

On-Policy Distillation with Curriculum Turn-level Guidance for Multi-turn Agents

Gengsheng Li^{1,2,3*}, Mao Zheng^{3*}, Mingyang Song^{3*}, Ruiqi Liu^{2*},
Tianyu Yang^{1,2}, Jie Sun^{3,4}, Qiyong Zhong^{3,5}, Dan Zhang⁶,
Haiyun Guo^{1,2†}, Junfeng Fang^{6†}, Jinqiao Wang^{1,2,7}

¹Foundation Model Research Center, Institute of Automation, Chinese Academy of Sciences

²School of Artificial Intelligence, University of Chinese Academy of Sciences

³Foundation Model Department, Tencent ⁴University of Science and Technology of China

⁵Zhejiang University ⁶National University of Singapore ⁷Wuhan AI Research

ligengsheng2024@ia.ac.cn

Abstract

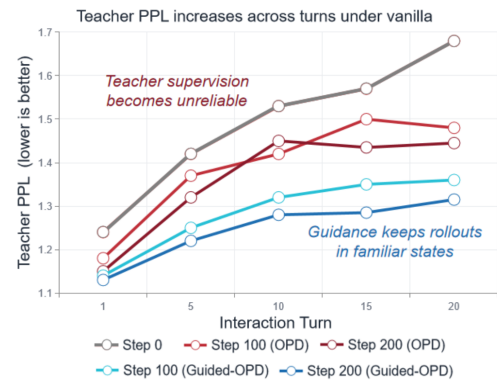
Multi-turn agents that plan, invoke tools, and interact with environments offer a promising paradigm for solving complex tasks, yet their capabilities typically rely on very large models whose inference cost is prohibitive in practice. On-Policy Distillation (OPD) is a natural recipe for transferring such capabilities to smaller students, but we find that it suffers a characteristic failure mode in this setting: small student errors compound across turns and push the trajectory out of the teacher’s familiar state distribution, so the teacher’s supervision becomes least reliable precisely where the student needs it most. We propose Guided On-Policy Distillation (Guided-OPD), a simple yet effective algorithm that mixes teacher- and student-generated turns within each rollout and schedules the teacher’s intervention probability along a curriculum that decays to zero. Strong guidance keeps early trajectories close to the teacher distribution and is then gradually withdrawn to recover the purely on-policy regime used at inference. On ALFWorld, ScienceWorld, and WebShop, distilling Qwen3 students from a Qwen3-30B-A3B teacher, Guided-OPD yields average relative gains of 21.1% in Score and 25.5% in Success Rate over vanilla OPD, with larger gains on smaller students.

1 Introduction

Multi-turn agents that autonomously plan (Wang et al., 2023), use tools (Schick et al., 2023; Singh et al., 2025), and interact with their environment (Yao et al., 2023; Chen et al., 2025) are becoming an essential paradigm for solving complex

*Equal contribution. †Corresponding authors. Code: <https://github.com/Zzzz-166/Guided-OPD>

(a) Teacher PPL increases across agent turns



(b) Guided-OPD enables exploration in familiar distribution

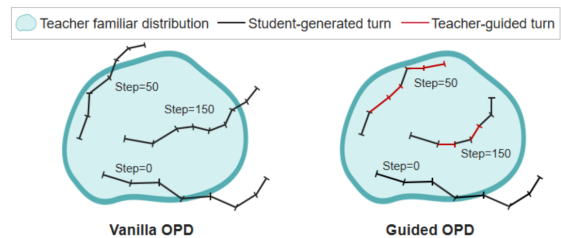


Figure 1: Failure mode of vanilla OPD on multi-turn agents and how Guided-OPD addresses it. (a) Teacher PPL on student-generated tokens versus interaction turn at different training steps: under vanilla OPD, PPL grows with the turn index and remains high even after 200 steps, while Guided-OPD lowers PPL across all turns, with the gap widening at later turns. (b) Schematic of the rollout dynamics. The teal region denotes the teacher’s familiar distribution, where black and red segments denote student- and teacher-generated turns. Vanilla OPD samples every turn from the student and quickly leaves this region, while Guided-OPD intersperses teacher turns to anchor the rollout, with the curriculum schedule injecting more teacher turns early and fewer as training proceeds.

