Agent-G^2: Gaussian Guidance for Agentic Reinforcement Learning

Zixuan Wang[1,2*] **Yanrui Miao**[1*] **Zhengxi Lu**[1] **Teng Pan**[1,2]
Yiwen Qiu[1] **Hongxing Li**[1] **Peng Qiu**[2] **Ruiqing Zhang**[2] **Yongliang Shen**[1†]
[1]Zhejiang University [2]Baidu Inc.
{wang.zixuan, syl}@zju.edu.cn

Abstract

Hint-based reinforcement learning addresses reward sparsity in long-horizon agentic tasks by retaining a prefix of an expert trajectory before each rollout, letting the policy explore from a state closer to success. Its effectiveness hinges on the guidance depth: how much of the trajectory to keep. Existing methods treat this depth as a deterministic scalar. Scheduled approaches share one value across samples and ignore per-task heterogeneity; per-sample probing estimates it separately at the cost of extra rollouts. We find that useful guidance occupies a band of depths whose informativeness profile is approximately Gaussian around the band center, rather than concentrating at a single optimal point. We propose Agent-G^2, a Gaussian guidance framework that draws the depth per task from a Gaussian whose center and spread are estimated online from rollouts already collected for policy optimization, requiring no probe rollouts or learned depth predictor. The center combines a global baseline with per-cluster difficulty, and the spread tracks within-cluster variance. We evaluate Agent-G^2 on ALFWorld and WebShop on Qwen2.5-1.5B / 7B-Instruct. Agent-G^2 outperforms the strongest hint-based, hint-free, and Aux-RL baselines on ALFWorld by 2.3 / 3.9 / 7.4 points at under one-third the rollout cost of per-sample probing. Code is available at https://github.com/ZJU-REAL/Agent-G2.

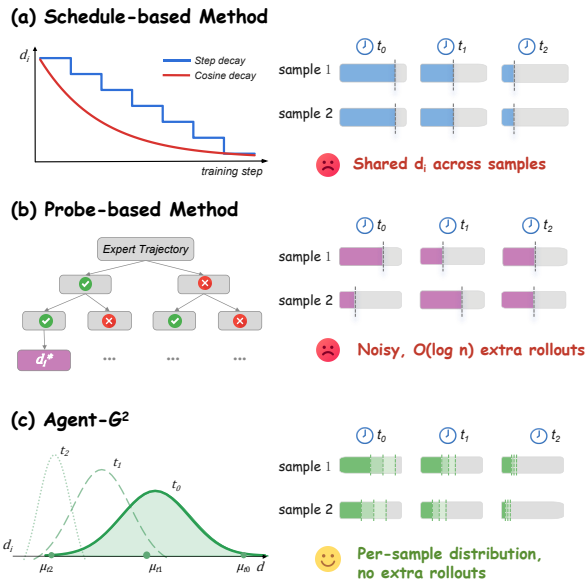

Figure 1: **Hint-based RL paradigms** across training steps t_0, t_1, t_2. **(a)** Schedule-based: shared d_t across samples. **(b)** Probe-based: $O(\log n)$ per-sample probes. **(c)** Agent-G^2 (ours): $d_i \sim \mathcal{N}(\mu_t, \sigma_t^2)$ per task, estimated online from existing rollouts.

1 Introduction

LLM agents trained with reinforcement learning have made progress on long-horizon decision-making tasks (Shen et al., 2023; Zhang et al., 2026b), including web navigation (Yao et al., 2022; Zhou et al., 2024a; Deng et al., 2023), embodied control (ichter et al., 2023; Huang et al., 2022; Wang et al., 2026b), and scientific experimentation (Boiko et al., 2023; Bran et al., 2023). A recurring obstacle is *reward sparsity*: tasks span dozens of sequential decisions, yet only a binary signal is issued at termination, and on-policy exploration from the initial state rarely reaches a successful terminal state. Hint-based RL mitigates this obstacle by retaining a prefix of an expert trajectory before each rollout, so the policy explores from a state closer to success (Xi et al., 2024a; Zhang et al., 2026a; Su et al., 2026), directly addressing *advantage collapse* in GRPO-style training (Zhang et al., 2025a; Wang et al., 2026a; Li et al., 2025).

The effectiveness of hint-based RL depends critically on the *guidance depth*: how much of the expert trajectory to retain. Too little guidance leaves few successful rollouts; too much saturates the reward and eliminates the contrast needed for advantage estimation. Despite this sensitivity, existing

[*] Equal contribution.
[†] Corresponding author.

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 6096–6111
October 24-29, 2026 ©2026 Association for Computational Linguistics

methods uniformly treat depth as a *deterministic scalar*, differing only in *how* they select it (Figure 1). Schedule-based methods (Xi et al., 2024a; Guo et al., 2026; Huang et al., 2026) derive one depth from the training step or batch-level feedback and share it across all samples, ignoring per-task heterogeneity. Per-sample methods (Su et al., 2026; Zhang et al., 2025a; Li et al., 2025; Zhang et al., 2026a) estimate a separate depth through binary search or enumeration, but pay $O(\log n)$ extra rollouts per sample or $N\times$ the rollout budget. Moreover, these methods have been developed almost exclusively on mathematical reasoning, where tasks share uniform structure; agentic tasks present a different challenge, as difficulty varies widely within a single batch (e.g., a two-step "Pick" vs. a twenty-step "Pick Two" in ALFWorld) (Tu et al., 2026; Lian et al., 2026b; Li et al., 2026; Chen et al., 2026).

Our diagnostic in Section 2 quantifies both failure modes: shared-depth schedulers place over 60% of assignments outside the informative band (Figure 2(a)), while per-sample probing reduces this mismatch only by spending extra rollouts that remain too noisy under a GRPO-matched budget (Figure 2(b)). Both families share a deeper assumption: that one optimal depth exists per task, and the goal is to pinpoint it. Our analysis in Section 2 reveals that this assumption is mismatched to the problem structure: informative depths form a *band* around the optimal point, with a training-signal profile that is unimodal, approximately symmetric, and well fit by a Gaussian (σ=0.22, R^2=0.92; Figure 3(b)). **The right guidance depth is not a point to be found, but a neighborhood to be covered.**

This observation motivates Agent-G^2, a Gaussian guidance framework for hint-based RL on long-horizon agentic tasks. Agent-G^2 models guidance depth as a per-task Gaussian whose center and spread are estimated online through a global-local decomposition: a global baseline tracks the policy's overall progress, while per-cluster statistics adjust the center by task difficulty and widen the spread to match within-cluster variance. For each task, one depth is drawn from its cluster's Gaussian and converted to a prefix length shared by all rollouts. The same rollouts that update the policy also refresh the Gaussian parameters, achieving per-task depth variation at no extra rollout cost.

We evaluate Agent-G^2 on ALFWorld and WebShop with Qwen2.5-1.5B-Instruct and Qwen2.5-7B-Instruct. On ALFWorld, Agent-G^2 reaches

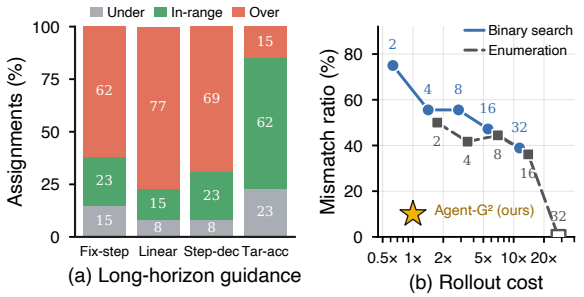

Figure 2: **Scalar-depth schedulers misallocate guidance.** (a) Shared-depth schedulers keep fewer than half of assignments in-range; (b) per-sample probing lowers the mismatch ratio ρ only by spending extra rollouts.

95.3% overall success at 1.5B and 98.4% at 7B, improving over the strongest hint-based RL baseline by 1.5 and 2.3 points and over the strongest hint-free RL baseline by 3.9 points at both scales. On WebShop, Agent-G^2 obtains a 92.3 reward score at both scales, with 78.9% and 84.4% final-purchase success. These gains hold across benchmarks, model scales, and horizon groups, without the probe-rollout overhead of per-sample search.

In summary, our contributions are as follows:

- We reveal the *neighborhood structure* of guidance depth: informative depths form a band with approximately Gaussian training-signal profile, explaining the structural failure modes of both shared-depth schedulers and per-sample probing.
- We propose Agent-G^2, a Gaussian guidance framework that estimates the per-task depth distribution online from rollout statistics via a global-local decomposition, requiring no probe rollouts or learned depth predictor.
- On ALFWorld and WebShop with Qwen2.5-1.5B / 7B-Instruct, we show that Agent-G^2 consistently outperforms the strongest non-probing hint-based, hint-free, and Aux-RL baselines on both benchmarks, at under one-third the rollout cost of per-sample probing.

2 Preliminary Analysis

We diagnose guidance-depth assignment on Qwen2.5-1.5B-Instruct / ALFWorld at step 50, where the policy's no-hint success rate is near 50% and the spread of per-task informative depths is widest. We sweep a depth grid $\mathcal{D}$ and estimate $p_i(d)$ from 32 rollouts per (task, depth) pair. The in-range set of task i is $\mathcal{R}_i = \{d \in \mathcal{D} : p_i(d) \in [0.4, 0.6]\}$, where training is most informative (Florensa et al., 2018); an assignment d_i is under-guided

if $p_i(d_i) < 0.4$ and over-guided if $p_i(d_i) > 0.6$. The mismatch ratio ρ is the fraction of tasks with $d_i \notin \mathcal{R}_i$. Full details appear in Appendix A.

