FORGE: FEEDBACK-OPTIMIZED RUNTIME GUIDANCE ENGINE FOR AGENTIC RL

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning (RL) improves language agents through interaction, but sparse terminal rewards can leave difficult tasks without informative training feedback. In Group Relative Policy Optimization (GRPO), an all-failure rollout group yields zero task-reward advantages, removing its contribution to the task-directed policy gradient. Guidance can open new paths to success, yet its value for training depends on whether the resulting experience improves the students subsequent unassisted behavior. We introduce **FORGE** (**F**eedback-**O**ptimized **R**untime **G**uidance **E**ngine) that trains a harnessed guidance agent alongside the student using post-update learning value. When a clean rollout group entirely fails, FORGE launches candidate tutoring interactions, adapting its guidance to the students evolving execution history. An episode workspace and a retrievable experience library support decisions about what guidance to provide, when to intervene, and when to end assistance. To evaluate each interaction, FORGE compares paired temporary student updates with and without its trajectory group, measuring the resulting loss difference on independent, unassisted probe trajectories. This estimates the candidates contribution to student learning beyond the common training data. The resulting feedback trains the guidance policy over complete interactions, including retrieval and follow-up decisions. All eligible candidates contribute to guidance-policy training, while at most the highest-value positive group enters the students task-reward update. Experiments on multiple interactive agent benchmarks show that FORGE outperforms GRPO and prior guidance-based baselines in unassisted task performance, demonstrating the effectiveness of training dynamic guidance with post-update learning value.

1 INTRODUCTION

Reinforcement learning (RL) enables language agents to learn from their interactions with an environment. Tasks such as household manipulation and online shopping require a sequence of decisions before success can be checked (Shridhar et al., 2020; Yao et al., 2022). A missed precondition or an incorrect action early in the sequence can cause the entire attempt to fail.

Group Relative Policy Optimization (GRPO) (Shao et al., 2024) learns from these outcomes by comparing rewards within a group of sampled rollouts. Under binary terminal rewards, however, a group in which every rollout fails has zero task-reward advantages and contributes no task-directed policy gradient. Repeated failures therefore leave difficult tasks consuming interaction budget without providing a reward-based direction for improvement.

Several approaches seek to recover useful training signal from this situation. Dynamic sampling filters groups with identical rewards and collects additional samples (Yu et al., 2025). Scaf-GRPO provides progressively stronger scaffolding, while NuRL injects self-generated hints into questions with all-incorrect rollouts (Zhang et al., 2026b; Chen et al., 2026). HiLL goes further by jointly training a hinter and a reasoner: hints adapt to the reasoner's current failures, and a hint-reliance measure encourages transfer to unassisted reasoning (Xia et al., 2026). These methods make guidance part of the training process. In interactive environments, this raises a further challenge: the information needed to guide a student changes as the student acts.

Extending guidance to interactive agents requires addressing two connected problems. **Guidance must remain useful as execution unfolds.** An initial hint may identify the right subgoal, yet sub-

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.

sequent actions can expose a missing precondition or lead the student into repeated failures. The guide therefore needs access to the current execution history and a mechanism for deciding when to intervene again. **Guided success must produce useful training experience.** Recovering successful rollouts can restore variation in group rewards, but the resulting update may still favor behavior that depends on assistance. The training value of an intervention consequently depends on what the student learns from the complete trajectory group, including its unsuccessful attempts.

These problems call for a guide that can both respond to execution feedback and learn from the effect of its assistance on the student's next update.

We introduce **FORGE** (**F**eedback-**O**ptimized **R**untime **G**uidance **E**ngine), a framework for training a harnessed guidance agent alongside a student under GRPO. FORGE combines two design choices: *runtime guidance*, which adapts assistance to the student's observed progress, and *post-update learning value*, which evaluates the experience produced by that assistance. Together, they connect the guide's decisions during an episode to the student's subsequent unassisted learning.

FORGE intervenes when an eligible group of unassisted student rollouts entirely fails. Given the recorded failures, the guidance agent generates multiple candidate tutoring interactions. Each candidate begins with a shared initial guidance decision and launches a group of student trajectories from the original environment state. As these trajectories diverge, the guide can provide further instructions, wait for additional evidence, or end assistance. An episode workspace maintains the failure records, current observations, and prior guidance, while an experience library makes relevant information from earlier training events available for retrieval. The runtime harness manages this context and the delivery of interventions; the guidance policy learns which information to retrieve and how to use it in its instructions and follow-up decisions. A candidate thus comprises the full interaction through which the guide and student produce a trajectory group.

