ANCHORED ANYTIME VALUE LEARNING FOR LONG-HORIZON AGENT REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon agent training calls for *anytime* value learning that decouples from trajectory-completion constraints. We introduce a barrier-synchronized streaming framework that optimizes in-flight trajectory chunks, eliminating straggler-induced idle time while preserving chunk-level policy freshness. Value learning over nonterminal chunks, however, corrupts along two axes: it inflates the *magnitude* variance of the bootstrap propagated within a trajectory, and it scrambles the *ordering* of values across concurrent trajectories. Mechanistic analysis shows: by unfolding the anisotropic observation manifold into a higher-resolution belief space, thinking widens task-level (between-task) separation far faster than it changes surface-level (within-task) variation, boosting the Fisher signal-to-noise ratio of the value representation. We then propose **Anchored Anytime Value Learning (A2VL)**: (1) *longitudinal magnitude anchoring* shrinks noisy boundary bootstraps toward the low relative-variance belief-state value to limit temporal error propagation; and (2) *transverse ordering anchoring* imposes pairwise ordering constraints across concurrent trajectories using the well-separated belief-state ordering as a detached reference. Across WebShop, ALFWorld, and Crafter, A2VL stabilizes anytime training across chunk granularities, matching episode-synchronous task performance while substantially improving wall-clock throughput.

1 INTRODUCTION

On-policy reinforcement learning (RL) has emerged as a central paradigm for training language agents through interactive environments, from reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022) to agentic training (Yao et al., 2023; Cheng et al., 2025). However, current implementations predominantly require complete trajectories before advantage computation. Synchronous methods such as GRPO (Shao et al., 2024), RLOO (Ahmadian et al., 2024), and PPO (Schulman et al., 2017) with full generalized advantage estimation (GAE) (Schulman et al., 2016) wait for entire groups or batches of trajectories to terminate. Asynchronous methods such as AReaL (Fu et al., 2025) and SAO (Hou et al., 2026), building on classical actor-learner architectures (Espeholt et al., 2018), update the model as trajectories arrive but suffer from off-policy staleness. In long-horizon tasks, both methods must endure time-consuming rollouts due to task horizons, environment latency, and tool execution delays. Synchronous methods suffer even more from the heavy-tailed distribution of rollout completion times, where a small number of slow rollouts hold back the other completed ones.

Long-horizon tasks demand on-policy streaming training: the rollouter should admit trajectory chunks to the optimizer as soon as a minimum batch size is reached, rather than waiting for complete episodes. We realize this with barrier-synchronized updates that keep the admitted data locally on-policy even when episode durations vary widely (Section 2), and build value learning on chunk-level GAE with periodic Monte Carlo (MC) calibration from completed trajectories. This substrate enables *anytime* value learning—value updates at any chunk boundary rather than only at termination.

While classical n-step temporal-difference (TD) methods—including SARSA, the canonical on-policy bootstrap algorithm for control (Sutton & Barto, 2018)—support streaming training in principle, chunk-level training with mostly nonterminal data requires bootstrapped targets that accumulate errors across chunk boundaries. Language-form observations aggravate this bootstrapping problem by injecting surface-level noise that corrupts value estimates along two distinct axes. *Magnitude*: within a single trajectory the underlying decision state changes little from turn to turn, yet the

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.

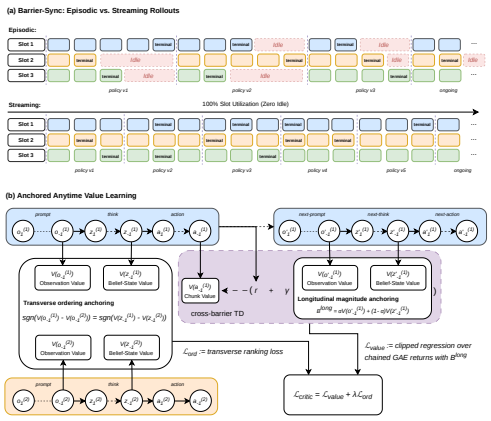


Figure 1: **Anchored Anytime Value Learning (A2VL) framework.** (a) Barrier-synchronized streaming versus episodic rollouts. Each row is one rollout slot; completed chunks are admitted to the optimizer as soon as a minimum batch size is reached, so the streaming arm keeps every slot busy (*100% slot utilization, zero idle*), while the episodic arm lets finished slots sit empty waiting for the slowest straggler (*idle*).(b) Each slot executes the `prompt`→`think`→`action` sequence $o \to z \to a$; the critic reads three positions per row—*Observation Value* $V^o := V_\phi(s, 0)$, *Belief-State Value* $V^z := V_\phi(s, p_{\text{think}})$, and *Chunk Value* $V(a)$ at the closing action position—and produces two belief-derived anchors. *Longitudinal magnitude anchoring* shrinks the open-boundary bootstrap toward the belief-state value via $B^{\text{long}} = \alpha V^o + (1-\alpha)V^z$, where $\alpha \in [0, 1]$ controls the shrinkage strength; the resulting target $r + \gamma B^{\text{long}}$ is regressed by the chunk value of the preceding row (*cross-barrier TD*), with γ the discount factor. *Transverse ordering anchoring* pairs concurrent trajectories (e.g., (1) and (2)) and drives the observation-position ordering to match the belief-state ordering, $\text{sgn}(V_i^o - V_j^o) = \text{sgn}(V_i^z - V_j^z)$, with a confidence scale $\tau > 0$ controlling the softness of the belief-state sign. The full critic objective is $\mathcal{L}_{\text{critic}} = \mathcal{L}_{\text{value}} + \lambda_{\text{ord}}\mathcal{L}_{\text{ord}}$, where $\mathcal{L}_{\text{value}}$ is clipped regression against chained-GAE returns built on B^{long} and $\mathcal{L}_{\text{ord}}$ is the transverse ranking loss.

surface variability of natural-language observations makes the observation-position value swing widely, so the bootstrap magnitude propagated to preceding segments is unreliable. *Ordering*: across concurrent rollouts the on-policy objective implies that successful trajectories should be valued above unsuccessful ones, yet observation noise scrambles this ordering, so the critic's relative values disagree with the returns they are meant to rank. Any stabilizer must therefore address both the magnitude of individual bootstraps and the ordering across trajectories.

We address this challenge through **Anchored Anytime Value Learning (A2VL)**, which stabilizes both axes by anchoring the critic to the model's own reasoning block—the *think* content—read as an explicit belief state (DeepSeek-AI et al., 2025a). Although the generated text is lexically unconstrained, it forms a task-conditioned abstraction that integrates trajectory history, averages over the surface form of the current observation, and incorporates anticipated plans; belief states are in fact linearly decodable from residual-stream activations during chain-of-thought reasoning (Balcells et al., 2026). As we show in Section 3, this abstraction is not merely a denoised copy of the observation: thinking geometrically reorganizes the value representation so that distinct tasks separate far faster than their surface variation grows, sharpening the value signal the critic must read. Reading this reorganization along each corrupted axis yields the two anchors of A2VL:

1. **Longitudinal magnitude anchoring** targets the magnitude axis. At open rollout boundaries, the noisy observation-position bootstrap is shrunk toward the belief-state value, whose low variance *relative to* between-task separation makes it a stabler reference for the magnitude propagated to preceding segments.

2. **Transverse ordering anchoring** targets the ordering axis. Pairwise constraints align observation-position values with the well-separated belief-state ordering, using detached belief-state value differences as a high-resolution ranking reference.

Our contributions:

Streaming training framework for language agents. We implement a barrier-synchronized streaming on-policy framework for long-horizon language agents that shifts the atomic update unit from complete episodes to trajectory chunks. This system substrate reduces straggler-induced idle time while preserving chunk-level policy freshness.

Anchored anytime value training. We formulate the agent's reasoning block as an information-condensed belief state and derive two complementary anchors in A2VL: longitudinal magnitude anchoring stabilizes bootstrap targets across chunk boundaries, while transverse ordering anchoring preserves cross-trajectory value rankings.

Empirical quality-throughput frontier. On three agent benchmarks (ALFWorld, WebShop, and Crafter) across three model scales (1.5B, 4B, and 7B), A2VL performs better in each model group while training under continuous streaming rollout, especially in the long horizon tasks. Ablations attributes the improvement to the proposed two belief-state anchors.

