

Extended Abstract

Hybrid Advantage Shaping with Goal-Aware Attention for Per-Turn Credit Assignment in
LLM-Agent Reinforcement Learning
Joseph Li, Max Rodriguez, Samantha Leventis

Motivation. LLM agents acting over many turns in sparse-reward environments such as AlfWorld and WebShop receive only a single *terminal* reward, yet success depends on credit being assigned to the individual turns most responsible for the outcome. Although flatGRPO is stable, it does not distinguish among turns because it assigns the same group-normalized trajectory advantage to every turn. Recent step-structured variants such as GiGPO and HoGPO recognize this limitation, but still assign credit through fixed grouping rules or heuristic step structure rather than a learned, task-conditioned decomposition. Classical return decomposition addresses delayed reward, but not the language-agent setting where the same local action can be useful or harmful depending on the goal. The missing mechanism is a GRPO-compatible way to retain the reliable trajectory-level baseline while injecting learned, goal-aware per-turn credit.

Method. We address this gap with **Hybrid Advantage Shaping (HAS)**, a one-parameter framework whose per-turn advantage is a convex blend $A_H^t = \alpha A_{\text{traj}}^t + (1 - \alpha) A_{\text{turn}}^t$ of the group-normalized trajectory advantage and a per-turn signal produced by a *pluggable* decomposer. At $\alpha=1$ HAS recovers flatGRPO exactly, so the framework contains the standard baseline as a special case; $\alpha=0.5$ performs best in our sweeps. Our strongest decomposer is **TurnRD with goal-aware FiLM attention**: a compact bidirectional transformer over per-turn encoder hidden states whose attention and value heads are modulated by Feature-wise Linear Modulation (FiLM) parameters predicted from a per-trajectory goal embedding.

Implementation. The policy is Qwen2.5-1.5B-Instruct with a rank-32 LoRA adapter (attention + MLP targets), trained on Modal A100-80GB GPUs. Training alternates between two decoupled stages. In the policy stage, the current LoRA policy collects K rollouts for each task; trajectory advantages are computed from group-normalized terminal rewards, while a frozen TurnRD checkpoint predicts per-turn credits used in the HAS blend for the PPO/GRPO update. In the credit-model stage, after a round of policy rollouts has been written to replay, the TurnRD checkpoint is updated in a separate training job using the cumulative replay buffer with recency decay, then loaded by the next policy-training round. This separation keeps rollout and policy optimization lightweight while letting the credit model improve from accumulated experience without being updated online inside every policy-gradient step.

Results. On **WebShop**, HAS with TurnRD reaches **91.3%** success versus 80.5% for flatGRPO (+10.8pp), and beats the Progressive (+5.3pp) and LLM-Judge (+9.3pp) decomposers. On **AlfWorld**, TurnRD+FiLM reaches **80%** versus $\sim 70\%$ for flatGRPO (+10pp), with goal-aware FiLM contributing +5pp over a no-FiLM variant (75%).

Discussion. The results support HAS as a practical credit-assignment framework in our evaluated settings: the same α -blend improves over flatGRPO on both AlfWorld and WebShop, and TurnRD is the strongest decomposer among the learned, heuristic, and judge-based alternatives we tested. The central pattern is that trajectory-level GRPO supplies stable group-relative ranking, while a learned intra-trajectory signal separates decisive turns from incidental ones. Goal-aware FiLM further lets TurnRD interpret actions relative to the task instruction, which helps distinguish locally plausible steps from goal-relevant progress.

Conclusion. Hybrid Advantage Shaping with TurnRD keeps GRPO’s stable trajectory-level objective while adding learned per-turn credit where flatGRPO lacks intra-trajectory structure. The framework improves performance across two agentic benchmarks, preserves flatGRPO as the $\alpha=1$ fallback, and suggests that attention-based, goal-aware turn-level credit is a useful component for GRPO-style optimization of multi-turn LLM agents.

Hybrid Advantage Shaping with Goal-Aware Attention for Per-Turn Credit Assignment in LLM-Agent Reinforcement Learning

Joseph Li
Stanford University
shoupei@stanford.edu

Max Rodriguez
Stanford University
maxrod@stanford.edu

Samantha Leventis
Stanford University
samanthaleventis@stanford.edu

Abstract

Sparse terminal rewards make reinforcement learning for multi-turn LLM agents a central credit-assignment problem. Success is observed only at the trajectory level, while learning requires knowing which turns caused the outcome. GRPO-style training uses group-relative comparisons among rollouts of the same task, but flatGRPO assigns the same trajectory advantage to every turn and is therefore blind to intra-trajectory structure. Recent step-structured variants add finer-grained credit heuristics, but do not learn task-conditioned turn-level credit for language-agent trajectories. We propose *Hybrid Advantage Shaping* (HAS), a GRPO-compatible framework that preserves the stable trajectory-level baseline while adding learned per-turn credit through a pluggable decomposer. A single coefficient α interpolates between the trajectory advantage and the decomposed turn-level signal, recovering flatGRPO exactly at $\alpha=1$. We instantiate the decomposer with *TurnRD*, a goal-aware attention model that analyzes completed trajectories and assigns task-dependent credit to individual turns. On WebShop, HAS with TurnRD reaches 91.3% success, improving over flatGRPO by +10.8pp and over Progressive and LLM-Judge decomposers by +5.3pp and +9.3pp. On AlfWorld, TurnRD+FiLM reaches 80% success, +10pp over flatGRPO and +5pp over a goal-blind TurnRD variant. These results suggest that learned, goal-aware turn-level credit can improve stable GRPO-style optimization of sparse-reward LLM agents.

1 Introduction

Large language models are increasingly deployed as *agents* that act over long horizons: issuing tool calls, navigating web pages, and manipulating embodied environments through sequences of natural-language actions [Yao et al., 2023]. Reinforcement learning (RL) is attractive for these agents because it optimizes task success directly rather than a proxy such as next-token likelihood. Yet realistic agentic environments are often sparse: the environment returns a single scalar reward only at the end of an episode, even though the episode contains many turns. The learner must therefore solve a credit-assignment problem and determine which turns caused success or failure, and which were incidental.

The prevailing recipe for LLM-agent RL is Group Relative Policy Optimization (GRPO) [Shao et al., 2024, Guo et al., 2025], which avoids a learned value critic by z -scoring terminal rewards across a group of K rollouts of the same task. This makes GRPO simple and stable, but its standard flat form assigns the same trajectory-level advantage to every turn of the trajectory. As a result, every turn in a successful rollout receives positive credit, including redundant detours or actions that merely happened to precede success.

Recent work has begun to address this failure mode by adding step-level or hierarchical structure to policy optimization, or by training process reward models that provide denser supervision. These directions confirm that flat trajectory credit is insufficient, but they leave two gaps for sparse-reward LLM agents: many rely on fixed grouping or externally supplied process labels, and they do not provide a simple way to combine learned turn-level credit with GRPO’s stable trajectory-level baseline. In language-agent tasks, this distinction matters because the same local action can be decisive or irrelevant depending on the task instruction.

We therefore argue that the missing piece is not merely denser feedback, but a controlled, task-conditioned mechanism for inserting learned per-turn credit into GRPO without discarding the reliable group-relative trajectory signal. We make two contributions.

(1) Hybrid Advantage Shaping (HAS), a framework whose per-turn advantage is a convex blend of the trajectory advantage and a pluggable per-turn decomposer, controlled by a single coefficient $\alpha \in [0, 1]$. At $\alpha=1$ HAS is *exactly* flatGRPO, so it retains the baseline objective as an available fallback; intermediate α injects dense per-turn credit, with $\alpha=0.5$ performing best in our sweeps.

(2) TurnRD with goal-aware FiLM attention, the strongest realization of the decomposer slot: a compact bidirectional transformer over per-turn encoder hidden states that predicts per-turn credit weights and values, modulated by FiLM [Perez et al., 2018] parameters derived from a per-trajectory goal embedding. TurnRD is trained as a side network on a recency-decayed replay buffer with an anchored reward-prediction objective; in the main runs, per-turn credits are obtained by projecting value predictions onto the constraint that their sum equals the observed terminal return.

Within a single shared framework, HAS achieves **91.3%** success on WebShop (+10.8pp over flatGRPO) and **80%** on AlfWorld (+10pp), with goal-aware FiLM contributing +5pp on AlfWorld; learned attention-based credit improves over heuristic decomposers by +5.3 to +9.3pp on WebShop. Beyond aggregate performance, we provide mechanism evidence that FiLM uses goal information. The remainder of the paper covers related work (§2), the method and Algorithm 1 (§3), setup (§4), results (§5), discussion (§6), and conclusions (§7).

2 Related Work

GRPO and critic-free LLM policy optimization. RLHF popularized PPO [Schulman et al., 2017] for aligning language models [Ouyang et al., 2022]. GRPO [Shao et al., 2024, Guo et al., 2025] removes the learned value critic by normalizing rewards within a group of sampled completions, making it attractive for LLM fine-tuning where value estimation is costly and unstable. Recent systems such as DAPO [Yu et al., 2025] and VAPO [Yue et al., 2025] improve GRPO-style optimization through better sampling, clipping, and stability mechanisms. These advances make group-relative RL practical at scale, but they mainly address how to optimize from trajectory rewards rather than how to allocate those rewards across turns. In flat GRPO, one trajectory-level advantage is still broadcast to every turn. HAS is complementary: it preserves the group-relative trajectory baseline while changing only the intra-trajectory credit signal passed to the policy update.