Mismatch in Shared-Depth Scheduling Shared-depth schedulers set one guidance depth from the training step or batch-level feedback and apply it to all tasks in a batch. While avoiding additional probe rollouts, this scalar assignment ignores task-level heterogeneity. Figure 2(a) shows that among the shared-depth schedulers we test, the three step-based schedulers place only 15%–23% of rollout groups inside the in-range band, with most assignments being over-guided. Even the best shared scheduler, Target-acc, leaves 38% of its assignments outside the in-range band. This failure is structural rather than a tuning issue: a single depth cannot match tasks with different in-range bands.

Cost-precision Trade-off of Per-sample Probing Per-sample probing samples rollouts at candidate depths and selects the depth whose estimated success rate $\hat{p}_i$ is closest to 0.5, removing the shared-depth assumption at the cost of additional probe rollouts. Figure 2(b) quantifies this cost-precision trade-off. With M_{probe}=2 rollouts per candidate depth, Binary Search still leaves 75% of assignments mismatched. Enumeration is more accurate but costlier: M_{probe}=4 reaches 47% mismatch at 2× the GRPO rollout budget, while near-zero mismatch requires M_{probe}=32 at 20× the budget. Thus, methods that try to pinpoint a single depth from noisy rollout estimates must trade additional rollout cost for selection error.

Modeling Depth as a Gaussian. The two failure modes above stem from a shared premise: that one optimal depth exists per task. We now show that useful guidance has a richer structure.

Observation 1: Useful guidance occupies a band. Figure 3(a) plots $p_i(d)$ against d, with the under-guided, in-range, and over-guided regimes shaded. The in-range regime spans multiple neighboring depths rather than a single value, indicating that several nearby depths can keep rollouts in the just-learnable regime. Thus, a scheduler that localizes one exact depth ignores a neighborhood of informative choices.

Observation 2: Informative signal is Gaussian-like. We define $d_i^\star = \arg\min_{d\in\mathcal{D}} |p_i(d) - 0.5|$ and align task-depth pairs by $\Delta d = d - d_i^\star$. As a proxy for training informativeness we use the Bernoulli variance $p_i(d)(1 - p_i(d))$, which peaks at p_i=0.5 and

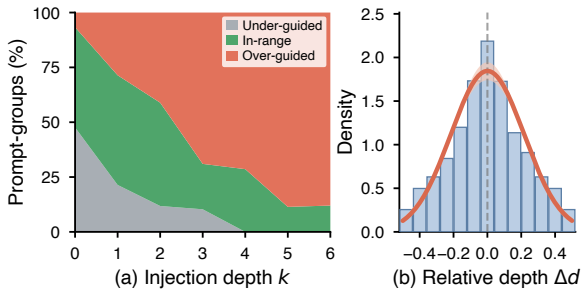

Figure 3: **Useful guidance forms a band around $d_i^\star$.** (a) Rollout success rate $p_i(d)$ vs. depth, with under-/in-range/over-guided regimes shaded; the in-range band spans several depths. (b) Bernoulli-variance informativeness aligned by $\Delta d = d - d_i^\star$, with a Gaussian fit (σ=0.22, R^2=0.92).

drops to 0 at deterministic rollouts. The peak at $\Delta d = 0$ follows from the definition of $d_i^\star$, but the symmetry and decay along Δd do not: they reflect how $p_i(d)$ varies with depth. The aligned profile (Figure 3(b)) is unimodal and approximately symmetric, with a Gaussian fit yielding σ=0.22 and R^2=0.92. The Gaussian provides the two parameters our sampler needs: a center μ for the band location and a spread σ for its width. Both map onto the mean and variance of batch rollouts, so the sampler in Section 3 runs online without extra probing. We discuss the choice of parametric family in Section A.

3 Method: Agent-G^2

Section 2 showed that scalar-depth schedulers either ignore per-task heterogeneity or require costly per-sample probing, while informative depths form Gaussian-like profiles around task-specific effective depths. Agent-G^2 translates these findings into a practical training framework (Figure 4): an adaptive Gaussian schedule derives per-task distributions from existing rollout statistics (Section 3.2), sampling draws one depth per task (Section 3.3), and the same rollouts update both the policy and the schedule (Section 3.4).

3.1 Problem Formulation.

We consider LLM agents trained with reinforcement learning on long-horizon tasks with reward only at termination. Each task i has a natural-language instruction q_i and one expert trajectory $\tau_i^\star = (a_{i,1}^\star, \ldots, a_{i,L_i}^\star)$ of length L_i. A hint scheduler selects a guidance ratio $r_i \in [0, 1]$, which is converted to a prefix length $n_i = \min(\lceil r_i L_i \rceil, L_i - 1)$. The first n_i expert actions

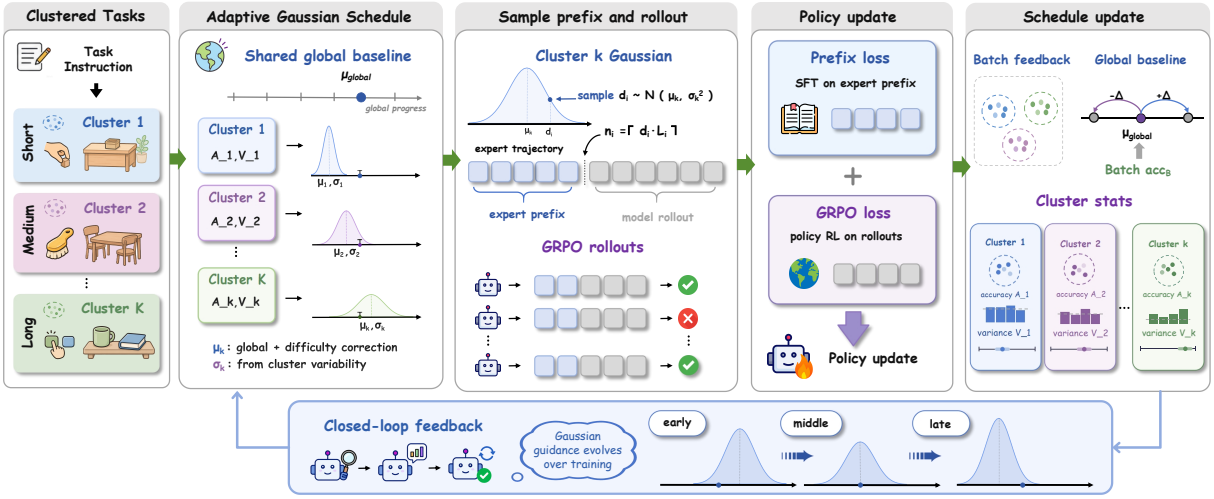

Figure 4: **Agent-G^2 pipeline.** **(1)** Tasks are clustered offline by difficulty. **(2)** A global baseline μ_{global} and per-cluster statistics (A_k, V_k) form a per-task Gaussian $\mathcal{N}(\mu_i, \sigma_i^2)$. **(3)** One prefix length is drawn per task; R rollouts run from the post-prefix state. **(4)** The policy is updated with prefix SFT plus GRPO loss. **(5)** The same rollouts refresh $\mu_{\text{global}}, A_k, V_k$ for the next batch.

are executed in the environment to reach a post-prefix state, from which the policy π_θ runs R independent rollouts. Each rollout $j \in \{1, \ldots, R\}$ terminates with a binary reward $y_{i,j} \in \{0, 1\}$, and $\hat{p}_i = \frac{1}{R}\sum_{j=1}^{R} y_{i,j}$ denotes the empirical success rate. We optimize π_θ with GRPO (Shao et al., 2024), which computes a group-normalized advantage from the R terminal rewards within each task.

3.2 Adaptive Gaussian Schedule

Agent-G^2 assigns each task i a Gaussian distribution $\mathcal{N}(\mu_i, \sigma_i^2)$ over the guidance ratio r_i. Its center and spread are computed online from rollouts already collected for policy optimization. The scheduler maintains three types of state: a global guidance baseline μ_{global} that tracks the overall guidance level, and for each offline cluster $\mathcal{C}_k$, EMA estimates A_k and V_k that summarize the level and dispersion of empirical success rates. We use the standard midpoint target $p_{\text{target}} = 0.5$ for binary success feedback, following the curriculum principle that training is most informative near the success–failure boundary (Florensa et al., 2018; Zhang et al., 2026a). All schedule states are initialized from weak priors that only determine the first-batch schedule, then continuously refreshed by rollout statistics.

Global baseline. The global baseline $\mu_{\text{global}} \in [0, 1]$ tracks the guidance level required by the current policy. For a training batch $\mathcal{B}$ with average success rate $\text{acc}_{\mathcal{B}} = \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} \hat{p}_i$, we shift μ_{global} by Δ in the direction that drives $\text{acc}_{\mathcal{B}}$ toward p_{target}:

$$\mu_{\text{global}} \leftarrow \text{clip}\Big(\mu_{\text{global}} + \text{sign}(p_{\text{target}} - \text{acc}_{\mathcal{B}})\,\Delta,\ 0,\ 1\Big). \tag{1}$$

Low batch success raises the baseline for deeper next-batch prefixes; high success lowers it.