2 Streaming Training for Long-Horizon Language Agents

2.1 Barrier-synchronized chunk admission

We model interaction as a partially observable Markov decision process (POMDP) $(\mathcal{S}, \mathcal{A}, \mathcal{O}, P, R, \Omega)$ (Kaelbling et al., 1998; Sutton & Barto, 2018), where P is the transition function, R the reward function, and Ω the initial-state distribution. At environment step t, the agent observes o_t, generates a reasoning block z_t (`think`), executes action a_t, and receives reward r_t, yielding the sequence $o_t \to z_t \to a_t \to o_{t+1}$ (the `prompt`→`think`→`action` rows in Figure 1b). Figure 1(a) contrasts this design with episodic synchronization. A fixed pool of rollout *slots* collects independent episodes and packages completed interaction steps into trajectory chunks. Under episodic synchronization, a slot that has reached its `terminal` state must idle until the slowest straggler in the batch finishes, so slots sit empty between policy updates. Streaming instead admits chunks as soon as they reach a minimum batch size: rollout pauses while the optimizer updates the model and then resumes under the updated policy, keeping every slot busy (100% *slot utilization*). The data admitted to each update remain locally on-policy, while an individual trajectory may span multiple policy versions (*policy v1*, *v2*, ... in Figure 1a). It admits ongoing trajectories without waiting for every active rollout to terminate. Consequently, most training rows end at open chunk boundaries rather than terminal states, and their targets must bootstrap from the critic instead of terminating in observed outcomes.

Let $V_\phi(s_t, p)$ denote the critic value at token position p in the response generated from state context s_t. We use two positions:

$$V_t^o := V_\phi(s_t, 0), \qquad V_t^z := V_\phi(s_t, p_{\text{think}}), \tag{1}$$

where $p = 0$ is the first response position and yields the observation-position value V_t^o, while p_{think} is the final position selected by `think_mask` and yields the belief-state value V_t^z. These are the *Observation Value* and *Belief-State Value* heads in Figure 1(b), read at the observation boundary and at the end of the `think` block, respectively; the value at the closing action position of a row is its *Chunk Value*, which receives the bootstrap target defined below. Here, *belief state* denotes the task-conditioned abstraction formed by the reasoning trace, which integrates history, the current observation, anticipated plans; it is not assumed to be an exact Bayesian posterior over latent states.

2.2 Chained GAE with periodic outcome calibration

Within each row, GAE computes

$$\delta_t = r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t), \qquad \hat{A}_t = \sum_{\ell \geq 0} (\gamma\lambda)^\ell \delta_{t+\ell}. \tag{2}$$

Chained GAE continues this backward recursion across rows from the same trajectory: the boundary value of row k bootstraps the scan of row $k-1$. A completed trajectory instead terminates with a zero bootstrap and yields an MC return $G_k^{\text{MC}} = \sum_{t \geq k} \gamma^{t-k} r_t$. At periodic calibration steps, the implementation blends eligible bootstrap targets with these realized returns:

$$B_k^{\text{anchor}} = (1-\beta) V_k^o + \beta G_k^{\text{MC}}, \tag{3}$$

where β controls the strength of the anchor. This calibration affects value targets only and does not enter the clipped policy objective.

2.3 Bootstrap instability under streaming updates

Let π denote the current policy and $e_t = V_\phi(s_t) - V^\pi(s_t)$ the critic error under π. For a transition from an ongoing episode, the one-step target satisfies

$$\hat{y}_t = r_t + \gamma V_\phi(s_{t+1}) = V^\pi(s_t) + \gamma e_{t+1} + \epsilon_t, \tag{4}$$

where ϵ_t captures transition and approximation noise. The error at an open boundary therefore enters the target propagated to the preceding row. Repeated bootstrapping can accumulate this error over a long trajectory before a terminal return becomes available. Periodic outcome calibration restores exact return information for completed trajectories, but provides no direct constraint on the intervening open boundaries. Language observations amplify this instability along the two axes of Section 1—magnitude within a trajectory and ordering across concurrent trajectories. Both call for one fix—a reference read from the belief-state value V^z—whose geometric justification we establish next (Section 3) before turning it into two anchors (Section 4).

3 The Reasoning Block as a Value-Resolution Anchor

Before constructing the anchors, we establish why the belief-state value V^z is a more reliable reference than the observation-position value V^o. We analyze the representation geometry at the two positions on the *base* model (Qwen2.5-1.5B-Instruct, before any RL), using real WebShop rollout states from two policy checkpoints (the initial policy and the policy after 140 training steps; 20 states each). Each state is augmented with 5 semantic-preserving perturbations (action-list shuffling, observation-item reordering, history truncation) to form an equivalence class of surface variants that share the same decision state (200 forward passes total). A good anchoring position should collapse these variants while keeping distinct tasks apart, i.e. maximize the Fisher signal-to-noise ratio $\mathrm{SNR} = \text{between-task dispersion}/\text{within-task dispersion}$.

Three findings emerge (Figure 2). **(i) The observation position is anisotropic.** Pairwise cosine similarity across variants at V^o is 0.998 and the Fisher SNR is only 1.62: distinct task states are crammed into a narrow cone where their values are hard to resolve. **(ii) The think block reorients rather than rescales.** Letting h^o and h^z denote the critic hidden states at the observation and think-end positions, the think-end representation is a near-orthogonal reconstruction of the observation embedding ($\cos(h^o, h^z) \approx 0.13$) at conserved norm ($\|h^z\|/\|h^o\| = 0.998$), so the change is a genuine rotation on the hypersphere, not a trivial scale change. **(iii) The SNR rises $4.8\times$ through between-task separation, not variance suppression.** Thinking does not shrink surface variability in absolute terms—the within-task L^2 dispersion in fact *grows* from 75.2 to 239.2—but it pushes distinct tasks apart at a far higher rate, lifting the SNR to 7.70. Figure 2 makes this decomposition visible on a shared axis scale: at V^o the task centroids collapse into a single overlapping cluster ($\bar{d} = 7.2$), whereas at V^z they spread across the projection ($\bar{d} = 38.6$) while the within-task ellipses stay comparable in size. The gain therefore comes from the numerator of the SNR. It is robust across rollout distributions: the same protocol on initial-policy states yields an even larger $9.4\times$ increase ($3.09 \to 29.21$). Because these estimates come from a focused probe (20 decision states per checkpoint, each with five surface-preserving variants), we treat the exact ratios as indicative rather than precise; Appendix C corroborates the qualitative gap—collapsed observation-position centroids versus well-separated belief-state centroids—across four training checkpoints with 128 states each.

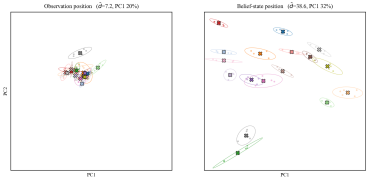


Figure 2: **Representation geometry at observation vs. belief-state positions** (shared axis scale; IQR-filtered; 5 variants per task). Task centroids collapse at V^o ($\bar{d} = 7.2$) but separate at V^z ($\bar{d} = 38.6$); within-task ellipses stay comparable.

This geometry gives each corrupted axis of the bootstrap its own remedy, and A2VL simply reads it twice. The critic's value head reads exactly these two positions, so V^z is computed from the better-separated think-end representation while V^o is computed from

the anisotropic observation representation; the geometry above is therefore the conditioning of the inputs the two value estimates consume. Because V^z has low variance *relative to* between-task separation, it is a stable *magnitude* reference for the open-boundary bootstrap; because it orders tasks by value at high resolution, it is a reliable *ordering* reference across concurrent trajectories. We treat this geometry as an inductive bias the pretrained model already exhibits—anchoring the critic here *exploits* rather than imposes it—and encourage the trained critic to preserve it, without assuming it persists unchanged throughout RL. Section 4 turns the two readings into two anchors.

4 Anchored Anytime Value Learning

Each anchor acts on one axis of the belief-state geometry of Section 3 (Figure 1b): longitudinal magnitude anchoring on the within-trajectory bootstrap, and transverse ordering anchoring on values across concurrent trajectories (rows (1) and (2) in Figure 1b). Together they supply chunk-level constraints between the sparse outcome-calibration events of Section 2.