Credit assignment in long-horizon LLM agents. Long-horizon agent training makes temporal credit assignment unavoidable. ArCHer [Zhou et al., 2024] introduces hierarchical value estimates over agent turns, while RAGEN [Wang et al., 2025] analyzes how GRPO behaves as multi-turn rollouts grow longer. Recent step-structured variants such as GiGPO [Feng et al., 2025] and HoGPO [He et al., 2026] add step- or hierarchy-level normalization to GRPO, and StepPO-style methods [Wang et al., 2026] align updates to per-step signals. These methods confirm that flat trajectory credit is insufficient. However, their credit granularity is usually determined by fixed grouping rules, positional hierarchy, or externally supplied step rewards. HAS instead separates the optimization baseline from the credit granularity: a single mixing coefficient controls how much learned, heuristic, or judge-based per-turn signal enters the update, while flatGRPO remains the $\alpha=1$ special case.

Return decomposition and process reward models. Classical return decomposition such as RUDDER [Arjona-Medina et al., 2019] redistributes delayed rewards across timesteps, and hindsight credit assignment [Harutyunyan et al., 2019, Andrychowicz et al., 2017] reweights experience by outcome. Process reward models provide denser supervision for reasoning and agentic tasks [Wu et al., 2023, Khalifa et al., 2025, Chen et al., 2025], and recent AgentPRM/HiPER-style approaches learn step-wise progress or explicit credit models for LLM agents [Xi et al., 2025, Peng et al.,

2026]. These methods are closely related in spirit to TurnRD because they replace sparse terminal feedback with more localized learning signals. The main difference is the supervision interface: many process-reward approaches rely on explicit process labels, external reward-model training, or a separate annotation pipeline. TurnRD learns from the agent’s own replay and terminal outcomes, then plugs into a GRPO-style update as a post-hoc per-turn decomposer.

Goal-aware credit for language agents. In language-agent environments such as AlfWorld [Shridhar et al., 2021] and WebShop [Yao et al., 2022], the same local action can be useful or harmful depending on the task instruction. ReAct [Yao et al., 2023] established prompting patterns for reasoning-and-acting agents, and trajectory optimization approaches such as ETO [Song et al., 2024] fine-tune agents from collected experience. Our focus is narrower: given collected trajectories and terminal outcomes, we ask how credit should be assigned within each trajectory. FiLM [Perez et al., 2018] provides a lightweight mechanism for conditioning intermediate representations by feature-wise affine modulation. TurnRD uses FiLM to condition per-turn credit on the trajectory goal, making the decomposer task-aware rather than position-only or goal-blind. The policy itself is updated with parameter-efficient LoRA [Hu et al., 2022].

3 Method

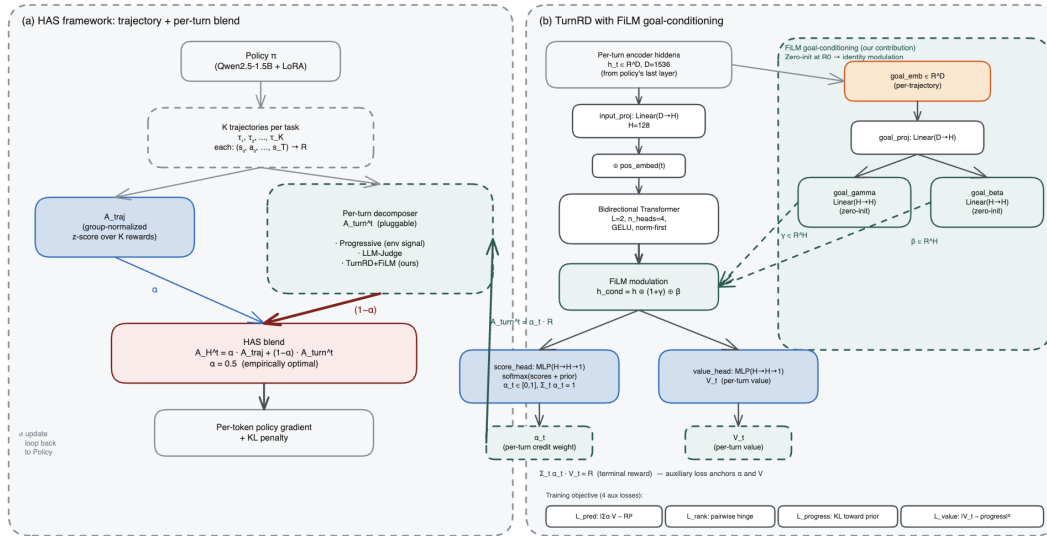


Figure 1: **Overview of our two contributions.** (a) The Hybrid Advantage Shaping (HAS) framework: a pluggable per-turn decomposer produces A_{turn}^t , blended with the group-normalized trajectory advantage A_{traj} through a single mixing coefficient α . At $\alpha=1$ we recover flatGRPO; $\alpha=0.5$ performs best in our sweeps. (b) TurnRD with FiLM goal-conditioning, our strongest decomposer. A bidirectional transformer reads per-turn encoder hidden; a per-trajectory goal embedding produces FiLM parameters γ, β that modulate the hidden before an attention head (weights w_t) and a V -head (per-turn values). The auxiliary reconstruction $\sum_t w_t V_t \approx R$ anchors the attention/value factorization, while the policy update uses a sum-preserving projection of V_t for per-turn credits.

3.1 Background: Group-relative policy optimization

For a task, the current policy π_θ generates a group of K trajectories; each trajectory τ_i yields a terminal reward R_i . GRPO forms a group-normalized trajectory advantage by z -scoring,

$$A_{traj}(\tau_i) = \frac{R_i - \text{mean}_j R_j}{\text{std}_j R_j + \varepsilon_0}, \quad (1)$$

and updates π_θ with a PPO clipped surrogate plus a KL penalty to a reference policy. In *flat* GRPO this scalar is assigned to every turn (and token) of τ_i ; no intra-trajectory credit is expressed.

3.2 Hybrid Advantage Shaping

HAS replaces the flat assignment with a per-turn advantage that blends the trajectory baseline with a per-turn signal A_{turn}^t from a decomposer:

$$A_H^t = \alpha A_{\text{traj}} + (1 - \alpha) A_{\text{turn}}^t, \quad \alpha \in [0, 1]. \quad (2)$$

The blended quantity is computed *per turn* and then broadcast over that turn’s tokens for the surrogate loss. Three properties make Eq. (2) attractive. First, it is a *one-knob* interpolation: α trades trajectory-level variance reduction against per-turn credit density. Second, it is *backward-compatible*: at $\alpha=1$ the per-turn term vanishes and HAS reduces exactly to flatGRPO, so deployments can recover the baseline objective without changing the rest of the training stack. Third, the decomposer is *pluggable*: we support a parameter-free Progressive decomposer (env shaping signal), an LLM-Judge decomposer, and the learned TurnRD decomposer below.

Why blend, and why $\alpha=0.5$. The blend is best read as a bias–variance interpolation over *where* credit is placed. The trajectory advantage A_{traj} is low-variance because it ranks whole trajectories within the group and averages over much of the per-turn stochasticity. However, it is high-bias at the turn level, assigning identical credit to every turn of a trajectory, including turns that were irrelevant or harmful. A pure per-turn signal A_{turn}^t has the opposite profile: it can localize credit to the turns that mattered, but it is produced by a learned decomposer and is therefore higher-variance, especially early in training before that decomposer has converged. Equation (2) interpolates between these regimes with a single scalar; our α -sweep (§5) finds that the midpoint $\alpha=0.5$ performs best in our setting. This suggests that neither extreme suffices alone: pure trajectory credit is credit-blind, while an unanchored per-turn estimate is too noisy.

Proposition 1 (flatGRPO containment) *At $\alpha=1$ the per-turn term in Eq. (2) is multiplied by zero, so $A_H^t = A_{\text{traj}}$ for every turn and the per-token PPO surrogate gradient is identical to that of flatGRPO. HAS therefore contains flatGRPO as the $\alpha=1$ special case and recovers it exactly when the per-turn signal is disabled.*

Proposition 1 is a containment statement, not a guarantee that every intermediate α dominates the baseline. Its practical value is operational: if a mis-specified or under-trained decomposer destabilizes learning, one can anneal $\alpha \rightarrow 1$ and recover the exact flatGRPO objective without modifying the trainer. We further keep the mixture *convex* ($\alpha \in [0, 1]$) with weights summing to one) rather than permitting free reweighting. Convexity preserves the scale and sign of the advantage, so the PPO clipping range and KL trust region behave comparably to the baseline, and it keeps the blended per-turn advantages centered consistently with the group baseline.