tasks (Xi et al., 2025; Yu et al., 2025b). Such capabilities, however, are typically realized by mod-

els with very large parameter counts (Yang et al., 2025), which incur substantial inference cost and deployment overhead (Xu et al., 2024a). A central question for building practical agents is therefore how to stably transfer the strong agentic capabilities of a large teacher model to a much smaller student (Xu et al., 2024b; Kang et al., 2025; Qiu et al., 2025; Liu et al., 2025; Li et al., 2025). On-Policy Distillation (OPD) has recently emerged as an effective paradigm for transferring complex reasoning abilities (Lu and Thinking Machines Lab, 2025; Xiao et al., 2026; Sun et al., 2026; Zhong et al., 2026). By aligning the teacher distribution directly on trajectories sampled from the student itself (Agarwal et al., 2024; Gu et al., 2024), OPD combines the training-inference consistency of on-policy learning with the dense supervision of token-level distillation, and yields strong results on reasoning tasks such as mathematics (Jin et al., 2026; Xu et al., 2026) and code generation (Hou et al., 2026; Yang et al., 2026b). Nevertheless, previous work points out that OPD often suffers from noisy supervision signals (Fu et al., 2026; Li et al., 2026b) and unstable optimization (Zhu et al., 2026; Luo et al., 2026; Song and Zheng, 2026; Li et al., 2026a; Yang et al., 2026a). Since multi-turn agents constitute a dynamic interactive setting that differs sharply from static reasoning (Yu et al., 2025b; Zhou et al., 2024) and can plausibly amplify the issues already reported there (Xue et al., 2026; Wang et al., 2026a), whether OPD remains effective in this regime is an urgent open question.

In this paper, we apply OPD to multi-turn agents and identify a characteristic failure mode. For every agent turn, we compute the cross-entropy of the teacher with respect to the tokens generated by the student, and exponentiate it to obtain the teacher perplexity (PPL). As shown in Figure 1(a), under vanilla OPD the teacher PPL grows with the turn index, indicating that the reliability of the teacher’s supervision degrades as the agent interacts more extensively with the environment. We attribute this behavior to a form of distribution drift that is intrinsic to the multi-turn setting. Within a single rollout, each student action triggers unpredictable environment feedback, and small biases introduced in early turns can propagate and compound across the trajectory, so that within only a handful of turns the student may steer the rollout out of the state distribution familiar to the teacher. On such trajectories the teacher itself becomes uncertain, and its output distribution no longer serves as a reliable

distillation target. More critically, this effect is not naturally alleviated by longer training: as shown in Figure 1(a), the overall PPL drops only marginally from step 100 to step 200, suggesting that the elevated PPL early in training already constitutes a bottleneck that obstructs efficient knowledge transfer from the teacher to the student.

To address this issue, we propose Guided On-Policy Distillation (Guided-OPD), a simple yet effective training paradigm. Instead of letting the student complete an entire trajectory on its own, Guided-OPD samples each turn from the teacher with probability β and from the student with probability $1 - \beta$, with both models sharing the same context. Building on this design, we further introduce a curriculum schedule that decays β as training proceeds. Early in training, when the student is weak and prone to drift, a relatively large β injects more teacher guidance and keeps the trajectory within the teacher’s familiar distribution as much as possible, thereby securing reliable supervision. As the student improves and becomes less likely to push the rollout out of distribution, the decay of β encourages broader self-exploration and gradually returns training to a purely on-policy regime, which avoids the training-inference mismatch. As shown in Figure 1(a), Guided-OPD substantially lowers the teacher PPL and provides the student with more reliable supervision, while Figure 1(b) presents an intuitive schematic of the method.