4.1 Longitudinal magnitude anchoring

As illustrated in the *Longitudinal magnitude anchoring* block of Figure 1(b), consider the open boundary row k visited first in the backward scan of an ongoing trajectory. Its observation-position value V_k^o supplies the bootstrap propagated to row $k-1$, but its magnitude is inflated by observation noise. A2VL therefore shrinks this bootstrap toward the belief-state value V_k^z, which Section 3 identified as the stabler, low-relative-variance reading:

$$B_k^{\text{long}} = \alpha V_k^o + (1-\alpha) V_k^z, \qquad \alpha \in [0,1]. \tag{5}$$

The resulting cross-row target $\hat{y}_{k-1} = r_{k-1} + \gamma B_k^{\text{long}}$ is regressed by the closing action-position value of row $k-1$ (the *Chunk Value* in Figure 1b), the step labeled *cross-barrier TD*. Setting $\alpha = 1$ recovers the original observation-position bootstrap, while $\alpha = 0$ uses only the belief-state value. Shrinking is a variance-reduction move: because V_k^o's magnitude is high-variance under surface noise, blending it toward the lower relative-variance V_k^z reduces target variance even though V_k^z may carry its own bias, and α controls this bias–variance tradeoff.

The longitudinal anchor is active only when the backward scan is at the open end of a chain (`chain_i`$= 0$), the trajectory is nonterminal (`chain_done` is false), and the row contains at least one think token. Terminal trajectories retain the standard zero bootstrap. Return construction runs without gradient tracking, so the anchored value changes the regression target without creating a direct gradient path through either component. The computation reuses predictions already available to GAE and requires no additional trajectory pass.

Because V_k^o and V_k^z are read at the same decision boundary, the anchor changes the propagated magnitude without introducing a state mismatch, and the appropriate strength α is determined empirically rather than by assuming V_k^z is uniformly more accurate.

4.2 Transverse ordering anchoring

The second mechanism preserves relative value order across concurrent trajectories, taking the high-resolution belief-state ordering of Section 3 as its reference. Within each critic mini-batch, we group rows by trajectory identifier, discard terminal trajectories, and retain the latest row from each remaining trajectory. A random permutation of these N rows yields $\lfloor N/2 \rfloor$ disjoint pairs $\mathcal{P}$. Figure 1(b) shows one such pair as the concurrent trajectories (1) and (2): the anchor drives the observation-position ordering to match the belief-state ordering, $\text{sgn}(V_i^o - V_j^o) = \text{sgn}(V_i^z - V_j^z)$.

For each pair (i, j), define

$$\Delta_{ij}^z = \text{sg}(V_i^z - V_j^z), \qquad \Delta_{ij}^o = V_i^o - V_j^o, \tag{6}$$

$$w_{ij} = \tanh(\Delta_{ij}^z / \tau), \qquad q_{ij} = w_{ij} \Delta_{ij}^o, \tag{7}$$

where sg denotes stop-gradient and $\tau > 0$ controls the confidence scale. We optimize

$$\mathcal{L}_{\text{ord}} = \frac{1}{|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} \text{softplus}(-q_{ij}). \tag{8}$$

This objective contains no explicit margin. The detached difference Δ_{ij}^{z} instead serves as the ordering anchor: its sign specifies the preferred order, and its magnitude determines the strength of that constraint through w_{ij}. The softplus penalty decreases smoothly as the observation-position values align with this anchored order.

The asymmetric construction serves three purposes. First, $\nabla_{V^z}\mathcal{L}_{\mathrm{ord}} = 0$, so the reference order cannot move to reduce the loss it supervises. Second, $|w_{ij}|$ downweights pairs whose belief-state values are nearly tied. Replacing $\tanh$ with a hard sign would preserve the stop-gradient property but assign full weight to uncertain pairs and switch direction discontinuously at zero. Third, disjoint random pairing keeps the anchoring cost linear in the number of eligible trajectories and aligns its sampling scope with value regression.

4.3 Critic and actor objectives

Let $\mathcal{L}_{\mathrm{value}}$ denote clipped regression against the returns produced by chained GAE with Equation (5); in Figure 1(b) this is the $\mathcal{L}_{\mathrm{value}}$ term, paired with the transverse ranking loss $\mathcal{L}_{\mathrm{ord}}$. The full critic objective is

$$\mathcal{L}_{\mathrm{critic}} = \mathcal{L}_{\mathrm{value}} + \lambda_{\mathrm{ord}}\mathcal{L}_{\mathrm{ord}}. \tag{9}$$

The actor follows the standard clipped PPO objective, using advantages from the same return-construction pass. The transverse ordering loss and periodic MC calibration are applied only to the critic; neither is added to the actor objective. A2VL therefore changes neither rollout generation nor the policy objective.

5 Experiments

5.1 Experimental Setup

We evaluate on three partially observable, long-horizon agent benchmarks. **ALFWorld** (Shridhar et al., 2021) pairs language interaction with embodied household tasks across six subtask categories (Pick, Look, Clean, Heat, Cool, Pick2), reporting success rate (%). **WebShop** (Yao et al., 2022) tests multi-step web interaction with success rate (%) and a terminal task score (0–100) based on product matching. **Crafter** (Hafner, 2022) measures long-horizon exploration through 22 achievement categories, reporting the fraction of achievements unlocked (%).

We compare the results among (i) the base model. (ii) episodic barrier-sync **PPO** and **GRPO**, which wait for complete trajectories then broadcast the episode return to each turn. (iii) streaming barrier-sync **ChainedGAE** with chained GAE across chunk boundaries. Previous work (Chen et al., 2025b) attempts techniques such as dual discounting GAE and critic pre-training for stable critic learning. We enable periodic MC outcome calibration from completed trajectories, which completes our fundamental training pipeline. (iv) streaming barrier-sync **A2VL** , which is based on ChainedGAE but with longitudinal magnitude anchoring, and transverse ordering anchoring.

Our primary backbone is Qwen2.5-1.5B-Instruct; Qwen3-4B and Qwen2.5-7B-Instruct test whether the findings transfer across model architecture and scale. All comparisons share the same initialization, interaction budget, number of optimizer steps, and hardware allocation. The main difference resulting from barrier-sync mechanism is the number of environment steps released together for training at each training step. To keep the efficiency comparison conservative rather than favorable to streaming, we additionally cap the maximum training turns in streaming to below 80% of those allowed for non-streaming in ALFWorld and WebShop, and 20% in Crafter (whose rollouts are frequently early-stopped); the streaming arm therefore trains under a tighter per-episode turn budget than the episodic baselines it matches or exceeds. We average the results over at least three seeds. The streaming pipeline is implemented in verl (Sheng et al., 2024). Unless stated, we use $\gamma = 1$, $\lambda = 1$, $\alpha = 0.5$, $\lambda_{\mathrm{ord}} = 0.5$, and $\tau = 0.5$; Appendix B lists environment-specific settings and optimizer details.

5.2 Main Results

Overall task performance. Table 1 shows the primary evaluation results across three benchmarks and three model scales. We compare untrained prompting, episodic methods (PPO and GRPO under full-trajectory synchronization), and our streaming methods (ChainedGAE and A2VL). The

Table 1: Performance on ALFWorld, WebShop, and Crafter. Results are averaged over 3 random seeds. For ALFWorld, we report the average success rate (%) for each subtask as well as the overall result. For WebShop, we report both the average score and the average success rate (%). For Crafter, we report the average achievement score(%).