Policy objective. We now specify how the shaped advantage enters the policy update. For each task, the sampling policy $\pi_{\theta_{\text{old}}}$ generates K trajectories $\{\tau_i\}_{i=1}^K$. Each trajectory contains a sequence of turns, and each turn contributes one or more action tokens. Let $t(s)$ denote the turn containing token s . After the decomposer produces per-turn credits, we normalize them across the K rollouts at each valid turn position to obtain A_{turn}^t , then form the blended turn advantage

$$\hat{A}_{i,s} = A_H^{t(s)} = \alpha A_{\text{traj}}(\tau_i) + (1 - \alpha) A_{\text{turn}}^{t(s)}. \quad (3)$$

This turn-level advantage is broadcast to all action tokens in the corresponding turn. The policy is then trained with the usual clipped PPO surrogate and a KL penalty to a frozen reference policy π_{ref} :

$$\mathcal{L}_\pi(\theta) = -\mathbb{E}_{i,s} \left[\min(\rho_{i,s} \hat{A}_{i,s}, \text{clip}(\rho_{i,s}, 1-\epsilon, 1+\epsilon) \hat{A}_{i,s}) \right] + \beta_{\text{KL}} \text{KL}[\pi_\theta \parallel \pi_{\text{ref}}], \quad (4)$$

where $\rho_{i,s} = \pi_\theta(a_s \mid a_{<s}) / \pi_{\theta_{\text{old}}}(a_s \mid a_{<s})$ is the per-token importance ratio against the policy that generated the rollout. Thus HAS changes only the advantage supplied to PPO: the likelihood ratio, clipping rule, and reference-policy regularization are unchanged from flatGRPO. Because the blend is applied after group normalization and is not re-standardized afterward, the convex mixture preserves a comparable advantage scale. At $\alpha=1$, $\hat{A}_{i,s} = A_{\text{traj}}(\tau_i)$ for every token, recovering the flatGRPO policy gradient exactly (Prop. 1).

3.3 TurnRD: a goal-aware attention decomposer

TurnRD (Figure 1b) reads the policy’s per-turn last-hidden states $h_{1:T}$ ($D=1536$), projects them to a hidden width $H=128$, and encodes them with a bidirectional transformer (2 layers, 4 heads). Two heads sit on top: an attention head that applies a softmax over turns to produce normalized weights w_t , and a V -head that predicts per-turn scalar values V_t . The auxiliary prediction $\hat{R} = \sum_t w_t V_t$ anchors attention to return prediction rather than arbitrary saliency. For the policy update, however, our main implementation uses a value-projection credit signal: it shifts the active-turn values by a constant so their sum equals the terminal reward,

$$\hat{r}_t = \text{clip}(V_t, -c, c) - \frac{\sum_{s=1}^T \text{clip}(V_s, -c, c) - R}{T}. \tag{5}$$

Here $\text{clip}(\cdot, -c, c)$ bounds each value to $[-c, c]$ with $c=2$. The projected credits \hat{r}_t define A_{turn}^t after position-wise group normalization across the K rollouts. Variable-length trajectories are padded only for batched TurnRD computation; an attention mask excludes padded turns, and each turn position is normalized only over trajectories that actually reached that position. This projection preserves the observed return exactly while allowing negative blame and allowing one decisive turn to receive more credit than the final reward when compensated by negative turns.

Why bidirectional, non-causal attention. The encoder is deliberately *non-causal*: each turn attends to both earlier and later turns. This is essential because the credit due to an early turn typically depends on whether *later* turns exploit it. For example, picking up an object is valuable only if a subsequent turn uses it to satisfy the goal. A causal model, restricted to the past, cannot see the downstream turns that realize (or waste) an early action’s payoff, and would systematically under-credit setup actions. Non-causality is legitimate here because TurnRD is an *offline* post-hoc analyzer of completed trajectories used only to shape advantages; it never leaks future information into the acting policy.

Why a softmax attention head (selectivity). The attention head normalizes weights with a softmax over turns, so $\sum_t w_t = 1$. The model therefore allocates a fixed unit budget across the trajectory. This simplex constraint forces selectivity because raising one turn’s weight must lower another’s, rather than letting the model trivially flag every turn as important. used to anchor reward prediction and improve interpretability. The final per-turn shaping signal for the main experiments comes from the value projection in Eq. (5).

Anchoring and conservation. The reconstruction term $\sum_t w_t V_t \approx R$ anchors an otherwise under-determined attention/value factorization to the single observable quantity, the terminal reward, reducing the chance that the attention head drifts into a purely cosmetic saliency map. The value-projection step enforces exact conservation of return for the actual shaping credits: empirically the projected per-turn credits have maximum sum-to- R drift of only $\sim 2 \times 10^{-6}$ across probed trajectories, so the credit visualizations in Appendix A decompose reward rather than merely highlighting turns.

Goal-aware FiLM conditioning. A per-trajectory goal embedding is projected into FiLM parameters $\gamma, \beta \in \mathbb{R}^H$, which modulate each encoder hidden state before the attention and value heads:

$$\text{FiLM}(h_t) = \gamma \odot h_t + \beta. \tag{6}$$

The FiLM projector is zero-initialized, so the conditioned model starts as the unconditioned decomposer at round 0. As training proceeds, the model learns to use goal information to reshape per-turn credit.

Why credit must be goal-aware. In instruction-conditioned agent environments, the same local action can have different credit depending on the task instruction. For example, opening the fridge may be useful when the goal is to cool an object, but irrelevant when the goal is to place a book on a desk. A goal-blind decomposer may learn an average credit pattern over superficially similar trajectories. This can blur the turns that matter for a specific instruction. Conditioning on the goal lets TurnRD assign credit relative to the task being solved, rather than only relative to position or generic progress.

FiLM provides a lightweight way to inject this goal information. A single per-trajectory affine modulation conditions every turn representation with small parameter overhead and no dependence on

Algorithm 1 Hybrid Advantage Shaping (HAS) with a replay-trained TurnRD credit model

Require: policy π_θ (LoRA); rounds N , episodes/round M , rollouts/task K ; mix α ; recency half-life H

- 1: Replay buffer $\mathcal{B} \leftarrow \emptyset$; initialize TurnRD checkpoint ϕ_0
- 2: **for** round $r = 0, \dots, N - 1$ **do**
- 3: Load frozen TurnRD checkpoint ϕ_r for policy shaping
- 4: Sample M tasks and roll out K trajectories each with π_θ
- 5: Append replay records to \mathcal{B} $\triangleright R, \text{hiddens } h_{1:T}, \text{goal_emb}, \text{round index}$
- 6: **for** each rollout group of size K **do**
- 7: $A_{\text{traj}} \leftarrow$ group-normalized z -score of R over the K rollouts
- 8: $A_{\text{turn}}^t \leftarrow$ per-turn credit from frozen TurnRD ϕ_r .
- 9: $A_{\text{H}}^t \leftarrow \alpha A_{\text{traj}} + (1 - \alpha) A_{\text{turn}}^t$ \triangleright broadcast over turn t 's tokens
- 10: **end for**
- 11: $\theta \leftarrow$ PPO/GRPO clipped-surrogate update using A_{H} with KL penalty
- 12: Train ϕ_{r+1} from \mathcal{B} using recency-weighted minibatches for the next round
- 13: **end for**
- 14: **return** π_θ

trajectory length. Because the FiLM projector is zero-initialized, the model starts as the unconditioned decomposer and learns goal dependence only when the data supports it. Section 5 reports mechanism probes, and Appendix A contrasts goal-aware and goal-blind credit on matched AlfvWorld trajectories.

Auxiliary TurnRD objective. TurnRD is trained to predict terminal return from per-turn representations while exposing a per-turn credit signal. Its primary objective is the reward-reconstruction loss

$$\mathcal{L}_{\text{pred}} = \mathcal{L}_{\text{MSE}}\left(\sum_t w_t V_t, R\right), \quad (7)$$

which anchors the attention/value factorization to the observed return. In the main runs we add lightweight implementation regularizers rather than changing the core TurnRD objective. A within-batch hinge ranking loss encourages higher-return trajectories to receive larger predicted returns. When environment progress signals are available, a progress-prior term biases attention toward turns where progress changes, and a masked per-turn value loss supervises V_t on active, non-padded turns. These terms improve stability but the defining TurnRD signal remains the reconstruction loss in Eq. (7).

3.4 Putting it together

Algorithm 1 summarizes the round-level training protocol. At the start of round r , the policy loads the latest TurnRD checkpoint trained after round $r-1$ and keeps it frozen during policy optimization. The current LoRA policy then collects K trajectories per task, writes replay records containing per-turn hidden states, goal embeddings, terminal rewards, and any available progress signals, and updates the policy by blending trajectory advantages with frozen TurnRD per-turn credits according to Eq. (2). After the round ends, a separate credit-model update consumes the cumulative replay buffer with recency weighting; the resulting TurnRD checkpoint is loaded by the next round. At $\alpha=1$, the per-turn term vanishes and the policy objective reduces exactly to flatGRPO, independent of the decomposer.