We systematically validate Guided-OPD on three representative multi-turn agentic benchmarks: ALFWorld (Shridhar et al., 2021), ScienceWorld (Wang et al., 2022), and WebShop (Yao et al., 2022), under three distillation settings that transfer a Qwen3-30B-A3B teacher to 0.6B, 1.7B, and 4B students. Across the three configurations Guided-OPD delivers average relative gains of +21.1% in Score and +25.5% in Success Rate over vanilla OPD, while reducing the average number of interaction turns by 6.0%. The improvement grows as the student becomes smaller: on the 0.6B student, the relative gains reach +35.3% in Score and +32.8% in Success Rate, with a 7.9% reduction in interaction turns. Our main contributions are summarized as follows:

- We identify a characteristic failure mode of OPD on multi-turn agents, in which the teacher PPL over student-generated tokens grows steadily along the trajectory and is not relieved by longer training, exposing rollout-

level distribution drift as a fundamental obstacle to reliable distillation.

- We propose Guided-OPD, which mixes teacher- and student-sampled turns within each rollout and equips this mixture with a curriculum schedule, keeping early trajectories close to the teacher distribution while smoothly converging to a purely on-policy regime in later training stage.
- We validate Guided-OPD on ALFWorld, ScienceWorld, and WebShop across 0.6B, 1.7B, and 4B students distilled from Qwen3-30B-A3B, observing consistent gains in Score and Success Rate, fewer interaction turns, and larger benefits on smaller students.

2 Preliminary

Multi-turn Agent. We consider an agent interacting with an environment over a finite horizon T , indexed by $t \in \{0, \dots, T-1\}$. At each turn t , the agent receives an observation o_t and produces a response a_t that consists of a reasoning trace followed by an executable action (Yao et al., 2023), after which the environment returns the next observation o_{t+1} . We use *turn t* to refer to one such interaction round (i.e., the cycle $o_t \rightarrow a_t \rightarrow o_{t+1}$), and reserve a_t for the agent-produced segment within it. Since the environment is partially observable, we define the agent state as the full interaction history up to the current observation,

$$h_t = (o_0, a_0, \dots, o_{t-1}, a_{t-1}, o_t), \quad (1)$$

and a complete trajectory as $\tau = (h_0, a_0, \dots, h_{T-1}, a_{T-1})$, which terminates when a termination action is executed or when T is reached. The agent is parameterized as an autoregressive language model $\pi_\theta(a_t | h_t)$.

On-Policy Distillation. Given a frozen teacher policy π_ϕ and a student policy π_θ , OPD aligns the student with the teacher *under the student’s own state distribution*: the student rolls out a trajectory $\tau \sim \pi_\theta$, and the teacher then scores τ via a token-level divergence at each visited state. The training objective is

$$\mathcal{L}_{\text{OPD}}(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{T-1} \mathcal{D} \left(\pi_\phi(a_t | h_t), \pi_\theta(a_t | h_t) \right) \right]. \quad (2)$$

where $\mathcal{D}(\cdot, \cdot)$ is a token-level divergence between the teacher and student conditional distributions, generally including three standard instantiations: (i) *forward KL* $\mathcal{D}_{\text{KL}}(\pi_\phi \| \pi_\theta)$, (ii) *reverse KL* $\mathcal{D}_{\text{KL}}(\pi_\theta \| \pi_\phi)$, and (iii) *Jensen–Shannon divergence*, the symmetric interpolation $\frac{1}{2} \mathcal{D}_{\text{KL}}(\pi_\phi \| M) + \frac{1}{2} \mathcal{D}_{\text{KL}}(\pi_\theta \| M)$ with $M = \frac{1}{2}(\pi_\phi + \pi_\theta)$.