		ALFWorld							WebShop		Crafter
Method	Barrier-Sync	Pick	Look	Clean	Heat	Cool	Pick2	All	Score	Succ.	Score
Qwen2.5-1.5B-Instruct											
1.5B		6.5	0.0	3.4	7.1	10.0	0.0	4.7	20.0	5.5	14.5
PPO	Episodic	67.7	66.7	65.5	78.6	60.0	59.1	65.6	83.0	63.0	10.1
GRPO	Episodic	77.4	79.2	60.3	60.7	77.5	79.5	72.3	76.0	60.0	10.5
ChainedGAE	Streaming	100.0	100.0	96.6	100.0	100.0	95.5	98.4	80.1	68.5	35.8
A2VL	Streaming	100.0	100.0	100.0	100.0	100.0	100.0	100.0	82.9	71.5	39.6
Qwen3-4B											
4B		32.3	16.7	34.5	35.7	45.0	36.4	34.4	20.2	4.7	10.1
PPO	Episodic	77.4	83.3	55.2	57.1	80.0	77.3	71.1	79.3	60.1	15.2
GRPO	Episodic	54.8	83.3	72.4	78.6	45.0	68.2	54.8	72.2	55.6	16.7
ChainedGAE	Streaming	96.8	100.0	100.0	100.0	95.0	86.4	96.1	83.4	67.2	31.9
A2VL	Streaming	96.8	100.0	100.0	100.0	100.0	100.0	99.2	82.0	64.8	32.7
Qwen2.5-7B-Instruct											
7B		6.5	0.0	0.0	14.3	5.0	0.0	3.9	26.4	7.8	0.0
PPO	Episodic	83.9	91.7	86.2	50.0	90.0	86.4	82.8	81.9	68.5	6.6
GRPO	Episodic	67.7	75.0	100.0	64.3	65.0	81.8	77.3	80.8	68.8	22.1
ChainedGAE	Streaming	100.0	100.0	100.0	100.0	100.0	100.0	100.0	85.0	71.9	30.5
A2VL	Streaming	100.0	100.0	100.0	100.0	100.0	100.0	100.0	82.1	73.4	43.1

central question is whether streaming credit assignment can match or exceed episodic training while permitting continuous, in-flight rollout. The table answers this affirmatively: A2VL is the strongest method in every model group, and on two of the three benchmarks it wins by a wide margin.

ALFWorld. Streaming methods nearly solve the suite. A2VL attains a perfect 100.0% overall success rate at both 1.5B and 7B and 99.2% at 4B, with per-subtask scores at or near 100.0% across all six task families. This exceeds the episodic baselines: at 1.5B, A2VL improves over PPO and GRPO by 34.4 and 27.7 points; at 7B, over PPO and GRPO by 17.2 and 22.7 points. ChainedGAE alone already lifts overall success to 98.4%–100.0%, indicating that the streaming substrate accounts for most of the ALFWorld gain.

WebShop. A2VL delivers the best success rate at each scale (71.5% at 1.5B, 64.8% at 4B, 73.4% at 7B), surpassing episodic PPO and GRPO, while remaining competitive or better on task score. In WebShop, episodic training is already a strong baseline here; streaming nonetheless closes the remaining success-rate gap without sacrificing throughput.

Crafter. The open-ended, long-horizon setting separates the methods most sharply. A2VL reaches achievement scores of 39.6%/32.7%/43.1% across the three scales, roughly 2–6$\times$ the episodic baselines. Episodic critics, trained only on synchronized full-episode returns, struggle to propagate credit across Crafter's sparse achievement structure, whereas the belief-state anchoring in A2VL sustains a usable value signal throughout the long rollout.

Scale trend. The advantage of streaming credit assignment holds across 1.5B, 4B, and 7B backbones. The gap over episodic training is largest exactly where the horizon is longest (Crafter, 200 turns) and smallest when the horizon gets short (WebShop, 15 turns), consistent with A2VL targeting the bootstrap failure that episodic training incurs under long, streamed rollouts.

5.3 Ablation Studies

Streaming framework ablation. Figure 3 dissects the streaming substrate on WebShop (1.5B, chunk-1). Three observations emerge.

MC calibration is essential. Removing the periodic MC outcome calibration from the streaming baseline (ChainedGAE w/o MC) severely degrades learning: the critic receives no grounded return signal and the policy stagnates below 0.65 validation accuracy. The MC-calibrated base setting (ChainedGAE w MC) recovers the full learning curve, confirming that periodic outcome anchoring is the necessary backbone of the streaming framework.

Individual optimizations saturate. Adding either the longitudinal magnitude anchor (ChainedGAE + shrinkage) or the transverse ordering anchor (ChainedGAE + rankloss) to the MC-calibrated base setting yields curves that nearly overlap with ChainedGAE (w MC) on WebShop, indicating that the streaming substrate with MC calibration already captures most of the learnable signal on this task.

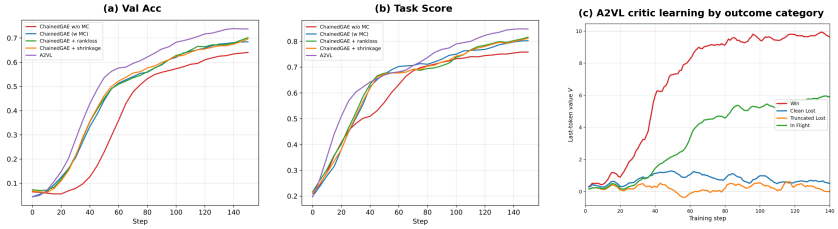


Figure 3: **Streaming framework analysis (WebShop 1.5B, chunk-1). (a)** Validation accuracy and **(b)** task score under module ablation. ChainedGAE (w MC) is the base setting with periodic MC calibration; ChainedGAE w/o MC removes the MC calibration; ChainedGAE + rankloss/shrinkage add the transverse ordering / longitudinal magnitude anchor individually; A2VL combines both anchors. MC calibration is the indispensable backbone; individual anchors saturate on WebShop; the two-anchor combination lifts performance above all single-anchor variants. **(c)** A2VL critic learning by trajectory outcome category (3-run average, smoothed): Win trajectories converge to the true return ($V \approx 10$), Clean Lost and Truncated Lost are suppressed toward zero, and In Flight settles at an intermediate value ($V \approx 6$). The widening separation between Win and Lost categories indicates decreasing inversion gap and improving calibration during training.

Table 2: Chunk-size ablation on WebShop (1.5B). Val Acc and Task Score at different chunk sizes.

Chunk size	Val Acc	Task Score
1	71.5±0.8	82.9±2.3
2	72.6±0.4	85.2±3.1
4	72.7±0.5	83.3±2.6

Table 3: Directionality study of the ordering anchor on WebShop (1.5B, chunk-1). Gradient flow indicates which estimate is detached ($\nabla = 0$) versus updated by $\mathcal{L}_{\text{ord}}$.

Variant	Gradient flow	Val Acc
Think → Obs	$\nabla_{V^z} = 0, \nabla_{V^o} \neq 0$	71.5±0.8
Obs → Think	$\nabla_{V^o} = 0, \nabla_{V^z} \neq 0$	15.3±3.7
Symmetric	$\nabla_{V^z} \neq 0, \nabla_{V^o} \neq 0$	0

Two anchors together lift performance. A2VL, which combines both longitudinal magnitude anchoring and transverse ordering anchoring on top of MC calibration, consistently outperforms every single-anchor variant and the MC-only base setting, reaching the highest validation accuracy and task score. This supports the view that the two anchors address complementary failure modes of the bootstrap and their benefits are additive rather than redundant.

Chunk-size ablation. Table 2 illustrates A2VL performance at different chunk sizes on WebShop (1.5B). The streaming framework remains stable across chunk granularities: validation accuracy varies by only 1.2 percentage points (71.5% at chunk-1 to 72.7% at chunk-4), and task score stays within a 2.3-point band (82.9–85.2). Chunk-2 achieves the highest task score (85.2), while chunk-4 matches chunk-2 on accuracy (72.7 vs 72.6) with a slightly lower task score (83.3). The narrow spread suggests that A2VL's belief-state anchoring is robust to the choice of chunk size and does not rely on a particular granularity to stabilize critic learning.

Directionality study. To verify the asymmetric design, Table 3 compares our detached think-to-observation anchoring against the reversed (observation-to-think) and symmetric variants. The reversed direction collapses to 15.3%—detaching V^z and updating V^o forces the anisotropic observation position to carry the ordering signal, which it cannot resolve (Section 3). The symmetric variant, where both estimates receive gradient, fails completely (0%): the two value heads interfere and neither position retains a stable ordering reference. Only the think-to-observation direction, which reads the ordering signal from the well-separated belief-state representation, preserves performance.

5.4 Systems Throughput

To measure the systems benefit of streaming execution, we compare episodic PPO against streaming PPO on ALFWorld (1.5B, 4×GPU). Both methods use identical model, environment, and rollout configuration; the only difference is whether training waits for full episode completion (non-streaming)

Table 4: Training efficiency comparison on ALFWorld (1.5B, 4×GPU). Throughput is measured in generated tokens per second; time/step and phase breakdowns are wall-clock seconds averaged over the final 10 training steps.