4 Experimental Setup

Policy and optimization. All methods fine-tune Qwen2.5-1.5B-Instruct with a rank-32 LoRA adapter on attention and MLP projections [Hu et al., 2022]. We warm-start from a supervised checkpoint trained on oracle trajectories, so the RL phase improves an already competent instruction-following policy rather than learning the action format from scratch. Policy updates use $K=8$ rollouts per task, PPO-style clipping, a KL penalty to the reference policy, and greedy held-out evaluation ($K=1$) on $n=100$ episodes. Headline table values report the best held-out checkpoint evaluations from multiple seeded runs; the reported variation is the approximate standard deviation across those independent runs. Runs use a single A100-80GB on Modal, ~ 5 – 12 h each.

Environments. *AlfWorld* [Shridhar et al., 2021] is a text embodied household benchmark spanning navigation, object manipulation, receptacle placement, and object-state goals; for compute control, we train on a 1,000-game subset of its training split. *WebShop* [Yao et al., 2022] is a web-navigation shopping benchmark built from 1,000 products, where the agent searches, filters, compares product attributes, and purchases an item matching a natural-language goal. Both environments expose sparse terminal success rewards as the evaluation metric. When available, we use environment-specific progress signals only to instantiate or supervise per-turn decomposers, not as the final metric optimized for reporting.

Methods compared. We compare supervised fine-tuning (SFT); flatGRPO ($\alpha=1$); and HAS ($\alpha=0.5$) with Progressive, LLM-Judge, and TurnRD decomposers. The flatGRPO baseline computes a group-normalized terminal-reward advantage over the K rollouts for a task and broadcasts that same trajectory advantage to every action token, i.e., it is the HAS special case with the per-turn term disabled. Progressive uses environment-derived progress signals, LLM-Judge uses cached turn labels, and TurnRD is our learned goal-aware decomposer. We additionally compare TurnRD with and without FiLM goal-conditioning to isolate the contribution of goal-aware credit. All RL variants use the same SFT warm start, model, LoRA rank, rollout budget, optimizer block, and greedy evaluation protocol; they differ only in the per-turn signal supplied to HAS. Full experimental details and hyperparameters are provided in Appendix B.

5 Results

5.1 Quantitative evaluation

The quantitative results are consistent with two findings. First, HAS improves over trajectory-only GRPO on both benchmarks. Second, the learned TurnRD decomposer performs better than heuristic or judge-based per-turn signals. We report terminal success rate throughout, using the same greedy held-out evaluation protocol for all methods. Checkpoint selection follows the same protocol across methods: each method is evaluated on held-out tasks with greedy decoding after each round, and headline values report the best held-out checkpoint selected by that protocol.

WebShop. Table 1 and Figure 2 compare five methods on WebShop, using three independent runs. SFT alone reaches 40.0%, while flatGRPO raises success to 80.5%, showing that terminal-reward RL is already a strong improvement over imitation. HAS with Progressive and LLM-Judge reaches 86.0% and 82.0%, respectively, both above flatGRPO but below TurnRD. The learned TurnRD decomposer performs best at 91.3%, improving over flatGRPO by +10.8pp and over Progressive and LLM-Judge by +5.3pp and +9.3pp. This ordering indicates that the benefit is not merely the presence of a dense per-turn signal. A replay-trained decomposer adapted to the policy’s own trajectories provides a stronger shaping signal than generic progress heuristics or cached judge labels.

Table 1: WebShop terminal success rate (%), reported as mean \pm approximate standard deviation across independent seeded RL runs. HAS uses $\alpha=0.5$; TurnRD gives the strongest per-turn decomposition. Full hyperparameters and sweeps are in Appendix B (Table 5).

Method	Success rate (% \pm std.)
SFT (no RL)	40.0 \pm 0.0
flatGRPO ($\alpha=1$)	80.5 \pm 2.0
HAS + LLM-Judge	82.0 \pm 4.0
HAS + Progressive	86.0 \pm 2.0
HAS + TurnRD (ours)	91.3 \pm 3.0

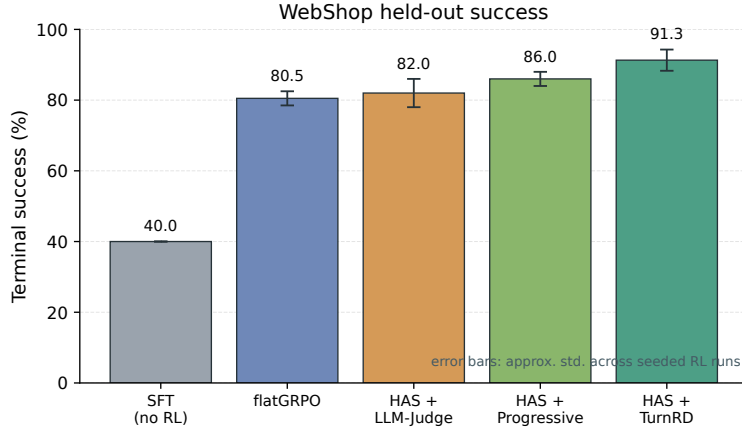


Figure 2: WebShop success by method with error bars showing approximate standard deviation across independent seeded RL runs. HAS with TurnRD improves by +10.8pp over flatGRPO and outperforms heuristic and judge-based decomposers.

AlfWorld. Table 2 and Figure 3 show the same pattern on a longer-horizon embodied benchmark, reported over multiple independent runs. The SFT warm start reaches 47% success, and flatGRPO improves this to 70%, indicating that trajectory-level RL is effective but plateaus below the strongest HAS variant. Adding TurnRD raises held-out success to 75%, and adding FiLM conditioning reaches 80%. Thus, learned per-turn credit contributes about +5pp beyond flatGRPO, and goal-aware FiLM adds another +5pp over the goal-blind TurnRD variant. The early-round curves overlap within uncertainty, but the TurnRD+FiLM advantage emerges later as the credit model accumulates useful replay.

The absolute success rates should not be compared directly across benchmarks. WebShop has a shorter interaction horizon (max 15 steps) and a more structured search/filter/purchase action space, whereas AlfWorld allows up to 40 steps and requires longer compositional action chains. We therefore emphasize within-environment improvements over flatGRPO rather than raw cross-benchmark success levels.

Table 2: AlfWorld held-out terminal success (%), reported as mean \pm approximate standard deviation across independent seeded RL runs. SFT is the oracle-trajectory warm start; flatGRPO uses trajectory-level credit only. TurnRD and FiLM each add measurable lift. Full hyperparameters and sweeps are in Appendix B (Table 6).

Method	Success rate (% \pm std.)
SFT (no RL)	47 \pm 0.0
flatGRPO ($\alpha=1$)	70 \pm 2.0
HAS + TurnRD (no FiLM)	75 \pm 5.0
HAS + TurnRD + FiLM (ours)	80 \pm 3.0

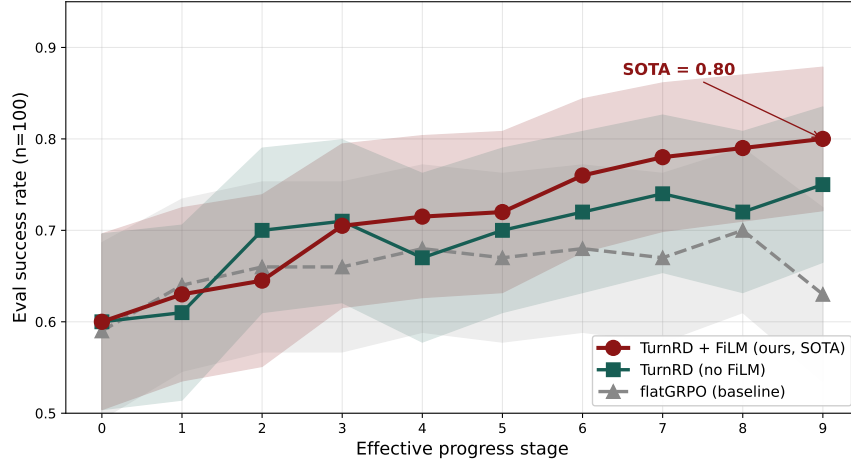


Figure 3: AlfWorld round-by-round held-out success ($n=100$, Wilson 95% CI bands per evaluation round). TurnRD+FiLM reaches 80%, while flatGRPO plateaus below it.

5.2 Ablation and mechanism analysis

Why does the hybrid blend help? The α -sweep supports the bias–variance interpretation of HAS. At $\alpha=1$, the update preserves stable group-level ranking but remains credit-blind within each trajectory. At very small α , the update relies too heavily on decomposed turn-level estimates, which are noisy early in training. The strongest setting is the hybrid regime, $\alpha=0.5$, where trajectory-level ranking anchors the update while TurnRD reshapes credit within the trajectory. This explains why HAS improves most clearly after replay has accumulated and the decomposer becomes better calibrated.

A pilot AlfWorld sweep under a smaller LoRA-rank-16 setup supports this interpretation (Table 3). Although the absolute success rates are not directly comparable to the final rank-32 runs, the relative ordering is informative: $\alpha=0.5$ performs best, while both the trajectory-only setting and more extreme mixtures underperform. We therefore use $\alpha=0.5$ in the main experiments.