The next step is to determine whether that group is useful for learning. FORGE evaluates each candidate against a shared control update using the common unassisted training data. Starting from the same student parameters and optimizer state, a candidate branch performs a temporary update with the candidate group added to those data. Both updated models are then evaluated on the same independent, unassisted successful probe trajectories. The reduction in probe loss relative to the control defines the candidate's *post-update learning value*: an estimate of how much the candidate improves the student's likelihood of successful unassisted behavior beyond the common-data update. This comparison uses the student's training objective and optimizer state, so the score reflects the effect of incorporating the candidate into an actual learning step.

The reference updates are used for scoring and then discarded. FORGE restores the original student state and performs the training update with the common data and, from each failure event, at most the complete candidate group with the highest strictly positive learning value. Successful and failed trajectories within the selected group are retained, and student advantages continue to come from environment rewards.

The guidance policy learns from all eligible candidate interactions, using returns that combine learning value with interaction costs and invalid-action penalties. Feedback covers the initial guidance decision, retrieval, and follow-up interventions within each candidate. As the student changes, new failures supply further tutoring opportunities, and update-based feedback trains the guide to generate experience suited to the student's current learning needs. The final evaluation removes the guide and its experience library and measures the student's unassisted task performance.

The main contributions are:

- **Runtime guidance for all-failure GRPO groups.** FORGE trains a guidance agent within a harness that supports experience retrieval and adaptive intervention throughout student execution, making complete tutoring interactions the unit of guidance-policy learning.
- **Guidance feedback based on the effect of a student update.** Post-update learning value compares candidate and control updates on independent unassisted probes. It supplies a training signal for the guide and a criterion for selecting positive-value trajectory groups for the student.
- **Controlled evaluation of the resulting learning benefit.** Experiments on ALFWorld and WebShop evaluate unassisted performance and recovery of informative rollout groups. Component ablations and selection comparisons distinguish guidance-policy feedback from student data selection.

2 Related Work

Learning from difficult tasks in RLVR. Recent work on reinforcement learning with verifiable rewards addresses both how sampled experience is optimized and how informative experience is obtained. Dr. GRPO removes normalization terms that introduce response-length and question-difficulty biases (Liu et al., 2025); DAPO filters constant-reward groups and resamples to maintain informative training batches (Yu et al., 2025). Guidance methods expand the experience available when independent exploration is unsuccessful. LUFFY combines off-policy demonstrations with on-policy rollouts and uses policy shaping to balance imitation and exploration (Yan et al., 2025). Scaf-GRPO supplies hierarchical in-prompt scaffolds after learning stagnates, while NuRL injects offline self-generated hints into questions whose rollout groups entirely fail (Zhang et al., 2026b; Chen et al., 2026). These approaches recover learning opportunities by changing the training data or the conditions under which they are collected. FORGE follows the guidance-based direction and studies which complete, dynamically guided trajectory groups are useful for the student's subsequent update.

Learning guidance from student feedback. The feedback used to assess guidance determines what a guide learns to provide. HINT combines strategic meta-hints with affinity-aware policy optimization, weighting updates according to their effective update ratio and consistency (Wang et al., 2025). HiLL jointly trains a hinter and a reasoner, rewarding hints for creating informative rollout groups and reducing dependence on the hinted input (Xia et al., 2026). Its hints are generated before the reasoner's new rollouts, conditioned on a current failure and a reference solution.

Student improvement can also provide feedback for a teacher. SOAR uses a nested RL procedure in which a teacher generates synthetic problems and receives rewards based on the student's improvement on difficult target problems after training (Sundaram et al., 2026). FORGE shares this interest in the learning effect of teaching, but evaluates runtime guidance on the original task. Its feedback covers an entire tutoring interaction, including retrieval and follow-up decisions. Each candidate is scored by the probe-loss difference between a shared common-data update and an update that additionally includes the candidate group, starting from identical student parameters and optimizer state. The probe consists of independent, unassisted successful trajectories. This comparison estimates the candidate's incremental contribution to learning and supplies feedback for guidance-policy optimization.