Method	Tokens/sec	Time/step	Gen time	Update actor	Resp. len
Episodic PPO	2575	335.2	106.4	43.7	81.4
Streaming PPO	**4805**	**113.4**	**68.4**	**25.5**	60.5

or proceeds at chunk boundaries (streaming). Table 4 lists throughput, step time, and per-phase breakdown averaged over the final 10 training steps.

Streaming PPO achieves **87% higher throughput** than episodic PPO (4805 vs 2575 tokens/s) and **66% shorter step time** (113s vs 335s). The speedup comes from chunk-level execution: streaming proceeds to the next training chunk at the chunk boundary without waiting for full episode completion. Both generation time (-36%) and actor update time (-42%) are shorter under streaming, indicating that chunk-level batching better utilizes GPU compute even after normalizing for the 26% shorter average rollout length (60.5 vs 81.4 tokens).

6 Related Work

Scalable RL pipelines. To eliminate rollout idle time caused by heavy-tailed generation, scalable RL pipelines typically adopt asynchronous actor-learner architectures (Mnih et al., 2016; Espeholt et al., 2018; Hoffman et al., 2020; Sheng et al., 2024; Hu et al., 2024). Modern large-model RL infrastructures mitigate the policy staleness and generation skewness introduced by asynchronous execution through bounded delays, fine-grained packing, or adaptive trust region constraints (Fu et al., 2025; Hou et al., 2026; Gao et al., 2025; Li et al., 2027; Zhang et al., 2026; Jiang et al., 2026; Yang et al., 2026; Dirhoussi et al., 2026). Industrial agentic systems further extend this asynchronous paradigm to support long-horizon reasoning and tool-use learning (Kimi Team, 2025; LLM-Core, Xiaomi, 2025).

Critic-free vs critic-based RL. Recent LLM RL has shifted toward critic-free, outcome-only objectives that discard the learned value function and derive advantages directly from terminal outcomes (Li et al., 2024; Ahmadian et al., 2024; Shao et al., 2024; DeepSeek-AI et al., 2025a; Yu et al., 2025; Pu et al., 2026; Chen et al., 2025a). Such critic-free objectives, however, face a fundamental credit-assignment problem in long-horizon agents: the terminal outcomes must be attributed across a long sequence of intermediate actions, each of which may contribute differently to the final outcome. To address this problem, recent work constructs step-level or turn-level signals: one line learns explicit critics via TD-style value estimation, as in AgentPRM (Xi et al., 2025), ArCHer (Zhou et al., 2024), and Turn-PPO (Li et al., 2026b); the other constructs critic-free step-level credits via tree-based Monte Carlo aggregation (Wang et al., 2026), score-to-credit conversion (Liao et al., 2026), and related step-level schemes (Deng et al., 2024; Wang et al., 2025; Li et al., 2026a; Liu et al., 2026; Zhang, 2026; Li et al., 2026c). Yet learning critics in agentic tasks remains difficult: natural-language observations introduce high semantic variance across surface-similar states, which amplifies bootstrap error and destabilizes value estimation.

7 Conclusion

In this work, we studied anytime value learning in streamingpipeline for long-horizon language agents, where chunk-boundary bootstrapping destabilizes value magnitude within trajectories and value ordering across concurrent trajectories. Leveraging reasoning blocks as stable belief-state references, A2VL introduces longitudinal magnitude anchoring and transverse ordering anchoring, layered on periodic Monte Carlo calibration. Across three benchmarks and three model scales, A2VL suppresses value inversion and improves task performance while nearly doubling training throughput relative to episode-synchronous pipeline. Currently, A2VL relies on explicit reasoning traces and per-task tuned anchor coefficients. Extending belief-state read-outs to agents without explicit reasoning blocks and learning adaptive schedules for anchor coefficients remain key directions for future work.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, manuscript restructuring, and LaTeX formatting. The authors reviewed the resulting text, equations, citations, and implementation descriptions, and remain responsible for verifying all technical claims, experimental results, and submitted artifacts. Generative AI tools were not used as autonomous authors or as a substitute for author approval of the final manuscript.

REPRODUCIBILITY STATEMENT

Section 4 specifies the objectives and their scopes, while Appendices A, B and D provide pseudocode, implementation details, hyperparameters, and metric definitions.

REFERENCES

Arash Ahmadian, Maximilian Mosbach, Uzair Akhtar, Sebastian Hofstätter, Deniz Yuret, and Christian Fichtel. Back to basics: Revisiting REINFORCE style optimization for learning from human feedback in LLMs. In *Annual Meeting of the Association for Computational Linguistics*, 2024. arXiv:2402.14740.

Daniel Balcells, Andrew Jun Lee, Chirag Rastogi, Paul M. Riechers, Adam Shai, and Xavier Poncini. Large language models develop belief state geometry in-context. *arXiv preprint arXiv:2609.17376*, 2026.

Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning*, pp. 89–96, 2005.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: From pairwise approach to listwise approach. In *Proceedings of the 24th International Conference on Machine Learning*, pp. 129–136, 2007.

Kevin Chen, Marco Cusumano-Towner, Brody Huval, Aleksei Petrenko, Jackson Hamburger, Vladlen Koltun, and Philipp Krähenbühl. Reinforcement learning for long-horizon interactive LLM agents. *arXiv preprint arXiv:2502.01600*, 2025a.

Wentse Chen, Jiayu Chen, Hao Zhu, and Jeff Schneider. Context-lite multi-turn reinforcement learning for LLM agents. In *ES-FoMo III: 3rd Workshop on Efficient Systems for Foundation Models*, 2025b. URL https://openreview.net/forum?id=6CE5PLsZdW.

Mingyue Cheng, Shuo Yu, Daoyu Wang, Qingchuan Li, Xiaoyu Tao, Jie Ouyang, Yucong Luo, Yiju Zhang, Qi Liu, and Enhong Chen. Agent-R1: A unified and modular framework for agentic reinforcement learning. *arXiv preprint arXiv:2511.14460*, 2025.

Andrew Choi and Wei Xu. RankQ: Offline-to-online reinforcement learning via self-supervised action ranking. *arXiv preprint arXiv:2605.11151*, 2026.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025a.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025b.

Zhirui Deng, Zhicheng Dou, Yutao Zhu, Ji-Rong Wen, Ruibin Xiong, Mang Wang, and Weipeng Chen. From novice to expert: LLM agent policy optimization via step-wise reinforcement learning. *arXiv preprint arXiv:2411.03817*, 2024.

Amine Dirhoussi, Quentin Gallouédec, Kashif Rasul, Edward Beeching, Lewis Tunstall, Albert Villanova, Mehdi Zaki, and Cameron Wolfe. Keep the tokens flowing: Lessons from 16 open-source RL libraries. Hugging Face Blog, 2026. `https://huggingface.co/blog/async-rl-training-landscape`.

Lasse Espeholt, Volodymyr Mnih, Julian Marinier, Daniel Russo, Julian Schrittwieser, and Thore Graepel. IMPALA: Scalable distributed deep-RL with importance weighted actor-learner architectures. In *International Conference on Machine Learning*, pp. 1407–1416, 2018.

Wei Fu, Jiaxuan Gao, Xujie Shen, Chen Zhu, Zhiyu Mei, Chuyi He, Shusheng Xu, Guo Wei, Jun Mei, Jiashu Wang, Tongkai Yang, Binhang Yuan, and Yi Wu. AReaL: A large-scale asynchronous reinforcement learning system for language reasoning. In *Advances in Neural Information Processing Systems*, 2025.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, pp. 1587–1596, 2018.

Wei Gao, Yuheng Zhao, Dakai An, Tianyuan Wu, Lunxi Cao, Shaopan Xiong, Ju Huang, and Weixun Wang. RollPacker: Taming long-tail rollouts for RL post-training with tail batching. *arXiv preprint arXiv:2509.21009*, 2025. Published at NSDI 2026.

Danijar Hafner. Benchmarking the spectrum of agent capabilities. In *International Conference on Learning Representations*, 2022.

Matt Hoffman, Bogdan Shahriari, John Aslanides, Sergey Bartunov, Fereshteh Sadeghi, Nan Wang, Raia Sheahans, and Martin Riedmiller. SEED RL: Scalable and efficient deep-RL with accelerated central inference. In *International Conference on Learning Representations*, 2020.