Table 3: Pilot AlfWorld α sweep under a smaller LoRA-rank-16 setup. Absolute numbers are not directly comparable to the final rank-32 runs, but the sweep supports the hybrid setting used in the main experiments.

Setting	Success rate (%)
flatGRPO ($\alpha=1.0$)	46
HAS + TurnRD ($\alpha=0.25$)	56
HAS + TurnRD ($\alpha=0.50$)	62
HAS + TurnRD ($\alpha=0.75$)	58

What does TurnRD add beyond dense heuristics? The decomposer comparison isolates the source of the gain. Progressive and LLM-Judge both provide denser turn-level signals, but neither matches TurnRD. This indicates that the important ingredient is not density alone, but a credit model adapted to the policy’s own replay distribution. Static heuristics encode generic notions of progress, whereas TurnRD can track which turns are predictive under the current policy and task distribution.

What does goal-aware FiLM add? The AlfWorld ablation separates goal-blind learned credit from goal-aware learned credit. TurnRD without FiLM already improves over flatGRPO, showing that learned per-turn decomposition is useful. Adding FiLM gives a further +5pp, which is consistent with AlfWorld goals making the same local action useful or irrelevant depending on the instruction. Figure 4 reports three mechanism probes. Zeroing γ/β changes the per-turn V -head credit by 39.7%, shuffling goal embeddings worsens V -head MSE by 3.16% (57.6% of samples worse), and the FiLM modulation strength grows over rounds (γ norm 0.34 \rightarrow 0.83). These probes show that FiLM is used

by the trained model, while the modest shuffle effect suggests that the current goal-awareness signal remains limited.

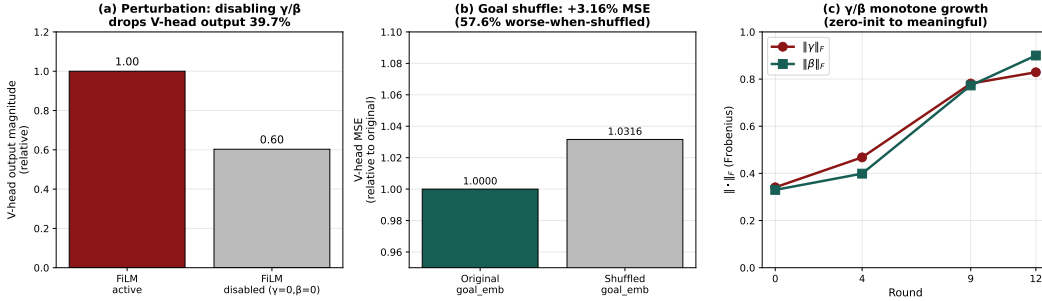


Figure 4: Offline FiLM mechanism probes: (a) perturbation magnitude, (b) goal-shuffle MSE, and (c) γ/β norm growth across rounds.

5.3 Qualitative analysis

The credit visualizations in Appendix A provide a mechanism-level view of the success-rate improvements reported above. On a matched 50-task AlfWorld probe slice, the goal-aware TurnRD+FiLM policy solved 44 tasks versus 39 for the goal-blind TurnRD variant; it solved every task solved by the goal-blind policy plus five additional tasks, with no reversals. This slice is smaller than the $n=100$ held-out evaluation behind the headline numbers, so we use it as qualitative evidence rather than a second benchmark. The probe evaluates whether TurnRD+FiLM assigns credit to semantically decisive turns, not only whether the resulting policy succeeds.

Across the matched examples, TurnRD concentrates positive credit on pivotal turns, such as locating the target object, moving it to the right receptacle, or issuing the final while assigning small or negative credit to detours. In cases where the goal-aware policy succeeds and the goal-blind policy fails, TurnRD+FiLM assigns high positive credit to instruction-relevant turns, while the goal-blind variant assigns credit to locally plausible but ultimately wrong actions. In shared-success cases, both decomposers identify terminal goal actions, but the goal-aware model places less credit on detours. In shared failures, negative credit concentrates on turns that misdirect the trajectory. These examples show how learned goal-aware credit differs from flatGRPO’s uniform trajectory credit and from heuristic progress signals, which can reward local changes that do not help the instruction.

Beyond the visualized cases, we inspected the top-credit turns on the 50-task probe slice. In successful TurnRD+FiLM trajectories, the highest-credit turn typically corresponds to a decisive progress action, such as acquiring the target object, opening the correct container, or moving to the goal-relevant location. The most informative contrasts are goal-sensitive: FiLM increases credit for actions that advance the specified instruction and suppresses generic exploration steps that are locally plausible but goal-irrelevant. These cases support the central mechanism claim: TurnRD+FiLM does not merely densify reward, but changes which turns receive credit based on the goal.

6 Discussion

Together, the α sweep, decomposer comparison, FiLM probes, and qualitative audit suggest that HAS helps for two separable reasons: trajectory-level normalization stabilizes policy updates, while TurnRD supplies goal-aware intra-trajectory structure that flatGRPO cannot represent. Additional constrained-budget ablations on TurnRD capacity, auxiliary objectives, group contrast, and replay scheduling support this interpretation.

Limitations and threats to validity. The main training risk is distribution lag between the policy and the credit model. During policy optimization, TurnRD is frozen; after each round, its checkpoint is updated from the cumulative replay buffer with recency weighting and then loaded for the next round. This decoupling avoids online credit-model drift within a policy update, but the decomposer can still be imperfectly calibrated when the policy distribution shifts across rounds. Recency weighting mitigates

stale replay, but does not eliminate this lag. Second, the goal-shuffle probe shows goal dependence, but the effect is modest. One plausible reason is that goal-aware variation in raw TurnRD values is attenuated by the downstream credit pipeline: projected per-turn credits are group-normalized, then blended with the trajectory advantage before entering the PPO objective. Thus, FiLM may be useful even when the measured goal-shuffle effect is small; stronger goal-discrimination losses or adaptive normalization could make this signal more visible. Third, all reported policy updates are LoRA-only on a 1.5B model, leaving open whether larger or fully fine-tuned policies would require different α schedules. Finally, the AlfWorld curve is noisy in early rounds. The strongest claim supported by the data is a later-stage advantage once TurnRD has enough replay to learn useful credit, not monotonic dominance from round 0.

Why is AlfWorld noisier than WebShop? AlfWorld has higher round-to-round variance because episodes are longer-horizon, partially observable, and evaluated with sparse binary success. A single early irreversible mistake can determine the entire trajectory, and the same local action can be useful or irrelevant depending on the goal. WebShop exposes more stable lexical and retrieval-based progress cues, so dense credit signals separate methods more smoothly in aggregate success rate. We therefore interpret AlfWorld ablations by trends across rounds and relative ordering rather than by a single-round success value.

7 Conclusion

This work studies credit assignment as a central bottleneck in sparse-reward RL for multi-turn LLM agents. We introduced Hybrid Advantage Shaping, a GRPO-compatible framework that preserves the stable group-relative trajectory objective while adding learned per-turn credit through a pluggable decomposer. Instantiated with TurnRD, a goal-aware attention-based decomposer, the framework improves success on both WebShop and AlfWorld relative to flatGRPO and to the heuristic or judge-based turn-level signals we evaluated.

These results indicate that dense credit is most useful when anchored to reliable trajectory-level comparisons. flatGRPO supplies stable group-relative ranking but cannot distinguish decisive turns from incidental ones; TurnRD adds intra-trajectory structure by learning from replay and conditioning credit on the task goal. The gains across two qualitatively different environments suggest that learned, goal-aware turn-level credit is a useful component for GRPO-style optimization of LLM agents.

More broadly, HAS provides a modular interface for studying credit granularity in agentic RL: $\alpha=1$ recovers flatGRPO, while intermediate values expose how much learned per-turn structure should influence the policy update. This makes the framework useful both as an empirical method and as a tool for analyzing when decomposed credit helps or fails in sparse-reward language-agent training.

8 Team Contributions

- **Joseph Li:** Researched credit assignment for LLM agents; proposed the HAS and TurnRD directions; designed the main experiments; implemented training infrastructure, RL algorithms, replay-buffer plumbing, and goal-conditioning with FiLM; and conducted more than 30 experiments across AlfWorld and WebShop.
- **Max Rodriguez:** Fine-tuned the supervised warm-start policy, explored alternative modeling and optimization approaches, and helped debug and improve WebShop model performance.
- **Samantha Leventis:** Ran training and evaluation scripts and trained WebShop models from SFT through RL variants, including TurnRD, LLM-Judge, Progressive, and flatGRPO.

Changes from Proposal. The original proposal framed the project as H-GRPO with several turn-level reward decomposition strategies. During experimentation, we consolidated the method around the HAS formulation and TurnRD as the strongest learned decomposer, and we added goal-aware FiLM conditioning to TurnRD after observing that goal-blind credit assignment was insufficient for instruction-conditioned AlfWorld tasks. We also completed the flatGRPO baseline and expanded the evaluation to include Progressive and LLM-Judge decomposer comparisons, making the final report focus on which source of per-turn credit actually improves sparse-reward LLM-agent RL.