Runtime harnesses and experience adaptation. Recent agent systems make external context and memory active components of task execution. Agentic Context Engineering (ACE) accumulates and refines contextual playbooks through generation, reflection, and curation (Zhang et al., 2026a). Memory-R1 trains memory-management and answer agents with outcome-driven RL (Yan et al., 2026). EvoHarness-RL brings runtime control into the policy's action space: environment actions are interleaved with operations over belief, progress, and experience stores, and GRPO trains their coordination under task and cost feedback (Ning et al., 2026). These systems motivate the use of structured external state and learned access decisions. In FORGE, the workspace and experience library support a separate guidance policy whose return depends on the student's post-update learning value. A fixed procedure maintains the library, while the guide learns retrieval and intervention decisions. The evaluation target is the student's performance with this auxiliary guidance system removed.

3 Methodology

3.1 Preliminary

We consider language agents acting in partially observed environments with binary terminal task rewards. The student policy is π_θ, and the guidance policy is μ_ϕ. A *clean* rollout uses the ordinary task observations and tools without auxiliary guidance or experience-library injection. For a task instance x, the student samples G clean rollouts from the same initial environment state. Let $z_i \in \{0, 1\}$ indicate whether rollout i completes the task. GRPO (Shao et al., 2024) assigns advantages

by comparing outcomes within the group:

$$A_i^R = \frac{z_i - \bar{z}}{\mathrm{std}(z_1, \ldots, z_G) + \epsilon_A}, \qquad \bar{z} = \frac{1}{G}\sum_{i=1}^{G} z_i. \tag{1}$$

All student-generated tokens in a rollout receive its group-relative task advantage. If every rollout fails, every task advantage is zero. The same holds for an all-success group: reward variation, rather than success alone, supplies the task-directed update.

Our student uses the clipped GRPO surrogate with global student-token normalization. For a collection of recorded trajectories $\mathcal{D}$, the loss is

$$\rho_{\tau,t}(\theta) = \frac{\pi_\theta(y_t \mid H_t)}{\pi_{\mathrm{old}}(y_t \mid H_t)},$$

$$L_{\mathrm{GRPO}}(\theta; \mathcal{D}) = \frac{1}{N_{\mathcal{D}}} \sum_{\tau \in \mathcal{D}} \sum_{t \in \mathcal{T}_R(\tau)} \ell^R_{\tau,t}(\theta), \tag{2}$$

$$\ell^R_{\tau,t}(\theta) = -\min\{\rho_{\tau,t} A^R_\tau,\ \mathrm{clip}(\rho_{\tau,t}, 1-\epsilon, 1+\epsilon) A^R_\tau\} + \beta_R \kappa_{\tau,t}(\theta).$$

Here H_t is the recorded history before token generation, $\mathcal{T}_R(\tau)$ contains only student-generated tokens, and $N_{\mathcal{D}}$ is their total count. The behavior policy π_{old} is the sampling snapshot; $\kappa_{\tau,t}$ is the sampled KL penalty against a separately frozen reference, defined in Appendix F. Uniform task rewards eliminate the clipped task-reward term, although KL regularization can still contribute to the update.

3.2 Why intervene on all-failure groups?

For a fixed policy and task x, let p_x be the clean success probability. With $G \geq 2$ independent rollouts,

$$\begin{aligned} P_{\mathrm{fail}}(x; G) &= (1 - p_x)^G, \\ P_{\mathrm{mix}}(x; G) &= 1 - (1 - p_x)^G - p_x^G. \end{aligned} \tag{3}$$

For small p_x, $P_{\mathrm{mix}} \approx G p_x$: obtaining one informative group requires approximately $1/(G p_x)$ clean groups under an unchanged policy (Xia et al., 2026).

After warm-up, FORGE intervenes on valid all-failure groups with probability p_{int}, generating M candidate tutoring groups. Let q_x be the conditional probability that a triggered event supplies an accepted mixed-outcome group. Then

$$P_{\mathrm{available}}(x) = P_{\mathrm{mix}}(x; G) + p_{\mathrm{int}}(1 - p_x)^G q_x. \tag{4}$$

The added term recovers training opportunities from all-failure events, which dominate when clean success is rare. Here q_x describes tutoring and selection, not an algorithmic score. Appendix A provides the derivation.

3.3 Runtime harness for adaptive tutoring

The runtime harness connects the guidance policy to an episode workspace, an experience-retrieval interface, and an intervention controller. It allows the guide to follow the student's execution and revise its assistance as new evidence arrives. The guide chooses retrieval, guidance content, strength, and follow-up actions; the harness maintains state, validates requests, and delivers interventions at legal execution boundaries.