Zhenyu Hou, Yujiang Li, Jie Tang, and Yuxiao Dong. Single-rollout asynchronous optimization for agentic reinforcement learning. *arXiv preprint arXiv:2607.07508*, 2026.

Jian Hu, Xibin Wu, Wei Shen, Jason Klein Liu, Zilin Zhu, Weixun Wang, Songlin Jiang, Haoran Wang, Hao Chen, Bin Chen, Weikai Fang, Xianyu, Yu Cao, Haotian Xu, and Yiming Liu. OpenRLHF: An easy-to-use, scalable and high-performance RLHF framework. *arXiv preprint arXiv:2405.11143*, 2024.

Zhida Jiang, Zhaolong Xing, Jiawei Lu, Yipei Niu, Qingyuan Sang, Liangxu Zhang, Wenquan Dai, Junhua Shu, Jiaxing Wang, Qiangyu Pei, Qiong Chen, Xinyu Liu, Fangming Liu, Ai Han, Zhen Chen, and Ke Zhang. Rollout-training co-design for efficient LLM-based multi-agent reinforcement learning. *arXiv preprint arXiv:2602.09578*, 2026.

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1–2):99–134, 1998.

Kimi Team. Kimi k1.5: Scaling reinforcement learning with LLMs. *arXiv preprint arXiv:2501.12599*, 2025.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1179–1191, 2020.

Haoyang Li, Sheng Lin, Fangcheng Fu, Yuming Zhou, Xiaodong Ji, Yanfeng Zhao, Lefeng Wang, Jie Jiang, and Bin Cui. StaleFlow: Staleness-aware data management for mitigating data skewness in fully disaggregated RL post-training. In *Proceedings of the 2027 International Conference on Management of Data*, 2027.

Jiazheng Li, Yawei Wang, Qiaojing Yan, Yijun Tian, Zhichao Xu, Huan Song, Panpan Xu, and Lin Lee Cheong. SALT: Step-level advantage assignment for long-horizon agents via trajectory graph. In *Findings of the European Chapter of the Association for Computational Linguistics*, 2026a. arXiv:2510.20022.

Junbo Li, Peng Zhou, Rui Meng, Meet P. Vadera, Lihong Li, and Yang Li. Turn-ppo: Turn-level advantage estimation with PPO for improved multi-turn RL in agentic LLMs. In *Findings of the Association for Computational Linguistics: EACL 2026*, 2026b. arXiv:2512.17008.

Yugu Li, Zehong Cao, Jianglin Qiao, and Siyi Hu. SSVPO: Effective step-level credit assignment for RL training of LLMs. In *International Conference on Learning Representations*, 2026c.

Ziniu Li, Tian Xu, Yushun Zhang, Zhihang Lin, Yang Yu, and Zhimin Luo. ReMax: A simple, effective, and efficient reinforcement learning method for aligning large language models. In *International Conference on Machine Learning*, 2024. arXiv:2310.10505.

Sicong Liao, Zhi Chen, and Yaohua Tang. TCPO: Turn-level credit policy optimization. *arXiv preprint arXiv:2608.01667*, 2026.

Xiaoqian Liu, Ke Wang, Yuchuan Wu, Fei Huang, Yongbin Li, Junge Zhang, and Jianbin Jiao. Agentic reinforcement learning with implicit step rewards. In *International Conference on Learning Representations*, 2026. arXiv:2509.19199.

LLM-Core, Xiaomi. MiMo: Unlocking the reasoning potential of language models through free-form reinforcement learning. *arXiv preprint arXiv:2505.07608*, 2025.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Askel K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pp. 1928–1937, 2016.

Andrew Y. Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward shaping: A general theory and methodology for accelerating reinforcement learning. In *Proceedings of the 16th International Conference on Machine Learning*, pp. 270–278, 1999.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Curtis Kelber, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022.

Liming Pu, Xiaoxia Li, Yifu Liu, Teng Cao, and Bin Yang. Explore more, drift less: Outcome-only reinforcement learning can suffice for long-horizon interactive agents. *arXiv preprint arXiv:2609.01245*, 2026.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, 2023.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations*, 2016.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, 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. *arXiv preprint arXiv:2409.19256*, 2024.

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

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2nd edition, 2018.

Hado van Hasselt. Double Q-learning. In *Advances in Neural Information Processing Systems*, volume 23, 2010.

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*, 2025.

Tao Wang, Suhang Zheng, and Xiaoxiao Xu. RTMC: Step-level credit assignment via rollout trees. *arXiv preprint arXiv:2604.11037*, 2026.

Zhiheng Xi, Chenyang Liao, Guanyu Li, Yajie Yang, Wenxiang Chen, Zhihao Zhang, Binghai Wang, Senjie Jin, Yuhao Zhou, Jian Guan, Wei Wu, Tao Ji, Tao Gui, Qi Zhang, and Xuanjing Huang. AgentPRM: Process reward models for LLM agents via step-wise promise and progress. *arXiv preprint arXiv:2511.08325*, 2025.

Junyao Yang, Yucheng Shi, Zongxia Li, Zhongzhi Li, Ruhan Wang, Xiangxin Zhou, Kishan Panaganti, Haitao Mi, and Leowei Liang. Stale but stable: Staleness-adaptive trust regions for stabilizing asynchronous reinforcement learning. *arXiv preprint arXiv:2607.18722*, 2026.

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.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Weixu Wang, et al. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

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

Liujie Zhang, Benzhe Ning, Rui Yang, Xiaoyan Yu, Jiaxing Li, Lumeng Wu, Jia Liu, Minghao Li, Weihang Chen, Weiqi Hu, and Lei Zhang. Relax: An asynchronous reinforcement learning engine for omni-modal post-training at scale. *arXiv preprint arXiv:2604.11554*, 2026.

Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. ArCHer: Training language model agents via hierarchical multi-turn RL. *arXiv preprint arXiv:2402.19446*, 2024.

A Implementation-Aligned Algorithm

Algorithm 1 A2VL anchor construction and critic update

Require: trajectory chains $\mathcal{C}$, values V, rewards r, response masks, think masks, `chain_done`
Require: longitudinal-anchor coefficient α, ordering-anchor weight λ_{ord}, temperature τ

for each chain $c \in \mathcal{C}$ **do**
  initialize terminal or truncated-tail bootstrap
  for each (i, k) in the reversed chain c **do**
    run the token-level GAE backward scan for row k
    $B \leftarrow V_k^o$
    if $i = 0$ and not `chain_done`$[k]$ and row k has think tokens **then**
      $B \leftarrow \alpha V_k^o + (1 - \alpha) V_k^z$ ▷ longitudinal anchor
    end if
    apply the eligible MC outcome anchor to later cross-row bootstraps
  end for
end for
construct returns from the resulting advantages and values
for each critic mini-batch $\mathcal{B}$ **do**
  $\mathcal{I} \leftarrow$ last row per trajectory identifier in $\mathcal{B}$
  remove rows in $\mathcal{I}$ whose trajectories are terminal
  randomly form $\lfloor |\mathcal{I}|/2 \rfloor$ disjoint pairs $\mathcal{P}$
  for $(i, j) \in \mathcal{P}$ **do**
    $w_{ij} \leftarrow \tanh(\text{sg}(V_i^z - V_j^z)/\tau)$
    $\ell_{ij} \leftarrow \text{softplus}(-w_{ij}(V_i^o - V_j^o))$
  end for
  update critic with $\mathcal{L}_{\text{value}} + \lambda_{\text{ord}} \, \text{mean}_{\mathcal{P}} \, \ell_{ij}$
end for

B Experimental Details

B.1 Configuration table

Within the chained-GAE scan, reasoning tokens receive unit discount and action tokens receive the configured token-level discount. We set $\gamma = 1$ in the chunked experiments to avoid compounding discount across long token sequences. Ranking statistics are computed and logged for each critic mini-batch rather than after aggregation over the full training step.

B.2 Prompt templates

We adopt a uniform prompt structure across all three environments: each step presents the current observation and admissible actions, and asks the model to first reason within `<think>` tags and then output an action within `<action>` tags. The system prompt is folded into the first user message (`remove_system_prompt=True`). Below we show the per-step template for each environment.