References

- Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
- 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 (NeurIPS)*, 2019.
- Xiushi Chen, Gaotang Li, Ziqi Wang, Bowen Jin, Cheng Qian, Yu Wang, Hongru Wang, Yu Zhang, Denghui Zhang, Tong Zhang, Hanghang Tong, and Heng Ji. RM-R1: Reward modeling as reasoning, 2025.
- Yunzhen Feng et al. GiGPO: Step-level group relative policy optimization for long-horizon reasoning, 2025.
- Daya Guo, Dejian Yang, Haowei Zhang, et al. DeepSeek-R1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- Anna Harutyunyan, Will Dabney, Thomas Mesnard, Mohammad Gheshlaghi Azar, Bilal Piot, Nicolas Heess, Hado van Hasselt, Greg Wayne, Satinder Singh, Doina Precup, and Rémi Munos. Hindsight credit assignment. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.
- Shuo He, Lang Feng, Qi Wei, Xin Cheng, Lei Feng, and Bo An. Hierarchy-of-groups policy optimization for long-horizon agentic tasks. In *International Conference on Learning Representations (ICLR)*, 2026.
- 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 (ICLR)*, 2022.
- Muhammad Khalifa, Rishabh Agarwal, Lajanugen Logeswaran, Jaekyeom Kim, Hao Peng, Moontae Lee, Honglak Lee, and Lu Wang. Process reward models that think, 2025.
- Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022.
- Jiangweizhi Peng, Yuanxin Liu, Ruida Zhou, Charles Fleming, Zhaoran Wang, Alfredo Garcia, and Mingyi Hong. HiPER: Hierarchical reinforcement learning with explicit credit assignment for large language model agents, 2026.
- Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *AAAI Conference on Artificial Intelligence*, 2018.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. In *arXiv preprint arXiv:1707.06347*, 2017.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, 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.
- 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 (ICLR)*, 2021.
- Yifan Song, Da Yin, Xiang Yue, Jie Huang, Sujian Li, and Bill Yuchen Lin. Trial and error: Exploration-based trajectory optimization for llm agents. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.
- Daoyu Wang, Qingchuan Li, Mingyue Cheng, Jie Ouyang, Shuo Yu, Qi Liu, and Enhong Chen. StepPO: Step-aligned policy optimization for agentic reinforcement learning, 2026.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, et al. RAGEN: Understanding self-evolution in llm agents via multi-turn reinforcement learning, 2025.

Yuhuai Wu et al. Let’s verify step by step, 2023.

Zhiheng Xi, Chenyang Liao, Guanyu Li, Yajie Yang, Wenxiang Chen, Zhihao Zhang, Binghai Wang, Senjie Jin, Yuhao Zhou, and Jian Xie. AgentPRM: Process reward models for llm agents via step-wise promise and progress, 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 (NeurIPS)*, 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 (ICLR)*, 2023.

Qiyong Yu et al. DAPO: An open-source llm reinforcement learning system at scale, 2025.

Yu Yue, Yang Yuan, et al. VAPO: Efficient and reliable reinforcement learning for advanced reasoning tasks, 2025.

Shunyu Zhou et al. ArCher: Training language model agents via hierarchical reinforcement learning, 2024.

A Goal-aware vs. goal-blind credit assignment

To make the effect of FiLM goal-conditioning concrete, we probe the trained TurnRD decomposer on held-out AlfWorld trajectories and visualize its per-turn credit. Each turn shows the observation and the agent’s action; the action cell is shaded by the per-turn credit (green = positive credit, red = blame), normalized per trajectory, with the most decisive turn marked ★. By construction the per-turn credits sum to the trajectory’s terminal reward via the value projection in Eq. (5). We contrast the *goal-aware* decomposer (TurnRD + FiLM) against the *goal-blind* variant (TurnRD without FiLM) on three matched cases. The headline case, shown first, is a task where the goal-aware policy succeeds while the goal-blind one fails; the other two cases show a shared success and a shared failure. In this probe slice, TurnRD+FiLM solves five additional tasks with no reversals relative to the goal-blind variant, while the examples below illustrate the mechanism: FiLM changes where credit and blame concentrate for the same instruction-conditioned task.

Remaining failures fall into three recurring categories: early navigation errors that prevent later recovery, object-aliasing mistakes where the policy interacts with a plausible but wrong object, and credit diffusion in long trajectories where several preparatory actions are needed before terminal success.

Case 1: goal-aware succeeds, goal-blind fails

The headline qualitative case: goal-aware credit lands on the pivotal goal turn, whereas goal-blind credit is mis-allocated.

Goal-Aware (TurnRD + FiLM):

Goal-aware rollout: Success (R = +1.00)

Turn 0: -= Welcome to TextWorld, ALFRED! -= You are in the middle of a room. Looking quickly around you, you see a bed 1, a desk 1, a drawer 17, a drawer 16, a drawer 15, a drawer 14, a drawer 13, a drawer 12, a drawer 11, a drawer 10, a drawer 9, a drawer 8, a drawer 7, a drawer 6, a drawer 5, a drawer 4, a drawer 3, a drawer 2, a drawer 1, a dresser ... (*initial*)

Action: look

- Turn 1:** You are in the middle of a room. Looking quickly around you, you see nothing.
Action: go to shelf 1 (*credit = +0.075*)
- Turn 2:** You arrive at shelf 1. On the shelf 1, you see a pencil 1, and a watch 1.
Action: look (*credit = +0.084*)
- Turn 3:** You are facing the shelf 2, and shelf 1. Next to it, you see nothing.
Action: go to drawer 10 (*credit = +0.001*)
- Turn 4:** You arrive at drawer 10. On the drawer 10, you see a creditcard 3.
Action: go to shelf 2 (*credit = -0.026*)
- Turn 5:** You arrive at shelf 2. On the shelf 2, you see a tissuebox 1.
Action: go to shelf 3 (*credit = -0.129*)
- Turn 6:** You arrive at shelf 3. On the shelf 3, you see nothing.
Action: go to shelf 4 (*credit = -0.004*)
- Turn 7:** You arrive at shelf 4. On the shelf 4, you see a pen 2.
Action: go to shelf 5 (*credit = -0.087*)
- Turn 8:** You arrive at shelf 5. On the shelf 5, you see a statue 1.
Action: go to garbagecan 1 (*credit = +0.006*)
- Turn 9:** You arrive at garbagecan 1. On the garbagecan 1, you see nothing.
Action: go to dresser 1 (*credit = -0.231*)
- Turn 10:** You arrive at dresser 1. On the dresser 1, you see a alarmclock 2, a cd 1, a creditcard 2, a desklamp 1, and a mug 2.
Action: use desklamp 1 (*credit = +0.758*) ★ (*decisive*)
- Turn 11:** You turn on the desklamp 1.
Action: take mug 2 from dresser 1 (*credit = +0.507*)

Goal-Blind (TurnRD, no FiLM):

Goal-blind rollout: Failure (R = +0.00)

Showing Turn 0 + turns 7–18 of 40 (window around the fatal turn).

- Turn 0:** -= Welcome to TextWorld, ALFRED! -= You are in the middle of a room. Looking quickly around you, you see a bed 1, a desk 1, a drawer 17, a drawer 16, a drawer 15, a drawer 14, a drawer 13, a drawer 12, a drawer 11, a drawer 10, a drawer 9, a drawer 8, a drawer 7, a drawer 6, a drawer 5, a drawer 4, a drawer 3, a drawer 2, a drawer 1, a dresser ... (*initial*)
Action: look
- ⋮ (*turns 1–6 elided*)
- Turn 7:** You arrive at shelf 4. On the shelf 4, you see a pen 2.
Action: go to shelf 5 (*credit = +0.159*)
- Turn 8:** You arrive at shelf 5. On the shelf 5, you see a statue 1.
Action: go to garbagecan 1 (*credit = +0.191*)
- Turn 9:** You arrive at garbagecan 1. On the garbagecan 1, you see nothing.
Action: go to dresser 1 (*credit = +0.011*)
- Turn 10:** You arrive at dresser 1. On the dresser 1, you see a alarmclock 2, a cd 1, a creditcard 2, a desklamp 1, and a mug 2.
Action: use desklamp 1 (*credit = +0.916*)
- Turn 11:** You turn on the desklamp 1.
Action: take cd 1 from dresser 1 (*credit = +0.215*)

Turn 12: You pick up the cd 1 from the dresser 1.
Action: go to safe 1 (credit = -0.748)

Turn 13: Nothing happens.
Action: move cd 1 to safe 1 (credit = -1.099) ★ (fatal)

Turn 14: Nothing happens.
Action: go to drawer 1 (credit = -0.134)

Turn 15: You arrive at drawer 1. The drawer 1 is closed.
Action: open drawer 1 (credit = +1.033)

Turn 16: You open the drawer 1. The drawer 1 is open. In it, you see nothing.
Action: close drawer 1 (credit = -0.917)

Turn 17: You close the drawer 1.
Action: go to drawer 2 (credit = -0.001)

Turn 18: You arrive at drawer 2. The drawer 2 is closed.
Action: open drawer 2 (credit = +1.033)

⋮ (turns 19–39 elided; trajectory ran to step limit without reaching the goal)

Case 2: shared success

Both policies solve the task; goal-aware credit concentrates on the goal-relevant turns, while goal-blind credit is more diffuse.