Failure context and persistent experience. At the start of an event, the workspace contains all G failed clean trajectories, the public initial observation, evidence identifiers, and remaining budgets. The guide can retrieve relevant records from a versioned library of earlier training events. The library remains read-only during collection. A fixed archival procedure updates it at batch boundaries using public observations, legal suggestions, and linked outcomes; retrieval requests are learned policy actions, while archival construction and record ranking follow fixed rules.

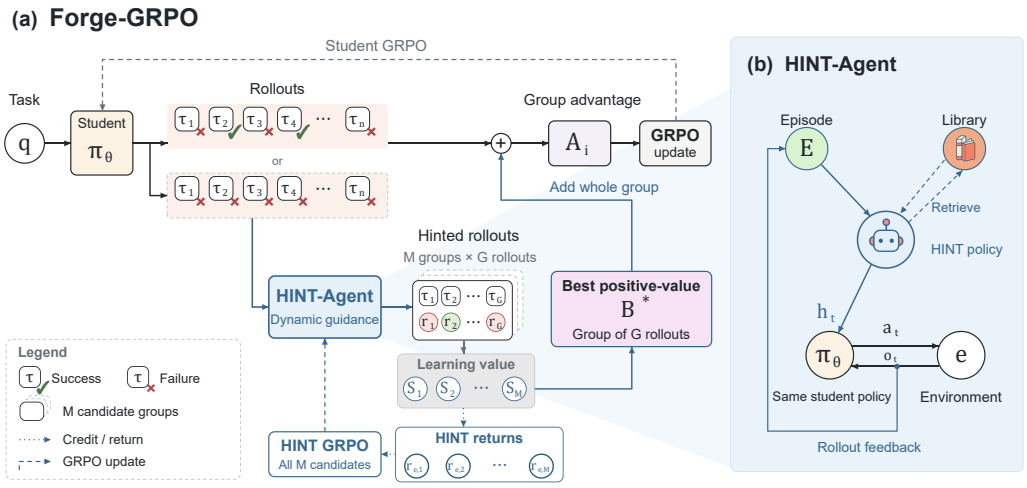

Figure 1: **Failure-triggered runtime guidance within GRPO.** A valid all-failure clean group initiates tutoring with probability p_{int}. The guide generates M candidate interactions, each started candidate containing G student rollouts with adaptive follow-up guidance. A workspace, experience retrieval, and intervention controller support these decisions. Learning-value feedback trains the guide and selects at most one positive-value group per event for the student's task-reward GRPO update. The student acts without guidance at evaluation time.

Each candidate has isolated evolving state. Once student execution starts, every trajectory maintains its own workspace containing its current observations, actions, delivered hints, and retrieval results. The original failure trajectories remain available throughout tutoring.

Guidance during execution. For each candidate, the guide first samples a shared initial hint or abstains. A started candidate launches G student executions from the original environment state, conditioned on that initial hint. Abstention retains the candidate slot and its incurred cost without launching student rollouts.

Subsequent guidance follows the realized history of each trajectory. At a decision point, the guide may make one `retrieve` request and then choose `hint`, `wait`, or `stop`. Hints specify a state check, subgoal, or constraint clarification, together with strength, evidence references, and a next-check rule. Waiting resumes student execution until a later checkpoint. Stopping ends assistance while allowing the student to continue to its terminal condition.

The controller checks for termination after an environment action returns and invokes the guide only when a scheduled interval or public error/repetition condition is reached. Thus, a follow-up can address a failed action or newly revealed precondition using evidence obtained during the current rollout. Budgets bound follow-up decisions, retrieval output, and delivered hint tokens; the action contract is specified in Appendix G.

Student and guide updates replay their actual interleaved histories. Each model scores only its own generated tokens. In particular, delivered hints condition the student's actions but are not student training targets. The shared initial guide call is counted once, followed by the distinct calls along individual trajectories. Appendix B states the interaction factorization.

3.4 Learning-value feedback for complete interactions

The guide is trained on complete tutoring interactions, including the initial decision, retrieval requests, and follow-up interventions. We evaluate their generated experience using *post-update learning value*: the change in a successful clean-trajectory probe loss caused by including one candidate group in a temporary student update.