ALFWorld. At each step the agent receives the current text observation and the list of admissible actions:

```
You are an expert agent operating in the ALFRED Embodied
Environment. Your task is to: {task}. Prior to this
step, you have already taken {step_count} step(s). Below
are the most recent {history_length} observations and the
corresponding actions you took: {action_history} You are
now at step {current_step} and your current observation is:
{current_observation} Your admissible actions of the current
situation are: [{admissible_actions}]. Now it's your turn
to take an action. You should first reason step-by-step
about the current situation. This reasoning process MUST be
enclosed within <think> </think> tags. Once you've finished
```

Table 5: Common hyperparameters for baseline streaming chunk-1 training across ALFWorld, WebShop, and Crafter. Environment-specific differences in prompt length, response budget, and buffer size are noted in the text.

Parameter	Value
Backbone	Qwen2.5-1.5B-Instruct
Actor learning rate	1×10^{-6}
Critic learning rate	1×10^{-5}
Actor PPO mini-batch	32
Actor micro-batch per GPU	16
Critic mini-batch	16
Critic micro-batch per GPU	16
Rollout samples per prompt (n)	8
KL loss coefficient	0.01
KL loss type	low-var
Entropy coefficient	0.001
PPO clip range (ϵ)	0.2
Token discount γ	1.0
GAE λ	1.0
Streaming segment steps	1
Max staleness	0
Rollout mode	async (streaming)
vLLM GPU memory utilization	0.6
Sampling temperature (train / val)	1.0 / 0.4
Gradient checkpointing	enabled
Optimizer offload	disabled
Ref model param offload	enabled
Total epochs	150
Test frequency	every 5 steps
Val batch size	128
Seeds	≥ 3

```
your reasoning, you should choose an admissible action for
current step and present it within <action> </action> tags.
```

WebShop. The agent receives a structured observation (web page content) and a dynamic action list (search, click, purchase, etc.):

```
You are an expert autonomous agent operating in the WebShop
e-commerce environment. Your task is to: {task}. Prior
to this step, you have already taken {step_count} step(s).
Below are the most recent {history_length} observations and
the corresponding actions you took: {action_history} You
are now at step {current_step} and your current observation
is: {current_observation}. Your admissible actions of the
current situation are: [{available_actions}]. Now it's your
turn to take one action for the current step. You should
first reason step-by-step about the current situation, then
think carefully which admissible action best advances the
shopping goal. This reasoning process MUST be enclosed
within <think> </think> tags. Once you've finished your
reasoning, you should choose an admissible action for
current step and present it within <action> </action> tags.
```

Crafter. The agent receives a textual observation of the current grid, inventory, vitals, and nearby objects. A system-level instruction (folded into the first user message) describes the 22 achievements and key object interactions. The per-step template appends the action list and the think-then-act instruction:

```
 {task_and_achievement_description}. Prior to this step,
you have already taken {step_count} step(s). Below are
the most recent {history_length} observations and the
```

```
corresponding actions you took: {action_history} You are
now at step {current_step} and your current observation is:
{current_observation}

Your available actions (output exactly one within <action>
tags): [{available_actions}]. Consider your vitals,
inventory, nearby objects, and achievements progress, then
decide your next action. The reasoning process MUST be
enclosed within <think> </think> tags. Output only one
action within <action> </action> tags.
```

B.3 Component ablation

The component ablation in Figure 3 isolates each element of the A2VL hierarchy on top of the streaming ChainedGAE substrate: (i) the MC-calibrated base setting with periodic outcome anchoring; (ii) the same base setting with MC calibration removed; (iii) the base setting augmented with the longitudinal magnitude anchor alone; (iv) with the transverse ordering anchor alone; and (v) the complete A2VL hierarchy combining both anchors. This design separates the contribution of the streaming backbone from the two belief-state anchors and identifies which bootstrap failure mode each anchor addresses. Unless otherwise noted, all runs use the default coefficients $\alpha = 0.5$, $\lambda_{\text{ord}} = 0.5$, and $\tau = 0.5$, matching Table 5.

B.4 Streaming Framework Ablations on ALFWorld

We conduct two ablation studies on ALFWorld to characterize the streaming training framework itself.

Fetch-size ablation. The core variable is *fetch size*: the number of trajectory chunks consumed per optimizer step. All runs use chunk-1 (one environment step per chunk), so fetch size directly controls the number of steps per training batch. Figure 4 shows that larger fetch sizes converge faster in wall-clock steps: fetch=5120 (the default) reaches near-ceiling accuracy within $\sim$150 steps, while fetch=512 requires $\sim$1000 steps, confirming that fetch size is a throughput–stability knob rather than a sample-efficiency lever.

Model-size ablation. Figure 5 compares streaming training at 1.5B and 7B model scale on ALFWorld. The 1.5B model converges smoothly to $\sim$1.0 accuracy within $\sim$100 steps. The 7B model exhibits two instability episodes (around steps 40 and 80) with sharp accuracy drops and eventual recovery, suggesting that larger models are more susceptible to critic instability under streaming updates—consistent with the higher variance of value estimation at larger capacity motivating the A2VL anchor mechanisms.

C Representation Geometry Across Training

Figure 6 extends the base-model analysis of Figure 2 from a single snapshot to four checkpoints spanning the full training run (steps 0, 50, 100, and 150), using 128 rollout states per checkpoint. Each row corresponds to one checkpoint and shows two principal component analysis (PCA) projections of the critic's hidden states: the left panel reads the value at the observation position (V^o) and the right panel reads it at the belief-state position (V^z). Within a panel, each colored group is one task, the small dots are its surface-preserving variants, the cross marks the task centroid, and the ellipse traces the within-task spread. All eight panels share identical PC1/PC2 axis limits, so both the difference between the two columns and its evolution across rows can be read off the same scale without per-panel rescaling.

The figure conveys a single, stable message: the value-resolution gap between the observation and belief-state positions is present at *every* checkpoint, not only at initialization. In the left column the task centroids stay collapsed into one heavily overlapping cluster throughout training, so distinct decision states remain hard to separate by V^o; in the right column the same tasks fan out across the projection, so V^z resolves them cleanly. Because this separation gap holds from the untrained step-0 policy through the converged step-150 policy, the belief-state position is a dependable anchoring

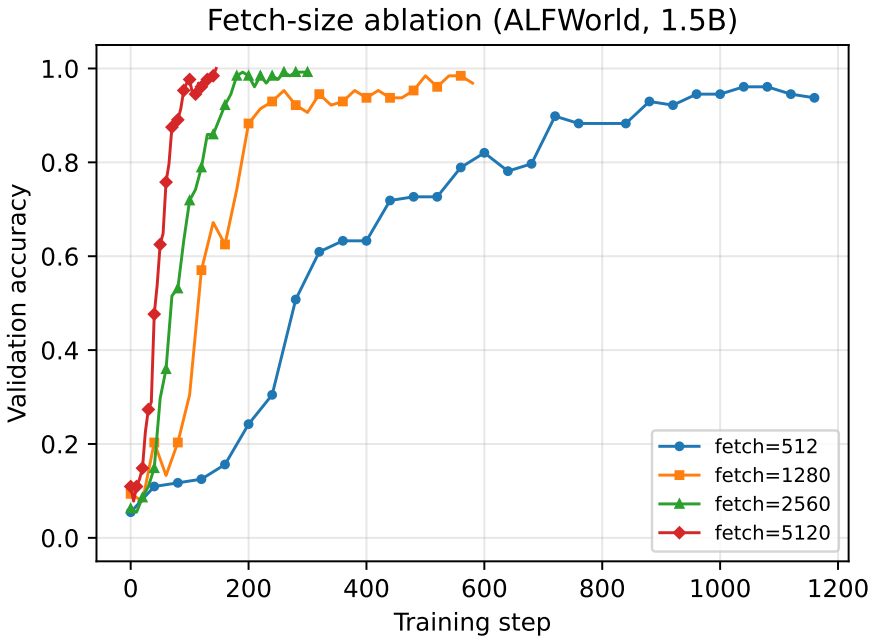


Figure 4: **Fetch-size ablation on ALFWorld (1.5B, 4-GPU).** Validation accuracy vs. training step for different fetch sizes (number of chunks per optimizer step). All runs use chunk-1 granularity. Larger fetch sizes converge faster in step count.

reference for the entire run rather than a property of a particular training stage—supporting its use as the detached reference in both A2VL anchors (Section 4).