Cluster statistics. The training set is partitioned offline into K clusters $\{\mathcal{C}_k\}_{k=1}^{K}$ by each task's expert-trajectory length, which serves as a simple proxy for separating tasks of different difficulty levels, and $k(i)$ denotes the cluster of task i. For each non-empty $\mathcal{B}_k = \mathcal{B} \cap \mathcal{C}_k$, the batch produces

$$\begin{aligned} \bar{p}_{\mathcal{B}_k} &= \frac{1}{|\mathcal{B}_k|}\sum_{i\in\mathcal{B}_k} \hat{p}_i, \\ v_{\mathcal{B}_k} &= \frac{1}{|\mathcal{B}_k|}\sum_{i\in\mathcal{B}_k} (\hat{p}_i - \bar{p}_{\mathcal{B}_k})^2, \end{aligned} \tag{2}$$

which are folded into A_k and V_k by EMA:

$$\begin{aligned} A_k &\leftarrow (1-\alpha)A_k + \alpha\bar{p}_{\mathcal{B}_k}, \\ V_k &\leftarrow (1-\alpha)V_k + \alpha v_{\mathcal{B}_k}. \end{aligned} \tag{3}$$

Clusters absent from the batch are left unchanged. A_k estimates the current success level of cluster k, while V_k measures how unevenly tasks in the cluster respond to the current policy.

Per-task distribution. The center and spread for task i in cluster $k{=}k(i)$ are

$$\begin{aligned} \mu_i &= \text{clip}(\mu_{\text{global}} + \lambda(p_{\text{target}} - A_k),\ 0,\ 1), \\ \sigma_i &= \max(\gamma V_k,\ \sigma_{\min}). \end{aligned} \tag{4}$$

A lower cluster success rate A_k shifts μ_i toward deeper guidance; a higher within-cluster variance V_k widens σ_i, giving the sampler broader coverage of the informative band identified in Section 2.

3.3 Per-Task Sampling and Rollout

Given (μ_i, σ_i) from the schedule, we draw one guidance ratio per task:

$$z_i \sim \mathcal{N}(\mu_i, \sigma_i^2), \qquad r_i = \mathrm{clip}(z_i,\, 0,\, 1). \tag{5}$$

The sampled ratio determines the prefix length n_i, executed once to reach the post-prefix state shared by all R rollouts of task i (Section 3.1). Since (μ_i, σ_i) are induced by the statistics of cluster $k(i)$, tasks in the same cluster share Gaussian parameters, while independent sampling yields task-level depth variation without extra rollouts.

3.4 Training

Each batch $\mathcal{B}$ closes the loop between policy optimization and schedule adaptation: rollouts generated from sampled prefixes drive both updates.

Policy update. We combine the GRPO loss $\mathcal{L}_{\mathrm{GRPO}}(\mathcal{B})$ with a teacher-forced loss on sampled expert prefixes, following common practice in hint-based RL (Zhang et al., 2025a; Huang et al., 2026):

$$\mathcal{L}_{\mathrm{aux}}(\mathcal{B}) = -\sum_{i\in\mathcal{B}} \sum_{t=1}^{n_i} \log \pi_\theta(a^\star_{i,t} \mid s_{i,t}, q_i), \tag{6}$$

where $s_{i,t}$ is the state reached after the first $t-1$ expert actions. The full objective is

$$\mathcal{L}(\mathcal{B}) = \mathcal{L}_{\mathrm{GRPO}}(\mathcal{B}) + \eta\, \mathcal{L}_{\mathrm{aux}}(\mathcal{B}), \tag{7}$$

with η controlling the auxiliary weight. The auxiliary term preserves imitation supervision on the sampled prefixes and is ablated in Section 4.4.

Schedule update. The terminal rewards from the same rollouts refresh $\mu_{\mathrm{global}}, A_k, V_k$ for the next batch as specified in Section 3.2. Schedule adaptation therefore reuses rollouts already collected for policy optimization and introduces no probe rollouts or learned depth predictor. Algorithm 1 summarizes the full per-batch loop.

Algorithm 1 Agent-G^2 per-batch training loop.

Require: Policy π_θ, expert trajectories $\{\tau_i^\star\}$, clusters $\{\mathcal{C}_k\}_{k=1}^K$
Require: Hyperparameters $\Delta, \alpha, \lambda, \gamma, \sigma_{\min}, \eta, R$; $p_{\mathrm{target}} = 0.5$
Initialize $\mu_{\mathrm{global}} \leftarrow 0.8$, $A_k \leftarrow 0$, $V_k \leftarrow 0$ ▷ weak priors
for each batch $\mathcal{B}$ **do**
  ▷ *Per-task schedule, sampling, and prefix execution*
  for each task $i \in \mathcal{B}$ with $k = k(i)$ **do**
    $\mu_i \leftarrow \mathrm{clip}(\mu_{\mathrm{global}} + \lambda(p_{\mathrm{target}} - A_k), 0, 1)$
    $\sigma_i \leftarrow \max(\gamma V_k, \sigma_{\min})$
    $z_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$; $r_i \leftarrow \mathrm{clip}(z_i, 0, 1)$
    $n_i \leftarrow \min(\lceil r_i L_i \rceil, L_i - 1)$
    Execute $\tau^\star_{i,1:n_i}$ to reach state s_i
    Record $\tau^\star_{i,1:n_i}$ for $\mathcal{L}_{\mathrm{aux}}$
    for $j = 1, \ldots, R$ **do**
      Roll out π_θ from s_i; record $y_{i,j}$
    end for
  end for
  ▷ *Policy update*
  Update π_θ: minimize $\mathcal{L}_{\mathrm{GRPO}}(\mathcal{B}) + \eta \mathcal{L}_{\mathrm{aux}}(\mathcal{B})$
  ▷ *Schedule feedback*
  $\hat{p}_i \leftarrow \frac{1}{R}\sum_j y_{i,j}$ for $i \in \mathcal{B}$
  $\mathrm{acc}_{\mathcal{B}} \leftarrow \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} \hat{p}_i$
  Refresh $A_k, V_k, \mu_{\mathrm{global}}$ (Section 3.2)
end for
return π_θ

4 Experiments

4.1 Setup

Benchmarks. We evaluate Agent-G^2 on two long-horizon agent benchmarks. ALFWorld (Shridhar et al., 2021) is a text-based embodied environment where the agent completes household tasks through multi-step interaction with sparse terminal rewards. We follow the standard six task types and group them by horizon into *Short* (Pick, Look), *Medium* (Clean, Heat, Cool), and *Long* (Pick2). WebShop (Yao et al., 2022) is a web navigation environment where the agent searches, filters, and purchases products matching a natural-language specification, with reward issued at final purchase.

We compare Agent-G^2 against five families. **(1) Prompting**: ReAct (Yao et al., 2023) and Reflexion (Shinn et al., 2023) on the same base, with GPT-4o (OpenAI et al., 2024) and Gemini-2.5-Pro (Comanici et al., 2025) as closed-source references. **(2) Imitation**: Full SFT, supervised fine-tuning on entire expert trajectories. **(3) RL without hints**: GRPO (Shao et al., 2024), GiGPO (Feng et al., 2025), and BEACON (Wang et al., 2026c), trained with sparse terminal rewards and no expert prefix. **(4) Aux-RL**: ETO (Song et al., 2024) and RLVMR (Zhang et al., 2026d), which add auxiliary supervision beyond the terminal reward. **(5) Hint-based RL**, subdivided into: *schedule-based* methods (Linear, Cosine, Step decay, Target acc); *search-based* methods (Binary Search, Enumeration); and *end-to-end* methods (StepHint (Zhang et al., 2026a), TraPO (Su et al., 2026)). All schedule- and search-based baselines share Agent-G^2's GRPO backbone, isolating prefix-depth allocation as the only variable.

Implementation. We instantiate Agent-G^2 on Qwen2.5-1.5B / 7B-Instruct (Qwen et al.,

Table 1: **Main results on ALFWorld and WebShop** (success rate, %). **Best** and second-best are highlighted.