Let C contain the valid clean groups in the current batch, and let $B_{e,j}$ be candidate j from event e. A control update uses C; a candidate update uses C plus the entire candidate group when its outcomes are mixed. Both apply one student optimizer step from identical saved parameters and optimizer state. Denote their temporary student parameters by θ_C^+ and $\theta_{e,j}^+$. We score them on the same event-specific probe $\mathcal{P}_e$:

$$S_{e,j}^{\text{ref}} = L_{\mathcal{P}_e}(\theta_C^+) - L_{\mathcal{P}_e}(\theta_{e,j}^+). \tag{5}$$

The probe consists of independently collected, verified successful clean trajectories, fixed before candidate generation and hidden from the guide. A positive score means that the candidate improves their likelihood relative to the common-data control. This is a local update-based feedback signal; its practical value for unassisted performance is evaluated through the reward ablation and end-to-end selection comparison.

Uniform-outcome candidates, initial abstentions, and invalid initial actions add no student data and receive zero evaluable reference value. An unavailable probe or incomplete required scoring marks the event unknown, excluding it from value-based selection and guide training. Appendix C gives the exact probe loss, token normalization, and optimizer-state handling.

3.5 GRPO UPDATES FOR THE STUDENT AND GUIDE

For each evaluable event, FORGE selects the complete mixed-outcome candidate with the largest strictly positive raw reference score; exact ties are broken by candidate index. If none is positive, the event contributes no tutoring data. The selected group retains all G trajectories, including successes, failures, and their recorded guidance histories. Writing j_e^* for the selected index, the actual student training data are

$$\mathcal{D}_{\text{student}} = C \uplus \biguplus_{e:j_e^* \neq \varnothing} B_{e,j_e^*}, \qquad L_{\text{actual}}(\theta) = L_{\text{GRPO}}(\theta; \mathcal{D}_{\text{student}}). \tag{6}$$

Here $\uplus$ denotes concatenation without dropping repeated observations. The student commits one joint optimizer update at the batch boundary, starting from the saved pre-update state. Learning value determines which tutoring data enter this update; student advantages continue to use terminal task rewards through Equation 1. Temporary reference updates are discarded after scoring.

The guide also uses a clipped group-relative policy objective. Each candidate receives a detached return comprising its reference value normalized by a pre-action historical scale, minus measured interaction costs and invalid-action penalties. Initial abstention incurs its cost, while an invalid initial action additionally incurs a format penalty. Returns are centered and standardized across the M candidates of the same event, and the resulting advantage is applied to all unique guide-generated tokens in that interaction. All eligible candidate slots train the guide, including unselected candidates and known zero-value groups. Appendix F provides the complete return and guide objective.

Collection fixes the student, guide, KL-reference, library, and scoring versions. Model updates and publication of new versions occur at complete batch boundaries. The guide remains frozen during scale calibration, while available raw reference scores can still select student groups; Appendix F describes calibration. At evaluation, the student acts independently without the guide or experience-library injection.

4 EXPERIMENTS

We evaluate FORGE on ALFWorld and WebShop, examining unassisted task performance, the recovery of informative rollout groups, and the contributions of guidance learning and runtime control. We denote FORGE combined with the student GRPO objective as **FORGE-GRPO**.

4.1 EXPERIMENTAL SETUP

Tasks and metrics. ALFWorld (Shridhar et al., 2020) evaluates six household task families in seen and unseen environments, with a 50-action budget per trajectory. We report per-family success and an overall rate pooling task outcomes across three runs. WebShop (Yao et al., 2022) evaluates shopping through search, navigation, and purchase actions. Task score is the mean environment

Table 1: Unassisted task success (%) of a Qwen2.5-7B-Instruct student across six task families in seen and unseen environments. Rates pool outcomes across three runs on the same evaluation tasks. All RL methods are evaluated at 50 training teps; the base model is evaluated before RL training.

Method	Pick	Look	Clean	Heat	Cool	Pick2	Overall
Seen environments							
Base model	42.86	38.46	24.69	12.50	4.00	5.56	22.14
GRPO	71.43	71.79	74.07	62.50	49.33	37.50	61.19
GRPO + Dynamic Sampling	69.52	76.92	80.25	66.67	53.33	41.67	64.29
GiGPO	94.29	82.05	86.42	77.08	66.67	45.83	76.43
HGPO	**99.05**	82.05	**88.89**	83.33	76.00	48.61	80.95
FORGE-GRPO	87.62	**84.62**	85.19	**91.67**	**84.00**	**56.94**	**81.43**
Unseen environments							
Base model	19.44	9.26	15.05	2.90	1.59	0.00	8.96
GRPO	65.28	37.04	53.76	53.62	39.68	37.25	49.25
GRPO + Dynamic Sampling	76.39	46.30	64.52	60.87	55.56	45.10	59.70
GiGPO	94.44	64.81	75.27	72.46	71.43	47.06	72.64
HGPO	**97.22**	68.52	**80.65**	79.71	76.19	50.98	77.36
FORGE-GRPO	83.33	**72.22**	76.34	**86.96**	**82.54**	**62.75**	**78.11**