3 Method

In this section, we propose Guided-OPD, an on-policy distillation algorithm designed for multi-turn agents and illustrated in Figure 2. Its central idea is to inject teacher guidance during the student rollout at the granularity of a single turn and to schedule this guidance in a curriculum fashion. Early in training, the teacher takes over a complete turn with relatively high probability, pulling the trajectory back toward the state distribution familiar to the teacher and alleviating the supervision degradation that vanilla OPD suffers from when errors accumulate over the long-horizon interaction of a multi-turn agent. As training progresses, the probability of teacher intervention decays along a predefined schedule and eventually reaches zero, so the training process converges smoothly to a pure OPD regime that is fully consistent with inference. Concretely, Guided-OPD consists of two complementary components. Turn-level Guidance (§3.1) specifies the granularity of teacher intervention and the distillation loss assigned to turns generated by each role, while Curriculum Guidance (§3.2) specifies how the teacher intervention probability β decays with the training step. The unified optimization objective is then presented in §3.3.

3.1 Turn-level Guidance

The turn as the unit of guidance. We refer to the complete ReAct (Yao et al., 2023) style thought-plus-action response a_t that the agent produces conditioned on history h_t , excluding the subsequent environment observation o_{t+1} , as a turn. We choose the turn rather than the token as the unit of guidance because, in a multi-turn agent, the quantity that actually shifts the subsequent state distribution is not an individual token but a complete action together with the observation it triggers (Zhong et al., 2026). Switching between teacher and student at turn boundaries therefore aligns the role transitions with the underlying state transitions of the environment.

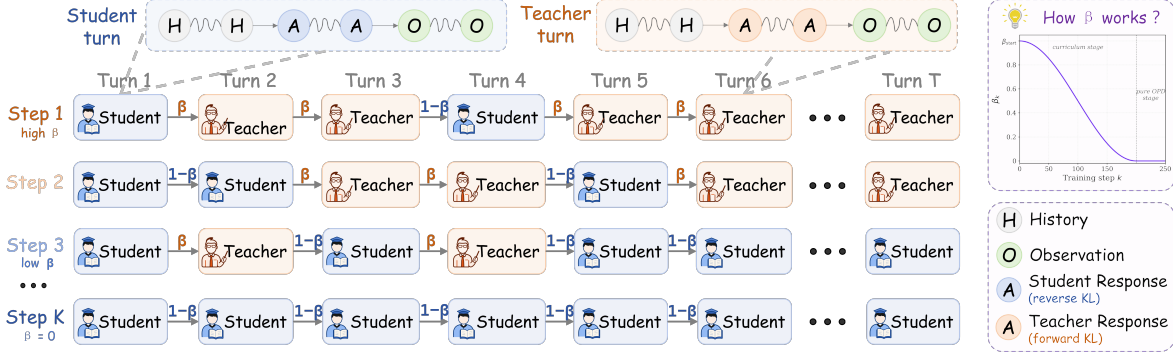


Figure 2: Overview of Guided-OPD. The central grid traces a rollout across training as a sequence of turns. Within each turn, a response A is generated token by token from the history H and triggers a new observation O , sampled from either the student (reverse-KL target, blue) or the teacher (forward-KL target, orange). The role of every turn is drawn independently at probability β for the teacher and $1 - \beta$ for the student, where β is set by the curriculum schedule shown in the top-right inset and decays smoothly from β_{start} to zero (cosine by default). Early in training (Step 1, high β), most turns are produced by the teacher and the rollout is anchored within the teacher’s familiar distribution. As training proceeds (Step 2 and Step 3), β decreases and the share of student turns grows. Once β reaches zero in the pure-OPD stage (Step K), every turn is generated by the student and the procedure coincides with standard OPD.

Role assignment within a turn. At this granularity, we draw on the stochastic mixing idea from imitation learning (Ross et al., 2011) and introduce a mixing parameter $\beta \in [0, 1]$ that controls the strength of teacher intervention during rollout. Independently at each turn of every trajectory, a_t is generated by the teacher with probability β and by the student with probability $1 - \beta$,

$$a_t \sim \begin{cases} \pi_\theta(\cdot | h_t), & \text{w.p. } 1 - \beta \quad (\text{student turn}), \\ \pi_\phi(\cdot | h_t), & \text{w.p. } \beta \quad (\text{teacher turn}). \end{cases} \quad (3)$$

The value of β is supplied by the curriculum schedule of §3.2 according to the current training progress and is held fixed within a single trajectory. Regardless of which role produces a_t , the response is appended to the shared interaction history, so that the next-turn history h_{t+1} retains an identical prompt structure under both roles.