Type	Method	ALFWorld							WebShop	
		Short		Medium			Long	All	Score	Succ
		Pick	Look	Clean	Heat	Cool	Pick2			
Closed-Source Models										
Prompting	GPT-4o (OpenAI et al., 2024)	75.3	60.8	31.2	56.7	21.6	49.8	48.0	31.8	23.7
Prompting	Gemini-2.5-Pro (Comanici et al., 2025)	92.8	63.3	62.1	69.0	26.6	58.7	60.3	42.5	35.9
Base: Qwen2.5-1.5B-Instruct										
Prompting	Direct Prompt (Qwen et al., 2025)	5.9	5.5	3.3	9.7	4.2	0.0	4.1	23.1	5.2
Prompting	ReAct (Yao et al., 2023)	17.4	20.5	15.7	6.2	7.7	2.0	12.8	40.1	11.3
Prompting	Reflexion (Shinn et al., 2023)	35.3	22.2	21.7	13.6	19.4	3.7	21.8	55.8	21.9
SFT	Full SFT	68.8	70.6	61.9	71.4	42.9	28.0	56.3	86.6	69.5
RL	GRPO (Shao et al., 2024)	85.3	53.7	84.5	78.2	59.7	53.5	72.8	75.8	56.8
RL	GiGPO (Feng et al., 2025)	96.0	76.5	91.8	91.3	71.7	79.5	86.1	83.1	65.0
RL	BEACON (Wang et al., 2026c)	**100**	88.2	86.7	**100**	78.9	92.9	91.4	86.1	75.6
Aux-RL	ETO (Song et al., 2024)	73.6	46.3	66.2	68.3	62.8	55.6	66.4	–	–
Aux-RL	RLVMR (Zhang et al., 2026d)	95.2	78.8	91.2	90.2	83.9	77.6	87.9	–	–
Hint-RL	Linear decay	92.5	83.3	90.0	66.6	71.4	82.6	83.6	88.9	74.2
Hint-RL	Cosine decay	97.6	70.0	83.3	58.3	85.0	76.2	83.6	83.2	73.4
Hint-RL	Step decay	85.0	**100**	95.0	83.3	95.2	86.9	89.8	89.1	74.2
Hint-RL	Target acc	**100**	83.3	**100**	**100**	90.5	82.6	93.8	86.4	75.0
Hint-RL	Enumeration	**100**	81.8	**100**	68.8	80.0	61.1	86.0	90.1	78.1
Hint-RL	Binary Search (Zhang et al., 2025a)	83.3	75.0	90.9	78.6	**95.7**	81.5	85.2	89.7	77.3
Hint-RL	StepHint (Zhang et al., 2026a)	86.0	60.0	84.9	63.7	75.0	65.2	77.3	87.6	71.9
Hint-RL	TRAPO (Su et al., 2026)	79.2	50.0	85.2	56.3	54.8	18.2	59.4	89.4	75.8
Hint-RL	**Agent-G^2 (Ours)**	96.8	**100**	**100**	92.9	84.2	**94.7**	**95.3**	**92.3**	**78.9**
Base: Qwen2.5-7B-Instruct										
Prompting	ReAct (Yao et al., 2023)	48.5	35.4	34.3	13.2	18.2	17.6	31.2	46.2	19.5
SFT	Full SFT	75.0	**100**	80.0	43.8	59.1	37.8	64.9	88.3	79.7
RL	GRPO (Shao et al., 2024)	90.8	66.1	89.3	74.7	72.5	64.7	77.6	79.3	66.1
RL	GiGPO (Feng et al., 2025)	91.8	88.6	95.9	90.2	86.5	85.2	90.2	84.4	72.8
RL	BEACON (Wang et al., 2026c)	**100**	81.8	96.3	92.9	94.7	90.0	94.5	87.7	79.7
Aux-RL	ETO (Song et al., 2024)	88.2	70.5	82.3	83.6	71.0	51.2	74.2	–	–
Aux-RL	RLVMR (Zhang et al., 2026d)	95.3	88.2	90.1	92.4	89.8	86.7	91.8	–	–
Hint-RL	Linear decay	97.5	92.9	66.7	85.2	87.5	92.0	85.2	91.1	82.8
Hint-RL	Cosine decay	97.1	**100**	78.9	85.7	83.3	90.0	89.4	90.1	82.0
Hint-RL	Step decay	95.0	92.9	90.5	91.7	**100**	80.0	91.4	92.1	83.6
Hint-RL	Target acc	97.5	**100**	95.2	83.3	81.3	92.0	92.9	89.3	79.7
Hint-RL	Enumeration	**100**	**100**	96.4	**100**	90.5	86.7	96.1	**96.0**	**89.8**
Hint-RL	Binary Search (Zhang et al., 2025a)	97.6	**100**	96.3	73.7	84.6	**93.8**	92.2	91.2	83.6
Hint-RL	StepHint (Zhang et al., 2026a)	97.3	**100**	89.4	94.1	79.1	91.7	91.4	90.9	82.0
Hint-RL	TRAPO (Su et al., 2026)	89.3	66.7	88.5	90.0	68.8	44.8	75.0	90.6	77.3
Hint-RL	**Agent-G^2 (Ours)**	**100**	**100**	**100**	**100**	**100**	91.7	**98.4**	92.3	84.4

2025). Agent-G^2-specific hyperparameters (Δ=0.1, α=0.2, γ=1.0) are fixed across both benchmarks without per-task tuning. We report success rate (%) on the held-out test set averaged over 5 seeds; full configurations appear in Appendix B.

4.2 Main Results

Overall performance. On ALFWorld, Agent-G^2 reaches 95.3% (1.5B) and 98.4% (7B). It outperforms RLVMR, the strongest method using auxiliary supervision, by +7.4 / +6.6, and Enumeration, the strongest probe-based baseline, by +9.3 / +2.3, without any auxiliary network or probe rollouts. On WebShop, Agent-G^2 is the strongest non-probing method at both scales (92.3 reward score, 78.9% / 84.4% purchase success), and at 1.5B also outperforms Enumeration by +2.2 reward-score points.

Cross-scale robustness. Agent-G^2 remains effective as the backbone scales from 1.5B to 7B. On ALFWorld, the margin over the strongest hint-based baseline grows from +1.5 to +2.3, so finer scheduling does not become redundant on stronger backbones. On WebShop, purchase success climbs from 78.9% to 84.4%, consistent with stronger action precision at scale. The 1.5B model also surpasses all 7B non-probing baselines, indicating that schedule design substitutes for backbone scaling rather than only compensating for limited capacity.

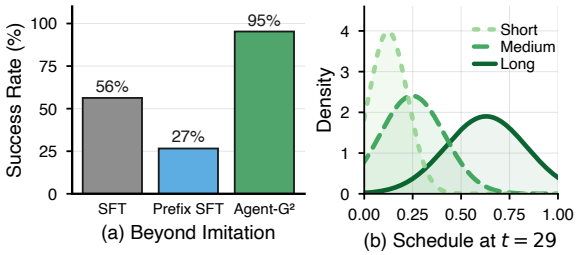

Figure 5: **Beyond imitation.** (a) ALFWorld success on Qwen2.5-1.5B: Agent-G^2 (95%) far exceeds Full SFT (56%) and Sampled-Prefix SFT (27%). (b) Per-cluster (Short / Medium / Long) depth densities at t=29.

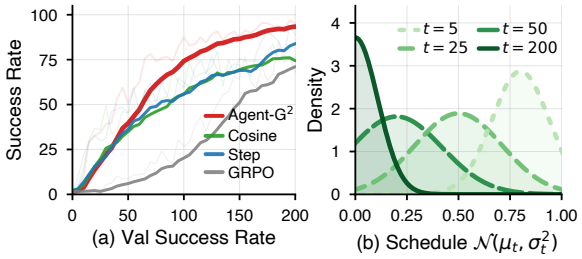

Figure 6: **Training dynamics of Agent-G^2.** (a) Success rate vs. gradient steps for Agent-G^2 and the strongest scalar-depth baseline. (b) Evolution of the per-task guidance-depth distribution across training.

4.3 Analysis

Beyond imitation. The gain of Agent-G^2 cannot be explained by imitation alone. Full SFT reaches 56.3%, and Sampled-Prefix SFT, which trains only on the prefix supervision pairs collected during Agent-G^2 training, reaches 26.6%; both are far below Agent-G^2's 95.3% (Figure 5(a)). The 68.4-point gap between Sampled-Prefix SFT and Agent-G^2 isolates the contribution of post-prefix RL: the missing ingredient is not copying expert tokens, but learning from sampled post-prefix states.

Heterogeneous depths in the same batch. Agent-G^2 gains most when samples in the same batch require different prefix depths. In Figure 6(a), the convergence gap is largest between steps 50 and 150, the window in which the depth distribution in Figure 6(b) is widest. A scalar d^{sched} is brittle here: a single depth is too deep for easy samples and too shallow for hard ones, pushing most rollouts outside the informative band. Agent-G^2 samples a neighborhood of depths instead, so both ends of the difficulty spectrum receive usable guidance.

Schedule emerges from the rollouts. The shape of the distribution in Figure 6(b) is set by training itself. At t=5 it favors deep prefixes, because the policy has yet to reach learnable states on its own. Around t=25–50 it widens, as the cluster variance V_k peaks while easy and hard clusters separate. By t=200 it concentrates near zero, because the success rate has cleared the threshold of Equation (1) on every cluster. The mean follows global and cluster success, the spread follows V_k, and both come from rollouts already collected for policy optimization. Figure 7 in Appendix B samples this pattern across four training phases, with the Long cluster retaining a wider σ_k for longer.

Table 2: **Per-step training cost on Qwen2.5-1.5B / ALFWorld.** Median wall-clock seconds per gradient step. Ratios are relative to Agent-G^2.

Method	Type	Cost/step (s)	Ratio
Agent-G^2	Distribution	88	1.00×
Step decay	Schedule	57	0.65×
Cosine decay	Schedule	60	0.68×
Linear decay	Schedule	61	0.69×
Target acc	Schedule	80	0.91×
Enumeration	Probe	425	4.83×
Binary Search	Probe	285	3.24×

Training efficiency. We assess efficiency by convergence speed and per-step wall-clock cost. Figure 6(a) shows that Agent-G^2 reaches the final accuracy of scheduled baselines in roughly half the gradient steps. Although Agent-G^2 incurs a modest per-step overhead over scheduled methods (88s vs. 57–80s), it is substantially cheaper than probing-based methods (285–425s, or 3.24–4.83× higher), which spend extra rollouts to estimate depths. Therefore, by combining faster convergence with much lower cost than probing, Agent-G^2 achieves the best overall training efficiency.