Bold denotes the best result in each column within a split. Dynamic Sampling denotes filtering uniform-outcome groups and collecting additional groups to fill the training batch. All methods use the same task manifests and action budget; equal training iterations do not imply equal compute or interaction costs.

reward rescaled to 0–100; success is the percentage of tasks receiving the maximum reward. Binary task-success labels determine guidance triggers and group diagnostics. All evaluations disable the guide, auxiliary hints, and experience retrieval.

Models and comparisons. The student is Qwen2.5-7B-Instruct and the guide is Qwen2.5-3B-Instruct (Qwen et al., 2024), trained with rank-32 and rank-16 LoRA adapters (Hu et al., 2021), respectively, and AdamW at a learning rate of 0.00001. The default group size and candidate count are $G = 8$ and $M = 4$. Comparisons include the base model, GRPO (Shao et al., 2024), GRPO with dynamic sampling (DS), GiGPO (Feng et al., 2025), and HGPO (He et al., 2026). DS filters uniform-outcome groups and replenishes the batch following DAPO (Yu et al., 2025). Methods share evaluation tasks, interaction limits, and observation/history settings within each benchmark. We evaluate ALFWorld after 50 training iterations and WebShop after 150; the base model is evaluated before RL training.

Intervention schedule. After warm-up, an eligible all-failure group triggers tutoring with probability $p_{\text{int}} = 0.5$. An all-success task instance is skipped in the next task-sampling round. These rules control auxiliary sampling and repeated attempts on already successful instances; evaluation sets remain fixed. Training steps denote outer iterations, and equal iteration counts need not incur equal computation.

4.2 Unassisted Task Performance

Table 1 shows the strongest overall ALFWorld performance for FORGE-GRPO in both splits. Its advantage over HGPO concentrates on Heat, Cool, and Pick2, whose initial success rates are low. These families require learning from tasks that rarely succeed under the initial student, matching FORGE's focus on all-failure groups. The gains extend to unseen environments, indicating that the learned behavior remains useful beyond the training environments.

FORGE also improves over GRPO on every other displayed family. The comparison with HGPO is more selective: HGPO remains stronger on Pick and Clean. Thus, the overall improvement comes from substantial progress on initially difficult families alongside gains over GRPO elsewhere. These results measure progress at the reported checkpoints.

Table 2: **WebShop performance.** Unassisted task score and success rate (SR).

Method	Score	SR (%)
Base model	26.4	7.8
GRPO	79.3	66.1
GRPO + DS	81.7	70.2
GiGPO	88.9	77.6
HGPO	89.0	78.5
FORGE-GRPO	**89.8**	**81.0**

DS: dynamic sampling. Scores use a 0–100 scale. All evaluations disable guidance and retrieval.

Table 3: **Component ablations.** Unassisted means after 50 ALFWorld and 150 WebShop training steps.

	ALFWorld		WebShop	
Variant	Seen	Unseen	Score	SR
FORGE-GRPO	**81.4**	**78.1**	**89.8**	**81.0**
Frozen guide	76.5	72.3	83.0	77.9
Success reward	77.3	74.5	84.2	76.8
Initial hint only	67.2	61.8	79.9	72.3
No retrieval	74.6	69.4	81.2	74.5

Success rates are percentages; scores use a 0–100 scale. All variants retain positive-value student selection. Variant definitions appear in the text.

On WebShop, FORGE-GRPO obtains the highest task score and success rate (Table 2). Improvements cover both partial satisfaction of shopping requirements and complete task success, extending the benefit from household interaction to web shopping.

4.3 Guidance Learning and Runtime Control

Table 3 changes one component at a time while retaining reference scoring and the student's positive-value group-selection rule. Freezing the guide reduces performance in both environments, supporting adaptation of the guidance policy during student training. Replacing its learning-value feedback with candidate success rate also reduces performance, supporting update-based feedback for producing useful tutoring experience. This reward ablation retains interaction costs and invalid-action penalties.