Goal-Aware (TurnRD + FiLM):

Rollout: Success (R = +1.00)

Turn 0: -= Welcome to TextWorld, ALFRED! -= You are in the middle of a room. Looking quickly around you, you see a bed 1, a desk 1, a drawer 5, a drawer 4, a drawer 3, a drawer 2, a drawer 1, a garbagecan 1, a laundryhamper 1, a shelf 1, and a sidetable 1. Your task is to: examine the pillow with the desklamp. (initial)
Action: look

Turn 1: You are in the middle of a room. Looking quickly around you, you see nothing.
Action: go to bed 1 (credit = -0.097)

Turn 2: You arrive at bed 1. On the bed 1, you see a book 1, a cellphone 1, a laptop 2, a laptop 1, a pillow 1, and a teddybear 1.
Action: take pillow 1 from bed 1 (credit = +0.883) ★ (decisive)

Turn 3: You pick up the pillow 1 from the bed 1.
Action: go to sidetable 1 (credit = -0.080)

Turn 4: You arrive at sidetable 1. On the sidetable 1, you see a cd 1, and a pen 1.
Action: go to laundryhamper 1 (credit = -0.212)

Turn 5: You arrive at laundryhamper 1. On the laundryhamper 1, you see nothing.
Action: go to garbagecan 1 (credit = -0.119)

Turn 6: You arrive at garbagecan 1. On the garbagecan 1, you see nothing.
Action: go to desk 1 (credit = -0.210)

Turn 7: You arrive at desk 1. On the desk 1, you see an alarmclock 1, a book 2, a desklamp 1, a pen 3, and a pen 2.
Action: use desklamp 1 (credit = +0.883)

Goal-Blind (TurnRD, no FiLM):

Rollout: Success (R = +1.00)

Turn 0: -= Welcome to TextWorld, ALFRED! -= You are in the middle of a room. Looking quickly around you, you see a bed 1, a desk 1, a drawer 5, a drawer 4, a drawer 3, a drawer 2, a drawer 1, a garbagecan 1, a laundryhamper 1, a shelf 1, and a sidetable 1. Your task is to: examine the pillow with the desklamp. *(initial)*

Action: look

Turn 1: You are in the middle of a room. Looking quickly around you, you see nothing.

Action: go to bed 1 *(credit = -0.138)*

Turn 2: You arrive at bed 1. On the bed 1, you see a book 1, a cellphone 1, a laptop 2, a laptop 1, a pillow 1, and a teddybear 1.

Action: take pillow 1 from bed 1 *(credit = +0.776)*

Turn 3: You pick up the pillow 1 from the bed 1.

Action: go to sidetable 1 *(credit = -0.231)*

Turn 4: You arrive at sidetable 1. On the sidetable 1, you see a cd 1, and a pen 1.

Action: go to laundryhamper 1 *(credit = -0.027)*

Turn 5: You arrive at laundryhamper 1. On the laundryhamper 1, you see nothing.

Action: go to garbagecan 1 *(credit = -0.043)*

Turn 6: You arrive at garbagecan 1. On the garbagecan 1, you see nothing.

Action: go to desk 1 *(credit = -0.064)*

Turn 7: You arrive at desk 1. On the desk 1, you see a alarmclock 1, a book 2, a desklamp 1, a pen 3, and a pen 2.

Action: use desklamp 1 *(credit = +0.860) ★ (decisive)*

Case 3: shared failure

Both policies fail; compare where each decomposer assigns blame.

Goal-Aware (TurnRD + FiLM):

Rollout: Failure (R = +0.00)

Showing Turn 0 + turns 28–39 of 40 (window around the fatal turn).

Turn 0: -= Welcome to TextWorld, ALFRED! -= You are in the middle of a room. Looking quickly around you, you see a bed 1, a desk 1, a drawer 5, a drawer 4, a drawer 3, a drawer 2, a drawer 1, a dresser 1, a garbagecan 1, a shelf 12, a shelf 11, a shelf 10, a shelf 9, a shelf 8, a shelf 7, a shelf 6, a shelf 5, a shelf 4, a shelf 3, a shelf 2, a shelf 1... *(initial)*

Action: look

⋮ *(turns 1–27 elided)*

Turn 28: You open the drawer 3. The drawer 3 is open. In it, you see a keychain 3.

Action: close drawer 3 *(credit = -0.737)*

Turn 29: You close the drawer 3.

Action: go to drawer 4 *(credit = -0.214)*

Turn 30: You arrive at drawer 4. The drawer 4 is closed.

Action: open drawer 4 *(credit = +0.956)*

Turn 31: You open the drawer 4. The drawer 4 is open. In it, you see nothing.

Action: close drawer 4 *(credit = -0.857)*

Turn 32: You close the drawer 4.

Action: go to drawer 5 *(credit = +0.138)*

- Turn 33:** You arrive at drawer 5. The drawer 5 is closed.
Action: **open drawer 5** (credit = +1.205)
- Turn 34:** You open the drawer 5. The drawer 5 is open. In it, you see nothing.
Action: **close drawer 5** (credit = -0.883) ★ (fatal)
- Turn 35:** You close the drawer 5.
Action: go to desk 1 (credit = -0.058)
- Turn 36:** You arrive at desk 1. On the desk 1, you see a cd 3, a cd 2, a desklamp 1, a keychain 2, a laptop 1, a pen 1, and a vase 2.
Action: **use desklamp 1** (credit = +1.061)
- Turn 37:** You turn on the desklamp 1.
Action: take cd 2 from desk 1 (credit = +0.038)
- Turn 38:** You pick up the cd 2 from the desk 1.
Action: use desklamp 1 (credit = -0.187)
- Turn 39:** You turn on the desklamp 1.
Action: **move cd 2 to shelf 1** (credit = -0.345)

Goal-Blind (TurnRD, no FiLM):

Rollout: Failure (R = +0.00)

Showing Turn 0 + turns 21–32 of 40 (window around the fatal turn).

- Turn 0:** -= Welcome to TextWorld, ALFRED! -= You are in the middle of a room. Looking quickly around you, you see a bed 1, a desk 1, a drawer 5, a drawer 4, a drawer 3, a drawer 2, a drawer 1, a dresser 1, a garbagecan 1, a shelf 12, a shelf 11, a shelf 10, a shelf 9, a shelf 8, a shelf 7, a shelf 6, a shelf 5, a shelf 4, a shelf 3, a shelf 2, a shelf 1... (initial)
Action: look

∴ (turns 1–20 elided)

- Turn 21:** You arrive at shelf 1. On the shelf 1, you see a mug 1.
Action: **go to drawer 1** (credit = -0.471)
- Turn 22:** You arrive at drawer 1. On the drawer 1, you see a creditcard 1, a keychain 1, and a pencil 3.
Action: **look** (credit = -0.568)
- Turn 23:** You are facing the drawer 2, drawer 1, and dresser 1. Next to it, you see nothing.
Action: **go to shelf 1** (credit = -0.449)
- Turn 24:** You arrive at shelf 1. On the shelf 1, you see a mug 1.
Action: **go to drawer 2** (credit = -0.537)
- Turn 25:** You arrive at drawer 2. On the drawer 2, you see nothing.
Action: **go to drawer 3** (credit = -0.308)
- Turn 26:** You arrive at drawer 3. The drawer 3 is closed.
Action: **open drawer 3** (credit = +1.106)
- Turn 27:** You open the drawer 3. The drawer 3 is open. In it, you see a keychain 3.
Action: **close drawer 3** (credit = -0.950) ★ (fatal)
- Turn 28:** You close the drawer 3.
Action: **go to drawer 4** (credit = -0.166)
- Turn 29:** You arrive at drawer 4. The drawer 4 is closed.
Action: **open drawer 4** (credit = +1.106)

Turn 30: You open the drawer 4. The drawer 4 is open. In it, you see nothing.

Action: `close drawer 4` (*credit* = -0.814)

Turn 31: You close the drawer 4.

Action: `go to drawer 5` (*credit* = +0.155)

Turn 32: You arrive at drawer 5. The drawer 5 is closed.

Action: `open drawer 5` (*credit* = +1.106)

∴ (turns 33–39 elided; trajectory ran to step limit without reaching the goal)

B Experiment Details

Data and evaluation. AlfWorld experiments use a 1,000-game subset of the training split for compute control and evaluate on a held-out pool with greedy decoding. WebShop experiments use the 1,000-product setup described in the main text. We report terminal success rate for both environments; dense progress signals are used only for decomposer construction or supervision.

SFT warm start and policy updates. The initial policy is Qwen2.5-1.5B-Instruct with a rank-32 LoRA adapter trained from oracle trajectories. RL updates continue from this SFT checkpoint and keep the base model frozen, updating only LoRA parameters on attention and MLP projection modules. Policy updates use $K=8$ rollouts per task, PPO-style clipping, and a KL penalty to a reference policy.