Decomposing the learned schedule. Figure 5(b) decomposes the learned schedule at t=29 by horizon group. These patterns are not manually specified: each cluster only maintains rollout-derived statistics A_k and V_k, and the separation emerges from training feedback.

4.4 Ablation Studies

Sampling drives the gain. Replacing the Gaussian draw $d_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$ with the deterministic mean $d_i = \mu_i$ removes stochastic depth coverage and drops the overall success rate from 95.3% to 89.8%. This shows that a single-depth estimate cannot cover the informative band. Replacing the Gaussian with a variance-matched uniform distri-

Table 3: **Ablation of Agent-G^2** on ALFWorld / Qwen2.5-1.5B-Instruct. Δ is the change in All relative to the full method. Removing $\mathcal{L}_{\text{GRPO}}$ is equivalent to Sampled-Prefix SFT.

Variant	Long	All	Δ
Agent-G^2	94.7	95.3	–
Sampling form			
w/o sampling: $d_{i,j}{=}\mu_i, \sigma_i{=}0$	84.6	89.8	$\downarrow$ 5.5
Uniform sampling: $\mathcal{U}[\mu_i \pm \sigma_i]$	80.8	88.3	$\downarrow$ 7.0
Adaptive moments			
w/o adaptive center: $\mu_i{=}\mu_{\text{global}}$	79.2	91.4	$\downarrow$ 3.9
w/o adaptive spread: $\sigma_i{=}\sigma_{\min}$	78.3	93.8	$\downarrow$ 1.5
w/o grouping: $K{=}1$	65.4	89.1	$\downarrow$ 6.2
Loss components			
w/o $\mathcal{L}_{\text{aux}}$	76.7	86.7	$\downarrow$ 8.6
w/o $\mathcal{L}_{\text{GRPO}}$	10.0	26.6	$\downarrow$ 68.7

bution reduces performance to 88.3%, confirming that the gain comes from coverage of the informative band, not the exact distributional shape.

Both center and spread matter. Removing the cluster-dependent center correction by setting $\mu_i = \mu_{\text{global}}$ lowers the overall success rate to 91.4%, showing that a global baseline alone cannot capture difficulty variation across clusters. Removing the adaptive spread by setting $\sigma_i = \sigma_{\min}$ reduces performance to 93.8%, with a pronounced drop on Long tasks (94.7% $\rightarrow$ 78.3%), indicating that dynamically adapting σ_i is important for maintaining sufficient depth coverage, especially on hard long-horizon tasks. Collapsing all tasks into a single cluster ($K = 1$) drops the overall success rate to 89.1%, with the largest degradation on Long tasks, decreasing by 29.3 points. Together, these results show that both cluster-aware centering and adaptive spreading are necessary for handling heterogeneous task difficulty.

Auxiliary loss is supportive. Removing $\mathcal{L}_{\text{aux}}$ from Equation (7) lowers the overall success rate by 8.6 points, showing that prefix supervision helps stabilize training. In contrast, removing $\mathcal{L}_{\text{GRPO}}$ reduces the method to Sampled-Prefix SFT and drops performance to 26.6%, a 68.7-point decrease. This indicates that imitating sampled prefixes alone is insufficient; the main gain comes from GRPO learning on sampled post-prefix states, while $\mathcal{L}_{\text{aux}}$ acts as a stabilizer for prefix-token imitation.

5 Related Work

Hint-Guidance Strategies. Hint-based RL injects an expert trajectory prefix into rollouts to address reward sparsity, and methods differ primarily in how they choose the prefix depth d. *Schedule-based* methods set d from the training step or batch success rate and share one value across samples (Xi et al., 2024a; Huang et al., 2026; Guo et al., 2026; Lu et al., 2026b). *Per-sample* methods make d instance-specific by probing each sample with extra rollouts (Zhang et al., 2026a, 2025a; Wang et al., 2026a; Zhang et al., 2026c; Su et al., 2026; Li et al., 2025). In both lines, d is a deterministic scalar that ignores per-task heterogeneity within a training context, and existing methods are evaluated almost exclusively on math reasoning. Agent-G^2 draws a per-task d_i from a cluster-level Gaussian whose mean and variance are estimated from GRPO's rollout statistics, removing both the shared-scalar assumption and the probing overhead, and we evaluate it on long-horizon agentic benchmarks.

Auxiliary Supervision for Agentic RL. In agentic settings, prior work typically converts expert trajectories into supervision for an auxiliary model, rather than using them to seed rollouts directly. These auxiliary signals take five forms: SFT targets for behavior cloning (Zeng et al., 2024; Xi et al., 2024b; Qi et al., 2025; Bai et al., 2024), preference pairs for trajectory-level DPO (Song et al., 2024; Putta et al., 2024; Lai et al., 2024; Yuan et al., 2025; Lian et al., 2026a), value heads or Q-critics distilled from demonstrations (Xiang et al., 2024; Zhou et al., 2024b; Feng et al., 2024; Gu et al., 2025), step-level process reward models (Choudhury, 2025; Wang et al., 2025; Xiong et al., 2024; Lu et al., 2026a), and milestone-shaped rewards extracted from successful traces (Wang et al., 2026c; Zheng et al., 2026). Each route adds an auxiliary network and inherits its annotation, Monte-Carlo labeling, or reward-model fragility cost (Zhang et al., 2025b; Pan et al., 2026). Agent-G^2 uses the same expert trajectories directly as rollout-starting prefixes inside an on-policy GRPO loop, training no auxiliary model and requiring no further offline annotation.

6 Conclusion

We proposed Agent-G^2, a Gaussian guidance framework for hint-based reinforcement learning on long-horizon agentic tasks. Rather than treating guidance depth as a deterministic scalar, Agent-G^2 draws it per task from a Gaussian whose center and spread are estimated online from rollouts collected for policy optimization, without learned

depth predictor or extra probe rollouts. Agent-G² is the strongest on ALFWorld and the strongest non-probing method on WebShop at both 1.5B and 7B scales, with 1.5B Agent-G² already surpassing 7B BEACON, showing that schedule design can substitute for backbone scaling.

Limitations

Agent-G^2 relies on the availability of one expert trajectory per training task. When such trajectories are unavailable or costly to obtain, the framework cannot be applied directly; extending it to weaker forms of supervision (e.g., suboptimal demonstrations or language hints) is an open direction.

The Gaussian parameterization is motivated by the empirical informativeness profile observed in Section 2, which fits well on our two benchmarks. Tasks with multimodal or heavily skewed depth profiles may benefit from richer distributional families, though our uniform-distribution ablation (Section 4.4) suggests that the exact shape matters less than stochastic coverage of the informative band.

The difficulty clusters are defined offline by expert-trajectory length, which serves as a simple proxy for task horizon and difficulty. Although effective in our experiments, this fixed partitioning does not adapt as the policy improves and task-relative difficulty changes. Online difficulty estimation could further improve adaptivity.

Ethics Statement

We adhere to the ACL Code of Ethics and Code of Conduct. Our work uses only publicly available benchmarks and pretrained open-source models. Code and training scripts will be released under the MIT License upon publication. We used Claude for grammatical refinement; all scientific content and conclusions are the authors' own work.

Acknowledgements

This work was supported by New Generation Artificial Intelligence-National Science and Technology Major Project (2025ZD0123100), National Natural Science Foundation of China (No. 62506332) and CCF-Baidu Open Fund.

References

Hao Bai, Yifei Zhou, Mert Cemri, Jiayi Pan, Alane Suhr, Sergey Levine, and Aviral Kumar. 2024. Digirl: Training in-the-wild device-control agents with autonomous reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 37, pages 12461–12495. Curran Associates, Inc.

Daniil A. Boiko, Robert MacKnight, and Gabe Gomes. 2023. Emergent autonomous scientific research capabilities of large language models. *Preprint*, arXiv:2304.05332.

Andres M Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. 2023. Chemcrow: Augmenting large-language models with chemistry tools. *Preprint*, arXiv:2304.05376.

Tongbo Chen, Zhengxi Lu, Zhan Xu, Guocheng Shao, Shaohan Zhao, Fei Tang, Yong Du, Kaitao Song, Yizhou Liu, Yuchen Yan, Wenqi Zhang, Xu Tan, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. 2026. Knowu-bench: Towards interactive, proactive, and personalized mobile agent evaluation. *Preprint*, arXiv:2604.08455.

Sanjiban Choudhury. 2025. Process reward models for llm agents: Practical framework and directions. *Preprint*, arXiv:2502.10325.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 3416 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. 2023. Mind2web: Towards a generalist agent for the web. In *Advances in Neural Information Processing Systems*, volume 36, pages 28091–28114. Curran Associates, Inc.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. 2025. Group-in-group policy optimization for llm agent training. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 46375–46408. Curran Associates, Inc.

Peiyuan Feng, Yichen He, Guanhua Huang, Yuan Lin, Hanchong Zhang, Yuchen Zhang, and Hang Li. 2024. Agile: A novel reinforcement learning framework of llm agents. In *Advances in Neural Information Processing Systems*, volume 37, pages 5244–5284. Curran Associates, Inc.