Removing follow-up interventions produces the largest decline among the tested variants. Guidance that responds to later observations and mistakes therefore contributes beyond the initial hint. Disabling retrieval also reduces performance, supporting access to prior training experience. The initial-hint-only variant retains initial retrieval, while the no-retrieval variant retains the episode workspace.

4.4 Training Signal and Candidate Selection

We examine the recovery of informative clean groups during training and the effect of candidate selection on final unassisted performance.

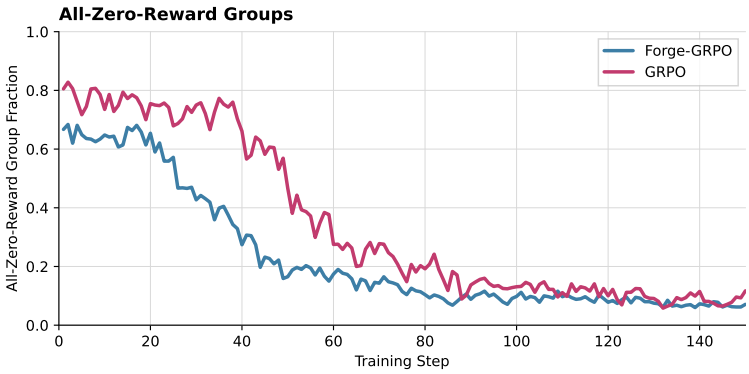

Figure 2: **Clean training groups on WebShop.** All-zero-reward group fractions before intervention, excluding all-success groups from the denominator.

Figure 2 shows that FORGE-GRPO reduces the all-failure fraction earlier in training. Within the monitored population, which excludes all-success groups, this corresponds to a larger fraction of mixed-outcome groups with nonzero task-reward advantages. The curves approach each other later as GRPO also recovers informative groups.

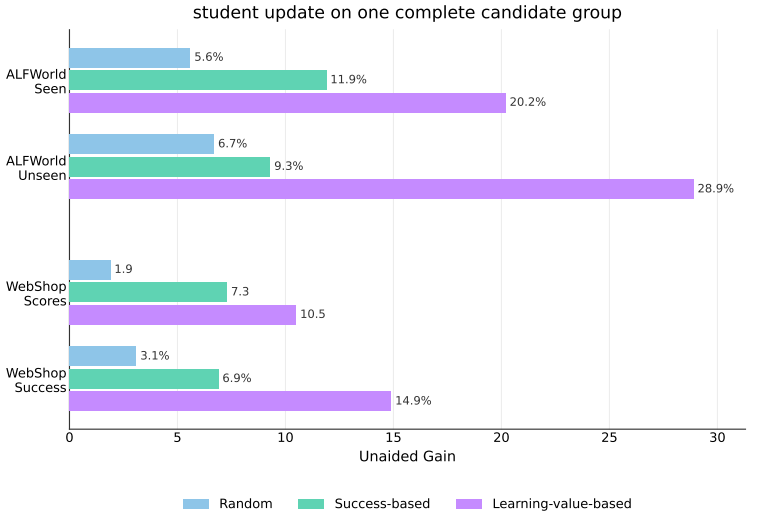

Figure 3: **Effect of candidate selection.** Final performance gains over GRPO under random, success-rate-based, and learning-value-based group selection. Gains are in percentage points for success rates and score points for WebShop score.

Figure 3 compares three rules for selecting a complete candidate group for student training. Learning-value selection achieves larger final gains than success-based and random selection across all four metrics. This comparison complements the guide-reward ablation by examining which generated experience is used for the student update.

5 Conclusion

FORGE integrates a trainable guidance agent into GRPO to address all-failure rollout groups. Its runtime harness supports adaptive interventions and experience retrieval, while post-update learning value supplies guide feedback and selects student training experience. Experiments on ALFWorld and WebShop show improved unassisted performance, with pronounced gains on initially difficult ALFWorld families and earlier recovery of informative clean groups on WebShop. Ablations and the selection comparison support the contributions of dynamic guidance, retrieval, and update-based feedback.

6 Limitations and Future Work

FORGE adds training cost through guided rollouts and temporary updates; equal-step gains therefore require further evaluation under matched compute budgets. Learning value measures local probe-loss changes, whose rankings may differ under joint updates and closed-loop execution. Future work will examine broader probes, adaptive intervention budgets, and generalization beyond ALFWorld and WebShop to other model scales and longer-horizon tasks.