D Metric Definitions

For terminal trajectories with outcome label $y \in \{0, 1\}$, we define the observation-position inversion gap as

$$\Delta^o_{\text{inv}} = \mathbb{E}[V^o \mid y = 0] - \mathbb{E}[V^o \mid y = 1], \tag{10}$$

with Δ^z_{inv} defined analogously. A positive gap means that failures have the higher group mean. We also estimate the pairwise inversion probability

$$\rho^o_{\text{inv}} = \Pr(V^o_i > V^o_j \mid y_i = 0, y_j = 1). \tag{11}$$

We measure calibration using mean absolute error and mean squared error against realized returns, stratified by distance to termination. To detect scale drift, we report the mean, standard deviation, and selected quantiles of V^o and V^z. Ranking diagnostics include the number of eligible trajectories, the resulting number of pairs, the fraction removed by terminal filtering, and the mean confidence $\mathbb{E}[|w_{ij}|]$.

System metrics include generated tokens per second, optimizer utilization, environment wait time, wall-clock time to a fixed interaction budget, and time to a fixed task score. Learning curves report individual seeds or uncertainty bands rather than only the best run.

E Additional Limitations

Because the two value positions share most critic parameters, stopping the direct gradient through V^z does not make their future predictions independent: ordinary value regression still updates their shared representation. The guarantee is therefore local to the ordering-anchor loss and its current forward pass. In addition, filtering terminal trajectories focuses this anchor on ongoing behavior but may leave too few pairs in a mini-batch. We consequently report effective pair counts alongside the loss.

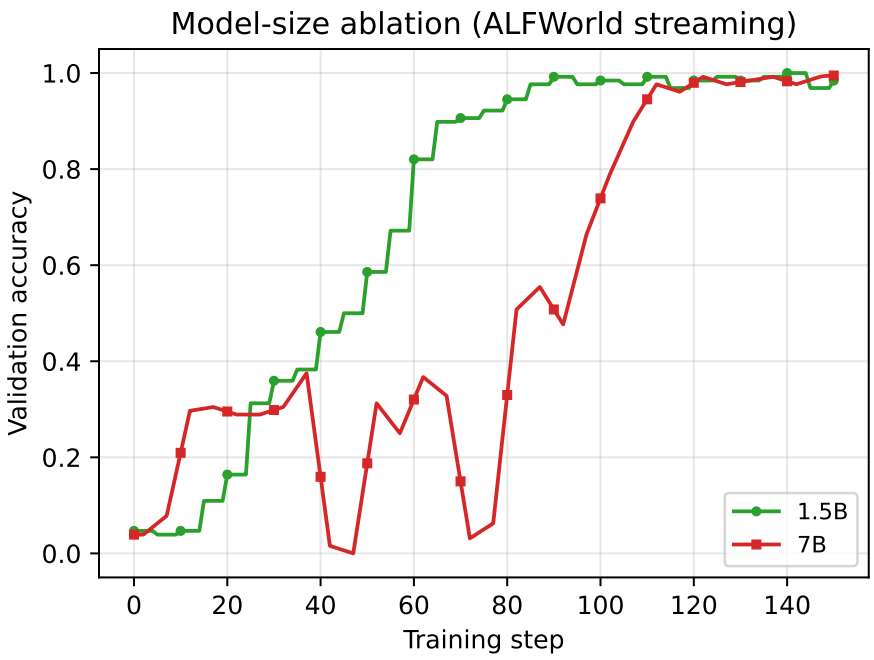


Figure 5: **Model-size ablation on ALFWorld streaming.** Validation accuracy vs. training step for 1.5B (stable) and 7B (smoothed; two instability episodes). Both runs use the same streaming framework; the 7B model shows greater critic instability.

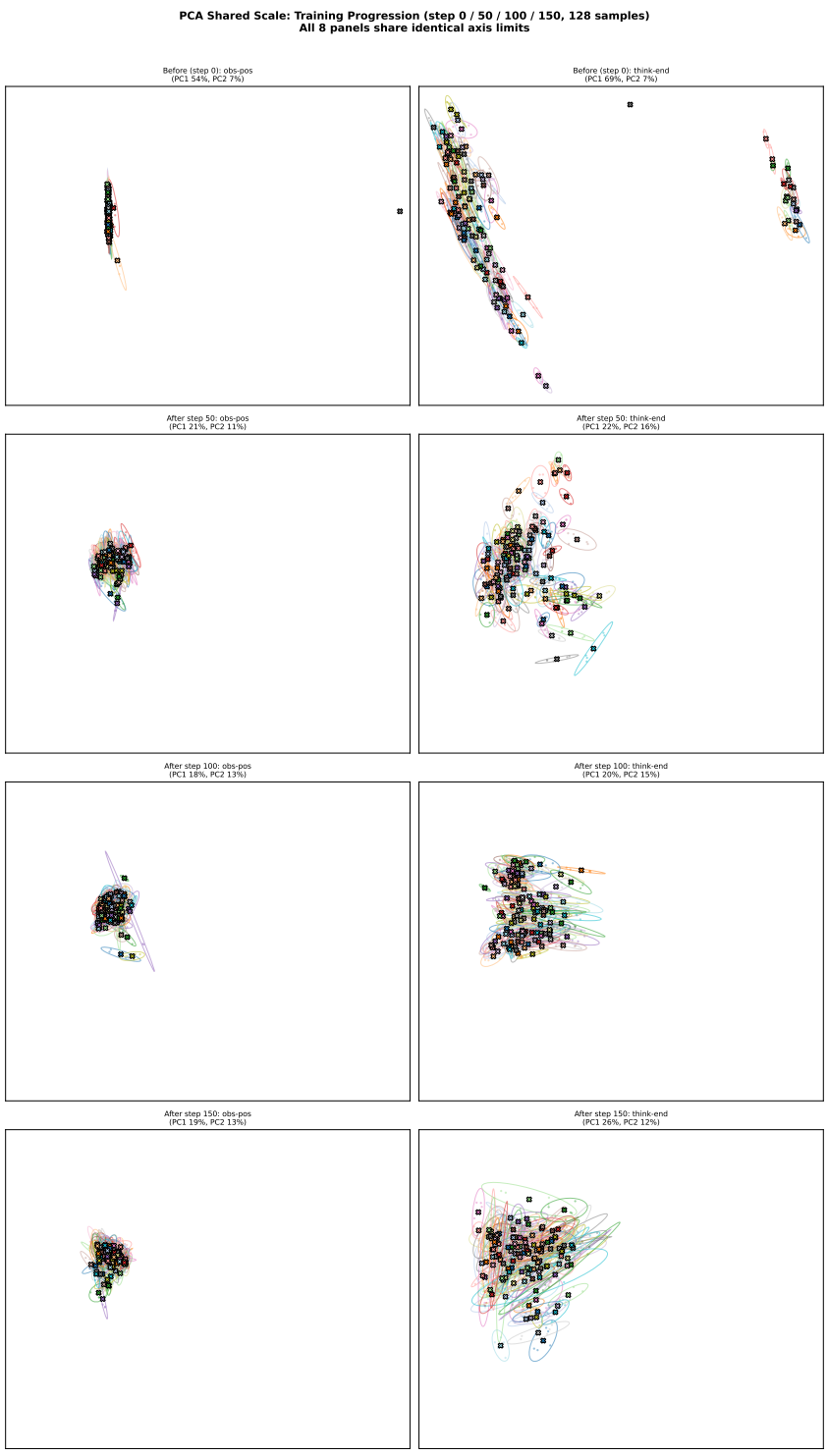


Figure 6: **Representation geometry across training checkpoints (128 samples, shared axis scale).** Rows are training checkpoints (steps 0, 50, 100, 150); the left column reads the observation-position value V^o and the right column the belief-state value V^z. Dots are surface-preserving variants of a task, crosses are task centroids, and ellipses show within-task spread. All eight panels share identical PC1/PC2 limits, so the two columns are directly comparable: at every checkpoint the observation-position centroids stay collapsed into one overlapping cluster while the belief-state centroids separate across the projection, showing that the V^o–V^z resolution gap persists throughout training rather than being an initialization artifact.