Carlos Florensa, David Held, Xinyang Geng, and Pieter Abbeel. 2018. Automatic goal generation for reinforcement learning agents. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1515–1528. PMLR.

Yu Gu, Kai Zhang, Yuting Ning, Boyuan Zheng, Boyu Gou, Tianci Xue, Cheng Chang, Sanjari Srivastava,

Yanan Xie, Peng Qi, Huan Sun, and Yu Su. 2025. Is Your LLM Secretly a World Model of the Internet? Model-Based Planning for Web Agents. *Transactions on Machine Learning Research*.

Yongxin Guo, Wenbo Deng, Zhenglin Cheng, and Xiaoying Tang. 2026. G^2RPO-a: Guided group relative policy optimization with adaptive guidance. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4525–4539, San Diego, California, United States. Association for Computational Linguistics.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. 2022. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 9118–9147. PMLR.

Zeyu Huang, Tianhao Cheng, Zihan Qiu, Zili Wang, Xu Yinghui, Edoardo Ponti, and Ivan Titov. 2026. Blending supervised and reinforcement fine-tuning with prefix sampling. In *Forty-third International Conference on Machine Learning*.

brian ichter, Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, Dmitry Kalashnikov, Sergey Levine, Yao Lu, Carolina Parada, Kanishka Rao, Pierre Sermanet, Alexander T Toshev, Vincent Vanhoucke, and 26 others. 2023. Do as i can, not as i say: Grounding language in robotic affordances. In *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pages 287–318. PMLR.

Hanyu Lai, Xiao Liu, Iat Long Iong, Shuntian Yao, Yuxuan Chen, Pengbo Shen, Hao Yu, Hanchen Zhang, Xiaohan Zhang, Yuxiao Dong, and Jie Tang. 2024. Autowebglm: A large language model-based web navigating agent. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, pages 5295–5306. ACM.

Dinging Li, Yingxiu Zhao, Xinrui Cheng, Kangheng Lin, Hongbo Peng, Hongxing Li, Zixuan Wang, Yuhong Dai, Haodong Li, Jia Wang, Yukang Shi, Liang Zhao, Jianjian Sun, Zheng Ge, Xiangyu Zhang, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. 2026. Spatialevo: Self-evolving spatial intelligence via deterministic geometric environments. *Preprint*, arXiv:2604.14144.

Ziheng Li, Zexu Sun, Jinman Zhao, Erxue Min, Yongcheng Zeng, Hui Wu, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Xu Chen, and Zhi-Hong Deng. 2025. Staying in the sweet spot: Responsive reasoning evolution via capability-adaptive hint scaffolding. *Preprint*, arXiv:2509.06923.

Niu Lian, Tongbo Chen, Zhehao Yu, Chengzhen Duan, Fazhan Liu, Hui Liu, Pei Fu, Jian Luan, Heng Qu, Shu-Tao Xia, and Jinpeng Wang. 2026a. Ui-mopd: Multi-platform on-policy distillation for unified gui agents. *Preprint*, arXiv:2607.04425.

Niu Lian, Yuting Wang, Hanshu Yao, Jinpeng Wang, Bin Chen, Yaowei Wang, Min Zhang, and Shu-Tao Xia. 2026b. From verbatim to gist: Distilling pyramidal multimodal memory via semantic information bottleneck for long-horizon video agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11601–11617.

Zhengxi Lu, Zhiyuan Yao, Zhuowen Han, Zi-Han Wang, Jinyang Wu, Qi Gu, Xunliang Cai, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. 2026a. Self-distilled agentic reinforcement learning. *Preprint*, arXiv:2605.15155.

Zhengxi Lu, Zhiyuan Yao, Jinyang Wu, Chengcheng Han, Qi Gu, Xunliang Cai, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. 2026b. Skill0: In-context agentic reinforcement learning for skill internalization. *Preprint*, arXiv:2604.02268.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Teng Pan, Yuchen Yan, Zixuan Wang, Ruiqing Zhang, Guiyang Hou, Wenqi Zhang, Weiming Lu, Jun Xiao, and Yongliang Shen. 2026. CoVerRL: Breaking the consensus trap in label-free reasoning via generator-verifier co-evolution. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 29833–29853, San Diego, California, United States. Association for Computational Linguistics.

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

Zehan Qi, Xiao Liu, Iat Long Iong, Hanyu Lai, Xueqiao Sun, Jiadai Sun, Xinyue Yang, Yu Yang, Shuntian Yao, Wei Xu, Jie Tang, and Yuxiao Dong. 2025. Webrl: Training llm web agents via self-evolving online curriculum reinforcement learning. In *International Conference on Learning Representations*, volume 2025, pages 79791–79821.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

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

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. In *Advances in Neural Information Processing Systems*, volume 36, pages 38154–38180. Curran Associates, Inc.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, EuroSys '25, page 1279–1297, New York, NY, USA. Association for Computing Machinery.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pages 8634–8652. Curran Associates, Inc.

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

Yifan Song, Da Yin, Xiang Yue, Jie Huang, Sujian Li, and Bill Yuchen Lin. 2024. Trial and error: Exploration-based trajectory optimization for LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*.

Mingyu Su, Jian Guan, Yuxian Gu, Minlie Huang, and Hongning Wang. 2026. Trust-region adaptive policy optimization. In *International Conference on Learning Representations*, volume 2026, pages 51806–51828.

Geng Tu, Dingming Li, Jun Huang, and Ruifeng Xu. 2026. Consensus-driven multi-agent cognitive reasoning for enhancing the emotional intelligence of large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(21):17751–17759.

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

Xinyi Wang, Jinyi Han, Zishang Jiang, Tingyun li, Jiaqing Liang, Sihang Jiang, Zhaoqian Dai, Ma Shuguang, Fei Yu, and Yanghua Xiao. 2026a. Don't tell the answer, truly guide the reasoning during RL rollouts. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 3437–3455, San Diego, California, United States. Association for Computational Linguistics.

Zixuan Wang, Dingming Li, Hongxing Li, Yanrui Miao, Shuo Chen, Yuchen Yan, Wenqi Zhang, Yongliang Shen, Weiming Lu, Jun Xiao, and Yueting Zhuang. 2026b. Groundact: Can llm agents ground actions in environmental states? *Preprint*, arXiv:2508.05614.

Zixuan Wang, Yuchen Yan, Hongxing Li, Teng Pan, Dingming Li, Ruiqing Zhang, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. 2026c. Milestone-guided policy learning for long-horizon language agents. In *Forty-third International Conference on Machine Learning*.

Zhiheng Xi, Wenxiang Chen, Boyang Hong, Senjie Jin, Rui Zheng, Wei He, Yiwen Ding, Shichun Liu, Xin Guo, Junzhe Wang, Honglin Guo, Wei Shen, Xiaoran Fan, Yuhao Zhou, Shihan Dou, Xiao Wang, Xinbo Zhang, Peng Sun, Tao Gui, and 2 others. 2024a. Training large language models for reasoning through reverse curriculum reinforcement learning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 54030–54048. PMLR.

Zhiheng Xi, Yiwen Ding, Wenxiang Chen, Boyang Hong, Honglin Guo, Junzhe Wang, Dingwen Yang, Chenyang Liao, Xin Guo, Wei He, Songyang Gao, Lu Chen, Rui Zheng, Yicheng Zou, Tao Gui, Qi Zhang, Xipeng Qiu, Xuanjing Huang, Zuxuan Wu, and Yu-Gang Jiang. 2024b. Agentgym: Evolving large language model-based agents across diverse environments. *Preprint*, arXiv:2406.04151.

Yufei Xiang, Yiqun Shen, Yeqin Zhang, and Cam-Tu Nguyen. 2024. Retrospex: Language agent meets offline reinforcement learning critic. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Weimin Xiong, Yifan Song, Xiutian Zhao, Wenhao Wu, Xun Wang, Ke Wang, Cheng Li, Wei Peng, and Sujian Li. 2024. Watch every step! LLM agent learning via iterative step-level process refinement. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1556–1572, Miami, Florida, USA. Association for Computational Linguistics.

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

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

Siyu Yuan, Zehui Chen, Zhiheng Xi, Junjie Ye, Zhengyin Du, and Jiecao Chen. 2025. Agent-r: Training language model agents to reflect via iterative self-training. *Preprint*, arXiv:2501.11425.

Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. 2024. AgentTuning: Enabling generalized agent abilities for LLMs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3053–3077, Bangkok, Thailand. Association for Computational Linguistics.

Kaiyi Zhang, Ang Lv, Jinpeng Li, Yongbo Wang, Feng Wang, Haoyuan hu, and Rui Yan. 2026a. StepHint: Multi-level stepwise hints enhance reinforcement learning to reason. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 37846–37864, San Diego, California, United States. Association for Computational Linguistics.

Ningyu Zhang, Yunzhi Yao, Jiaxin Qin, Haoming Xu, Yuqi Zhu, Zeping Yu, Mengru Wang, Yuqi Tang, Jia-Chen Gu, Shumin Deng, and Huajun Chen. 2026b. Towards principled knowledge editing methods for large language model reasoning. *Nature Machine Intelligence*, 8(8):1189–1200.