AI USE STATEMENT

We used generative AI tools solely to edit author-written text for grammar, wording, clarity, and readability. We did not use generative AI to generate research ideas, hypotheses, methods, algorithms, proofs, code, experimental designs, data, results, analyses, figures, or citations. All AI-assisted edits were reviewed and verified by the authors. We take full responsibility for the final content of this work.

ETHICS STATEMENT

Experiments use ALFWorld and WebShop benchmark environments. The guidance interface operates on public task observations and recorded interaction histories. The experience library separates observed evidence from generated suggestions. Deployment in live services would require additional controls for authorization, privacy, and irreversible actions.

REPRODUCIBILITY STATEMENT

Section 4.1 specifies the models, optimization settings, evaluation metrics, and training checkpoints. Appendices A–G describe the probability analysis, recorded histories, reference updates, training algorithm, objectives, and guidance interface. All reported evaluations disable auxiliary guidance and retrieval. The ablation definitions and figure captions state the comparisons and measurement populations used in the reported experiments.

REFERENCES

Justin Chih-Yao Chen, Becky Xiangyu Peng, Prafulla Kumar Choubey, Kung-Hsiang Huang, Jiaxin Zhang, Mohit Bansal, and Chien-Sheng Wu. Nudging the Boundaries of LLM Reasoning. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2509.25666.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-Group Policy Optimization for LLM Agent Training. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2505.10978.

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*, 2026. URL https://arxiv.org/abs/2602.22817.

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. *arXiv preprint arXiv:2106.09685*, 2021. URL https://arxiv.org/abs/2106.09685.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-Like Training: A Critical Perspective. In *Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=5PAF7PAY2Y.

Xuying Ning, Dongqi Fu, Tianxin Wei, Hanqing Zeng, Yuanchen Bei, Bingxuan Li, et al. EvoHarness-RL: Learning Self-Evolving Runtime Harness for Long-Horizon LLM Agents. *arXiv preprint arXiv:2608.05446*, 2026. URL https://arxiv.org/abs/2608.05446v1.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, et al. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, et al. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. *arXiv preprint arXiv:2010.03768*, 2020. URL https://arxiv.org/abs/2010.03768.

Shobhita Sundaram, John Quan, Ariel Kwiatkowski, Kartik Ahuja, Yann Ollivier, and Julia Kempe. Teaching Models to Teach Themselves: Reasoning at the Edge of Learnability. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2601.18778.

Xinyi Wang, Jinyi Han, Zishang Jiang, Tingyun Li, Jiaqing Liang, Sihang Jiang, Zhaoqian Dai, Shuguang Ma, Fei Yu, and Yanghua Xiao. Don't Tell the Answer, Truly Guide the Reasoning During RL Rollouts. *arXiv preprint arXiv:2510.09388*, 2025. URL https://arxiv.org/abs/2510.09388v2.

Yu Xia, Canwen Xu, Zhewei Yao, Julian McAuley, and Yuxiong He. Learning to Hint for Reinforcement Learning. *arXiv preprint arXiv:2604.00698*, 2026. URL https://arxiv.org/abs/2604.00698v1.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to Reason under Off-Policy Guidance. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 117157–117186. Curran Associates, Inc., 2025. doi: 10.52202/085713-3904. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/a9d5c33e38442450f9781bd7bad3c81e-Paper-Conference.pdf.

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian Kersting, Jeff Z. Pan, Hinrich Schuetze, Volker Tresp, and Yunpu Ma. Memory-R1: Enhancing Large Language Model Agents to Manage and Utilize Memories via Reinforcement Learning. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12805–12825, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.583. URL https://aclanthology.org/2026.acl-long.583/.

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*, 2022. URL https://arxiv.org/abs/2207.01206.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, juncai liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An Open-Source LLM Reinforcement Learning System at Scale. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 113222–113244. Curran Associates, Inc., 2025. doi: 10.52202/085713-3775. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/a4277440d50f1f15d2cb4c14f7e0c0d2-Paper-Conference.pdf.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, Urmish Thakker, James Zou, and Kunle Olukotun. Agentic Context Engineering: Evolving Contexts for Self-Improving Language Models. In *International Conference on Learning Representations*, 2026a. URL https://arxiv.org/abs/2510.04618.

Xichen Zhang, Sitong Wu, Yinghao Zhu, Haoru Tan, Shaozuo Yu, Ziyi He, and Jiaya Jia. Scaf-GRPO: Scaffolded Group Relative Policy Optimization for Enhancing LLM Reasoning. In *International Conference on Learning Representations*, 2026b. URL https://arxiv.org/abs/2510.19807.