplementary evidence across backbones, not a test isolating model size or individual reward components.

G NOTATION

Table 14 collects the symbols used in Sections 2–3. Episode indices i are omitted in the figures and, where unambiguous, in the text.

Table 14: Notation. Hyperparameter values refer to the canonical AttnSR runs.

Symbol	Meaning
i, t	episode index; collected training-batch index
K_i	number of assistant steps in episode i
k, j	step indices, $1 \le k < j \le K_i$
$S_{i,k}$	interaction step: assistant message and observation, when present
$A_{i,k}$, $O_{i,k}$	assistant message; associated environment observation
$\mathcal{I}_{i,k}$	token indices in the assistant span of step k
u	token index into the serialized trajectory
$e_{i,k}$	endpoint: last trainable token of step k
$r_{i,u}$	reward written at token u
R_i, $\bar{R}_t$	terminal reward; EMA updated using batch t before shaping
M_t	mean terminal reward in batch t; initializes $\bar{R}_1 = M_1$
l, $\mathcal{L}$, L	layer index; selected layers; $L = \lvert\mathcal{L}\rvert = 6$
$G^{(l)}_{i,\mathrm{AA}}$, $G^{(l)}_{i,\mathrm{AO}}$	step attention to historical assistant tokens; to paired observations
$G^{(l)}_i$	mixed step attention, $0.5G^{(l)}_{i,\mathrm{AA}} + 0.5G^{(l)}_{i,\mathrm{AO}}$
$a^{(l)}_{i,k}$	backtracked relevance of step k at layer l (Eq. 3)
$\Delta_{i,k}$, $d_{i,k}$	steps from k to the terminus, $K_i - k$; proximity feature $\exp(-(\Delta_{i,k} - 1)/\kappa)$
$p^{(l)}_{i,k}$, $p_{i,k}$	per-layer allocation; geometric-consensus allocation (Eq. 5)
$m^{(l)}_i$, $H^{(l)}_i$	terminal-step covered mass; normalized entropy of $p^{(l)}_i$
C_i	confidence gate, $C_i \in [0, 1]$
B_i, $s_{i,k}$	signed shaping budget; step reward $B_i p_{i,k}$ (Eq. 13)
$\hat{A}_u$, $\widetilde{A}_u$, y_u	unnormalized advantage; normalized actor advantage; critic target
V^{old}_u, V_ϕ	pre-update critic value; critic network
sg, norm, TV	stop-gradient; standardization; total-variation distance
$\rho = 0.85$	hop decay in the backtracking recurrence
$\eta = -0.25$, $\kappa = 4$	recency-correction weight and proximity decay
$T = 1$, $\epsilon = 10^{-12}$	allocation softmax temperature; lower bound before logarithms
$\beta = 0.25$	budget scale, bounding $\lvert B_i\rvert$