Xichen Zhang, Sitong Wu, Yinghao Zhu, Haoru Tan, Shaozuo Yu, Ziyi He, and Jiaya Jia. 2026c. Scaf-grpo: Scaffolded group relative policy optimization for enhancing llm reasoning. In *International Conference on Learning Representations*, volume 2026, pages 131946–131974.

Xuechen Zhang, Zijian Huang, Yingcong Li, Chenshun Ni, Jiasi Chen, and Samet Oymak. 2025a. Bread: Branched rollouts from expert anchors bridge sft & rl for reasoning. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 96726–96752. Curran Associates, Inc.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025b. The lessons of developing process reward models in mathematical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 10495–10516, Vienna, Austria. Association for Computational Linguistics.

Zijing Zhang, Ziyang Chen, Mingxiao Li, Zhaopeng Tu, and Xiaolong Li. 2026d. Rlvmr: Reinforcement learning with verifiable meta-reasoning rewards for robust long-horizon agents. In *International Conference on Learning Representations*, volume 2026, pages 114676–114700.

Congmin Zheng, Xiaoyun Mo, Xinbei Ma, Qiqiang Lin, Yin Zhao, Jiachen Zhu, Xingyu Lou, Jun Wang, Zhaoxiang Wang, Weiwen Liu, Zhuosheng Zhang, Yong Yu, and Weinan Zhang. 2026. Adaptive milestone reward for gui agents. *Preprint*, arXiv:2602.11524.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. 2024a. Webarena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, volume 2024, pages 15585–15606.

Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. 2024b. ArCHer: Training language model agents via hierarchical multi-turn RL. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 62178–62209. PMLR.

A Diagnostic Protocol Details

In-range set and mismatch ratio. Let $\mathcal{D}$ denote the guidance-depth grid and $\mathcal{T}$ the diagnostic task set. For each $(i, d) \in \mathcal{T} \times \mathcal{D}$, we estimate the per-depth success rate by enumeration with $M = 32$ independent rollouts:

$$\hat{p}_i(d) = \frac{1}{M} \sum_{m=1}^{M} \mathbf{1}[\text{rollout } m \text{ at depth } d \text{ succeeds}] . \tag{8}$$

Following Florensa et al. (2018), the *in-range set* of task i is the subset of depths whose estimated success rate lies in the just-learnable band $[0.4, 0.6]$:

$$\mathcal{R}_i = \{ d \in \mathcal{D} : \hat{p}_i(d) \in [0.4, 0.6] \} . \tag{9}$$

Depths with $\hat{p}_i(d) < 0.4$ are *under-guided* and those with $\hat{p}_i(d) > 0.6$ are *over-guided*. Given a scheduler that assigns depth d_i to each task i, the *mismatch ratio* is the fraction of tasks placed outside their in-range set:

$$\rho = \frac{1}{|\mathcal{T}|} \sum_{i \in \mathcal{T}} \mathbf{1}[d_i \notin \mathcal{R}_i] . \tag{10}$$

$\rho = 0$ corresponds to every task receiving an in-range depth, and $\rho = 1$ corresponds to every assignment being under- or over-guided.

Shared-depth schedulers. A shared scheduler outputs one depth d_t per training step that is applied to every task in the batch, so the per-task assignment is $d_i = d_t$ for all i. The stacked bars in Figure 2(a) report the empirical fraction of assignments in each of the three categories defined above: under-guided ($\hat{p}_i(d_t) < 0.4$), in-range ($d_t \in \mathcal{R}_i$), and over-guided ($\hat{p}_i(d_t) > 0.6$).

Mismatch across training checkpoints. The diagnostic in Figure 2(a) is computed at step 50. To verify that shared-depth mismatch is not specific to this checkpoint, we repeat the protocol at five additional Qwen2.5-1.5B / ALFWorld checkpoints (Table 4). The Fix-step baseline, which holds d_t constant within 20-step windows, exceeds 48% mismatch at every checkpoint, and the two step-decay schedules (Linear, Step-dec) stay near 50%. Target-acc, the only shared scheduler with batch-level feedback, achieves the lowest mismatch (39.8% on average) but still misses the in-range band on roughly two in five tasks. The structural mismatch of shared-depth scheduling thus persists across the training trajectory and is not a step-50 artifact.

Table 4: **Mismatch ratio ρ of shared-depth schedulers across five training checkpoints** on Qwen2.5-1.5B / ALFWorld. n is the number of diagnostic tasks at that step.

Step	n	Fix-step	Linear	Step-dec	Tar-acc
10	28	67.9	42.9	42.9	32.1
20	28	64.3	39.3	39.3	35.7
40	28	75.0	57.1	57.1	50.0
80	58	48.3	55.2	55.2	41.4
160	45	71.1	55.6	55.6	40.0
Avg	—	**65.3**	**50.0**	**50.0**	**39.8**

Probing schedulers. A probing scheduler samples M_{probe} rollouts per visited depth on a candidate subset $\mathcal{D}' \subseteq \mathcal{D}$, forms the noisy estimate $\tilde{p}_i^{(M_{\text{probe}})}(d)$, and outputs

$$d_i = \arg\min_{d \in \mathcal{D}'} \left| \tilde{p}_i^{(M_{\text{probe}})}(d) - 0.5 \right| . \tag{11}$$

$\mathcal{D}'$ is the full grid for Enumeration and the subset visited during the search for Binary Search. The output d_i is judged against the reference in-range set $\mathcal{R}_i$ via the mismatch ratio defined above. Figure 2(b) varies $M_{\text{probe}} \in \{2, 4, 8, 16, 32\}$. Scheduler implementations follow the hint-based RL baselines in Table 7.

Agent-G^2. For Agent-G^2 in Figure 2(b), we draw $d_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$ using the schedule parameters reached at step 50 (5 independent draws per task) and report the mismatch ratio averaged over draws and over tasks. No probing rollouts are performed: μ_i and σ_i are read from the rollouts already collected for the GRPO update at that step.

Rollout-cost axis. The horizontal axis in Figure 2(b) reports $|\mathcal{D}'| \cdot M_{\text{probe}}/R$, where $R = 8$ is the GRPO rollout budget per task. Agent-G^2 is plotted at $1\times$ because its schedule is refreshed from these R rollouts without any additional probing.

Reference depth $d_i^\star$. The per-task reference depth

$$d_i^\star = \arg\min_{d \in \mathcal{D}} |\hat{p}_i(d) - 0.5|, \tag{12}$$

with ties broken in favor of the smaller depth, is used only in Figure 3(b) to align profiles via $\Delta d = d - d_i^\star$. It plays no role in the mismatch computation of Section 2.

Gaussian fit. For Figure 3(b), we bin pairs by relative depth Δd with width 0.1 and fit $a \exp(-\Delta d^2 / 2\sigma^2)$ to the binned mean Bernoulli variance by least squares.

Table 5: **Agent-G^2 hyperparameters.** Shared across ALFWorld and WebShop unless noted. Hint-based RL baselines reuse the *Policy optimization*, *Rollout*, and *Training schedule* blocks.

Hyperparameter	Symbol	Value
Agent-G^2-specific		
Global baseline step	Δ	0.1
Cluster EMA rate	α	0.2
Center scale	λ	1.0
Variance scale	γ	1.0
Variance floor	$\sigma_{\min}$	0.1
Aux. loss weight	η	0.5
Target success rate	p_{target}	0.5
Initial baseline	$\mu^{(0)}_{\text{global}}$	0.8
Difficulty clusters	K	3
Policy optimization		
Optimizer	–	AdamW
Peak learning rate	–	1×10^{-5}
LR schedule	–	cosine, 10% warmup
AdamW (β_1, β_2)	–	$(0.9, 0.999)$
Weight decay	–	0.01
GRPO clip ratio	ϵ	0.2
KL penalty	β_{KL}	0.01
Precision	–	bfloat16
Rollout		
Tasks per step	$\lvert\mathcal{B}\rvert$	16
Rollouts per task	R	8
Max prompt length	–	7000
Max response length	–	512
Temperature (train/eval)	–	1.0/0.4
Training schedule		
Steps (ALFWorld)	–	200
Steps (WebShop)	–	150

Gaussian fit across training. Figure 3(b) reports the fit at step 50. To check that the Gaussian shape is not specific to this checkpoint, we partition the training trajectory into three non-overlapping windows, pool aligned pairs $(\Delta d,\, p_i(d)(1-p_i(d)))$ within each window to enlarge the per-window sample size and reduce statistical noise in the fit, and refit the same form (Table 6). The fit quality stays high across all three windows ($R^2 \in [0.89, 0.94]$), and the fitted width σ drifts only mildly from 0.198 early to 0.231 late, consistent with a gradually widening informative band as the policy improves and with the upward trend of the per-cluster σ_k in Figure 6(b). The two-parameter Gaussian thus captures the informativeness profile throughout training, not only at the mid-training checkpoint used in the figure.