Replay and TurnRD updates. Replay records store per-turn encoder hidden states, per-trajectory goal embeddings, terminal rewards, round indices, and any available progress signals. The replay buffer is cumulative and recency-weighted rather than physically trimmed, so older rounds remain available but newer policy data has larger influence. TurnRD checkpoint updates run separately from policy optimization, and the resulting checkpoint is used as a frozen credit model during the policy stage.

Compute and logging. Runs execute on Modal A100-80GB workers. Per-round rollout, TurnRD checkpoint, policy-update, and evaluation statistics are logged to manifests, including evaluation `pct_success`. Table 4 summarizes the shared hyperparameters.

Table 4: Shared hyperparameters.

Setting	Value
Base model	Qwen2.5-1.5B-Instruct
LoRA rank / targets	32 / attention + MLP
Rounds	10
Episodes per round	80
Rollouts per task K	8
Mixing coefficient α	0.5 (HAS), 1.0 (flatGRPO)
TurnRD encoder	2 layers, 4 heads, hidden $H=128$
Recency half-life H	4 rounds
Eval	greedy, $n=100$ held-out / round
Hardware	A100-80GB (Modal)

Per-environment hyperparameters. Tables 5 and 6 list the full configuration of the headline WebShop (HAS + TurnRD) and AlfWorld (HAS + TurnRD + FiLM) runs, exactly as used to produce the results in Tables 1 and 2. The two recipes are deliberately close: the GRPO/PPO optimizer block, LoRA configuration, and HAS blend are shared, and the main environment-specific differences are the interaction horizon (`max_steps` 15 vs. 40), the dense intermediate-reward source (attribute-progress for WebShop vs. TextWorld + facts-diff for AlfWorld), the TurnRD value-head weight λ_V (0.5 vs. 1.0, since AlfWorld’s facts-diff signal is denser), the TurnRD `max_turns` bound (mirrors `max_steps`), and the FiLM goal-conditioning knobs, which are enabled only for the AlfWorld TurnRD + FiLM variant.

Table 5: WebShop hyperparameters (HAS + TurnRD).

Setting	Value
<i>Environment</i>	
Environment / horizon	WebShop, max_steps=15
Observation mode	text
Dense intermediate reward	attribute-progress (per-turn)
<i>Policy / optimization (GRPO/PPO)</i>	
Base model	Qwen2.5-1.5B-Instruct
LoRA rank / targets	32 / attention + MLP (7 modules)
Learning rate	5×10^{-6}
Rollouts per task K	8
PPO clip ϵ	0.2
Target KL (adaptive)	0.04
Batch size / grad-accum	4 / 2
Max tokens / microbatch	1024
Rounds / episodes per round	10 / 80
Rollout temperature	1.0
Eval episodes	100 (greedy)
<i>HAS blend</i>	
Mixing coefficient α	0.5
Decomposer	TurnRD (v-projection)
V-projection clamp	2.0
<i>TurnRD credit model</i>	
Encoder	2 layers, 4 heads, $H=128$, non-causal
max_turns / dropout	16 / 0.1
TurnRD learning rate	5×10^{-4}
TurnRD epochs / round	5
$\lambda_V / \lambda_{rank} / \lambda_{prog}$	0.5 / 0.1 / 0.01
Rank margin	0.1
Recency half-life	4 rounds

Table 6: AlfWorld hyperparameters (HAS + TurnRD + FiLM).

Setting	Value
<i>Environment</i>	
Environment / horizon	AlfWorld, max_steps=40
Observation mode	text
Dense intermediate reward	TextWorld + facts-diff (per-turn)
Train games	1,000-game subset
<i>Policy / optimization (GRPO/PPO)</i>	
Base model	Qwen2.5-1.5B-Instruct
LoRA rank / targets	32 / attention + MLP (7 modules)
Learning rate	5×10^{-6}
Rollouts per task K	8
PPO clip ϵ	0.2
Target KL (adaptive)	0.04
Batch size / grad-accum	4 / 2
Max tokens / microbatch	1024
Rounds / episodes per round	10 / 80
Rollout temperature	1.0
Eval episodes	100 (greedy)
<i>HAS blend</i>	
Mixing coefficient α	0.5
Decomposer	TurnRD (v-projection)
V-projection clamp	2.0
<i>TurnRD credit model</i>	
Encoder	2 layers, 4 heads, $H=128$, non-causal
max_turns / dropout	50 / 0.1
TurnRD learning rate	5×10^{-4}
TurnRD epochs / round	5
$\lambda_V / \lambda_{rank} / \lambda_{prog}$	1.0 / 0.1 / 0.01
Rank margin	0.1
Recency half-life	4 rounds
<i>FiLM goal-conditioning (TurnRD+FiLM only)</i>	
Goal text / goal embedding	emitted
Goal-conditioned value head	enabled (zero-init FiLM)

Hyperparameter sweeps. The values above were selected from small grids rather than chosen a priori. We swept the policy learning rate over $\{1 \times 10^{-6}, 2 \times 10^{-6}, 5 \times 10^{-6}, 1 \times 10^{-5}\}$ and selected 5×10^{-6} , which gave the most stable improvement without the late-round collapse seen at higher rates. For the PPO/GRPO update we swept the KL-penalty coefficient over $\{0.02, 0.04, 0.1\}$ (selecting 0.04 as the best trade-off between staying close to the reference policy and allowing improvement) and confirmed the clip range ϵ on a coarse $\{0.1, 0.2, 0.3\}$ scan, selecting 0.2. We also swept the number of rollouts per task over $K \in \{4, 8, 12\}$ and the LoRA rank over $\{16, 32\}$. Increasing K from 4 to 8 improved within-group reward variation and stability, while $K=12$ gave only a modest further gain that did not justify its higher rollout cost under our budget constraints, so $K=8$ was the most compute-efficient setting at the ideal performance level. Rank 32 outperformed rank 16 and was used in all main runs. Sweeps were run on the constrained AlfWorld protocol and the selected optimizer block (learning rate, KL coefficient, clip range, K , and LoRA rank) was then transferred unchanged to WebShop, consistent with the recipe-transplant design noted in the configs. The remaining knobs (HAS α , TurnRD architecture) were held fixed across these sweeps.

C Additional Ablations and Diagnostics

This section reports additional ablations that support the main claims: learned turn-level credit is useful when it is calibrated to the policy distribution, anchored by trajectory-level comparisons, and

regularized without collapsing into a hand-engineered progress heuristic. These ablations were run under constrained budgets, shorter horizons, and in some cases smaller evaluation sets than the main experiments, so their absolute success rates should not be compared directly to Tables 1–2. We use them to analyze relative trends and failure modes. The main α sweep is reported in Section 5; here we focus on complementary ablations of TurnRD capacity, auxiliary objectives, group contrast, and replay scheduling.

TurnRD capacity and convergence. We also ran a larger TurnRD variant under a constrained AlfWorld protocol. Its per-round success curve was 60, 61, 70, 71, 67, 70, 72, 74, 72, 75, 75, 75% through rounds 0–11. This supports the capacity hypothesis: the larger decomposer improves faster in the first few rounds, but later gains flatten around 75% rather than continuing upward. A likely explanation is a bias–variance tradeoff in cumulative replay. Higher capacity fits sparse return supervision quickly, but stale trajectories from older policy distributions can interfere with later calibration even under recency weighting.

Auxiliary TurnRD objectives. Progress-value auxiliary variants gave smaller and less consistent gains than the standard TurnRD recipe. A progress-value run reached 50, 50, 54, 57, 55% over five rounds, while a companion progress-value run improved more smoothly to 61% by round 4. Stronger auxiliary-weight sweeps, including $(\lambda_V=1.0, \lambda_{prog}=1.0)$ and $(\lambda_V=2.0, \lambda_{prog}=1.0)$, remained around 50–54% in short pilots. These results suggest that progress auxiliaries can regularize early learning, but over-weighting them risks collapsing TurnRD toward a hand-engineered Progressive signal rather than a replay-adapted credit model.

Dead- K and group contrast. Increasing rollout group size from $K=4$ to $K=8$ and skipping policy-gradient updates for groups with identical terminal rewards improved the reliability of the constrained AlfWorld sweep. A matched hybrid rerun rose monotonically from 49% to 58% over five rounds, whereas the earlier $K=4$ comparison stayed near a 40–46% noise floor. The result does not isolate K from the skip guard, but it supports the practical conclusion that sparse-reward GRPO needs enough within-group reward variation before learned turn credit can matter.

Replay curriculum and recency weighting. Replay-curriculum pilots were mixed. The phase-2 run reached 48, 49, 52, 58, 57%, roughly matching the standard constrained rerun by round 4 but not exceeding it. This is consistent with the distribution-lag concern in Section 6: curriculum and recency choices affect stability, yet they do not remove the need for a well-calibrated decomposer. In the larger-TurnRD run, the late plateau is the clearest evidence that replay weighting and decomposer capacity should be tuned jointly.