Choice of Gaussian for the sampler. Our choice of Gaussian for the sampler in Section 3 rests on two design considerations rather than fit quality alone. First, the Gaussian provides a simple two-moment parameterization whose mean and variance align directly with the two statistics that batch rollouts naturally produce: μ_i is driven by the per-cluster empirical success rate, and σ_i by the per-cluster success-rate variance. Other symmetric families require additional shape parameters (for instance, the degrees of freedom of Student-t or the shape exponent of generalized normal) that cannot be read directly from rollout aggregates. Second, Gaussian sampling on the bounded ratio $r_i \in [0, 1]$ admits a closed-form truncated form and is numerically stable; heavy-tailed alternatives such as Cauchy produce unbounded magnitudes that would require ad-hoc clipping. The ablation in Table 3 confirms that on top of these design considerations, replacing the Gaussian with a variance-matched uniform distribution over the same band drops overall success by 7.0 points and Long-horizon success by 13.9 points.

Table 6: **Gaussian fit of the aligned informativeness profile across training windows** on Qwen2.5-1.5B / ALFWorld. Each window pools $(\Delta d,\, p_i(d)(1-p_i(d)))$ pairs collected within the listed checkpoint range; n counts pairs entering the least-squares fit.

Training Stage	σ	R^2	n
Early (ckpts 0–50)	0.198	0.906	116
Mid (ckpts 50–100)	0.206	0.888	146
Late (ckpts 100–150)	0.231	0.936	109

B Implementation Details

Hyperparameters. Table 5 lists the full Agent-G^2 configuration. On both benchmarks we build on the GiGPO codebase (Feng et al., 2025), adopting its training recipe for shared hyperparameters and inheriting its train/validation/test splits.

Baseline configurations. Table 7 summarizes the trained baselines; prompting baselines run inference-time without further specification. We use the authors' public implementations with default hyperparameters for GiGPO, BEACON, StepHint, and TraPO.

Expert trajectory source. The expert trajectories used by $\mathcal{L}_{\text{aux}}$ are drawn from the same ALFWorld and WebShop expert pool as RLVMR (Zhang et al., 2026d) and ETO (Song et al., 2024), so the underlying expert data is identical across the three methods. The difference lies in usage: Agent-G^2 treats the prefix tokens only as a token-level imitation target inside the joint ob-

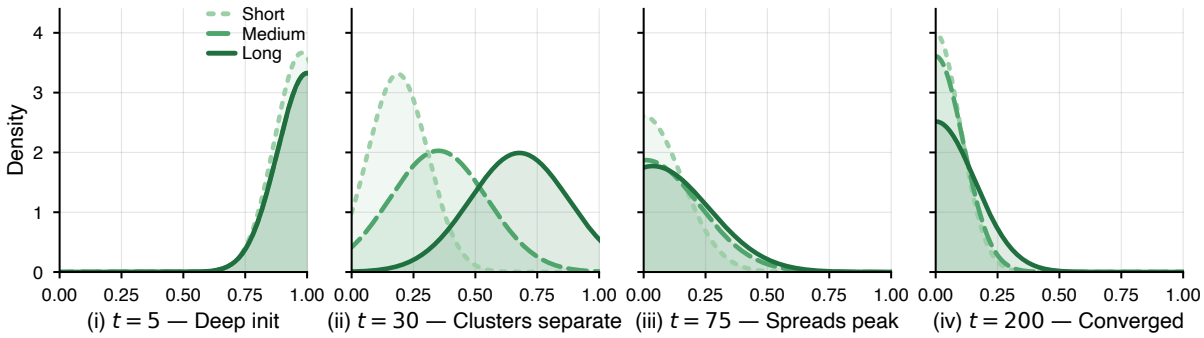

Figure 7: **Four-phase snapshot of the per-cluster Gaussian schedule** on Qwen2.5-1.5B / ALFWorld. Each panel shows the density $\mathcal{N}(\mu_k, \sigma_k^2)$ for the K=3 horizon clusters (Short, Medium, Long) at one training step. Their batch-weighted mixture yields the aggregate in Figure 6(b).

Table 7: **Baseline configurations.** Hint-based RL baselines reuse the training blocks of Table 5; only the depth-assignment rule differs.

Method	Configuration
Imitation	
Full SFT	LR 1×10^{-5}, $\lvert\mathcal{B}\rvert$=32, 2 epochs
RL without hints	
GRPO	Table 5 with d_i=0
GiGPO	authors' default
BEACON	authors' default
Hint-based RL: schedule	
Linear	linear, $1.0 \rightarrow 0.0$
Cosine	cosine, $1.0 \rightarrow 0.0$
Step	step decay 0.1 / 20 steps
Target-acc	online to batch-acc 0.8
Hint-based RL: search	
Binary Search	$O(\log \lvert\mathcal{D}\rvert)$ probes per task
Enumeration	exhaustive over $\mathcal{D}$
Hint-based RL: end-to-end	
StepHint	authors' default
TraPO	authors' default

jective, without introducing any auxiliary network or additional supervision signal. The gains over RLVMR and ETO in Table 1 therefore reflect algorithmic differences rather than any advantage in expert-data access.

Hardware and framework. We implement training in the verl framework (Sheng et al., 2025) and run all experiments on a single node with 8 NVIDIA H800 GPUs; a Qwen2.5-1.5B-Instruct training run on ALFWorld (200 gradient steps) takes approximately 10 hours.

Per-cluster training dynamics. Figure 7 samples the per-cluster Gaussian schedule $\mathcal{N}(\mu_k, \sigma_k^2)$ at four training steps that mark distinct phases of training. At t=5 (*saturation*), μ_k stays near 1 on every cluster (0.98, 1.00, 1.00 for Short, Medium, Long) with $\sigma_k \approx 0.11$, because the policy cannot yet reach learnable states unaided. At t=30 (*separation*), the means split by horizon (0.19, 0.35, 0.68) and σ_k widens with the cluster variance V_k. At t=75 (*peak spread*), the means contract toward 0 on all clusters while σ_k reaches its peak (0.15, 0.21, 0.23); the sampler draws a wide neighborhood around the shallow center to keep harder tasks in the just-learnable band. By t=200 (*collapse*), $\mu_k = 0$ on every cluster and σ_k recedes toward $\sigma_{\min}$=0.1, with only Long retaining a wider spread (0.16), consistent with the slower convergence of long-horizon tasks. Figure 8 plots the full (μ_k, σ_k) trajectories across all 200 training steps for the same three clusters.

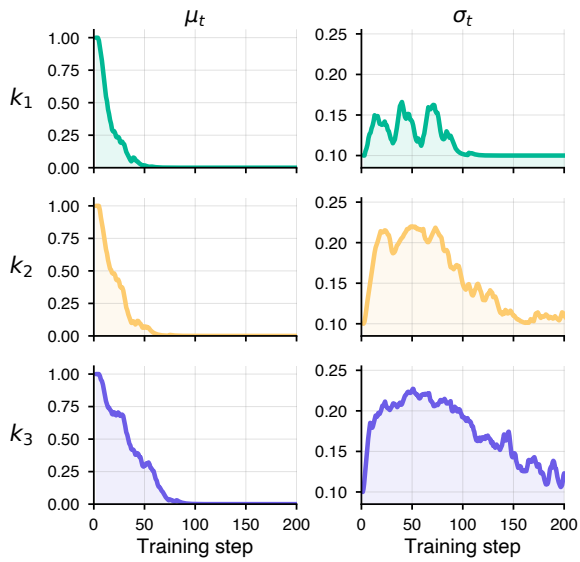

Figure 8: **Per-cluster schedule trajectories** on Qwen2.5-1.5B / ALFWorld. Each row shows one horizon cluster (k_1 Short, k_2 Medium, k_3 Long); columns plot μ_t and σ_t over training steps.

Table 8: **Clustering design ablation** on Qwen2.5-1.5B / ALFWorld (success rate %). Length-quantile bins tasks by horizon; random samples a partition uniformly. $\pm$ values are 5-seed standard deviations. K=1 and length-quantile K=3 are reproduced from Table 3.

Clustering signal	K	**All**
None	1	89.1
Length quantile	3	$95.3_{\pm 2.3}$
Length quantile	5	$96.5_{\pm 1.9}$
Length quantile	10	86.2
Random	3	89.4

Clustering design. Table 8 sweeps the cluster count for length-quantile clustering and adds a random-assignment control on Qwen2.5-1.5B / ALFWorld.

Length-quantile clustering is stable in the K=3–5 range: K=3 ($95.3_{\pm 2.3}$) and K=5 ($96.5_{\pm 1.9}$) overlap within run-to-run noise, so the difference between them is not significant under our seed budget. Pushing to K=10 thins each cluster to under two tasks per batch under $|\mathcal{B}|$=16 and drops accuracy by 9.1 points relative to K=3, as per-cluster sample size becomes inadequate for stable EMA estimation. Random assignment at K=3 comes within 0.5 points of K=1, so the gain from clustering relies on alignment with difficulty, not on partitioning itself.

We pick length as the alignment signal for two reasons. (i) Length correlates with the difficulty of reaching the terminal reward without guidance, so it accelerates the convergence of (A_k, V_k) relative to a difficulty-orthogonal signal. (ii) Length needs no task-type annotation or semantic embedding, so the same rule applies to both benchmarks.

We fix K=3 across both ALFWorld and WebShop without per-benchmark tuning, because it matches the Short / Medium / Long taxonomy of Section 4.1 and keeps per-cluster sample size sufficient for stable EMA under $|\mathcal{B}|$=16. The 6.2-point K=1 vs. K=3 gap in Table 3 bounds what a richer signal could recover; in benchmarks where horizon and difficulty decouple, additional signals beyond length may help.