Loss on a student turn. When a turn is produced by the student, we minimize the reverse KL between the student and the teacher on that turn,

$$\mathcal{L}_{\text{stu}}^{(t)}(\theta) = \mathcal{D}_{\text{KL}}(\pi_\theta(\cdot | h_t) \| \pi_\phi(\cdot | h_t)). \quad (4)$$

Loss on a teacher turn. When a turn is produced by the teacher, we minimize the forward KL between the teacher and the student on that turn,

$$\mathcal{L}_{\text{tea}}^{(t)}(\theta) = \mathcal{D}_{\text{KL}}(\pi_\phi(\cdot | h_t) \| \pi_\theta(\cdot | h_t)). \quad (5)$$

Why this asymmetry? The choice of divergence is not arbitrary; it is matched to the role that drives generation on each turn.

On a student turn, the student is the party actually advancing the trajectory, and what we care about is whether every action the student produces is one that the teacher would endorse. The reverse KL is mode-seeking (Agarwal et al., 2024; Song and Zheng, 2026): it requires the actions sampled by the student to lie in high-probability regions of the teacher, while not requiring the student to cover every action the teacher would support. This concentrates the training signal on correcting what the student currently outputs, tightening the student on its own turns toward the modes preferred by the teacher.

On a teacher turn, the teacher is demonstrating an alternative way of behaving in that state, and we want the student to absorb the full demonstration into its own repertoire. The forward KL is mass-covering: it penalizes the student for placing too little probability on the actions that the teacher actually produces, requiring the support of the student to cover all behaviors the teacher chooses in that state. Applying reverse KL on a teacher turn would instead encourage the student to evaluate the demonstration through the lens of its current preferences and to discard those actions that the student has not yet learned to appreciate but that the teacher considers reasonable, which contradicts the very purpose of introducing demonstrations.

In short, the asymmetry aligns the divergence with the role of the turn: a student turn uses reverse KL to tighten the student output onto modes preferred by the teacher, while a teacher turn uses forward KL to make the student cover all reasonable behaviors demonstrated by the teacher.

3.2 Curriculum Guidance

In the previous subsection, β governs the probability that any given turn is produced by the teacher or by the student. We now specify how β varies with training progress, so that the strong teacher guidance applied at the start of training gradually fades and the procedure eventually reverts to pure OPD.

Curriculum stage. Out of the total T_{total} training steps, we reserve the first fraction ρ (the curriculum ratio, with default $\rho = 0.8$) as the curriculum stage, comprising $T_{\text{decay}} := \rho T_{\text{total}}$ steps. During this stage, β decays smoothly from its initial value β_{start} along a predefined curve to zero. Over the remaining steps, β is held at zero, so that the tail of training is strictly equivalent to standard OPD.

Decay schedules. Let the decay progress be $\xi_t = \min(t/T_{\text{decay}}, 1)$. We consider three candidate schedules,

$$\text{Linear: } \beta_t = \beta_{\text{start}} + (\beta_{\text{end}} - \beta_{\text{start}})\xi_t, \quad (6)$$

$$\text{Cosine: } \beta_t = \beta_{\text{start}} + (\beta_{\text{end}} - \beta_{\text{start}}) \cdot \frac{1 - \cos(\pi\xi_t)}{2}, \quad (7)$$

$$\text{Exp: } \beta_t = \beta_{\text{start}} \left(\frac{\beta_{\text{end}}}{\beta_{\text{start}}} \right)^{\xi_t}, \quad (8)$$

$\beta_{\text{start}}, \beta_{\text{end}} > 0.$

All three schedules satisfy the boundary conditions $\beta_0 = \beta_{\text{start}}$ and $\beta_{T_{\text{decay}}} = \beta_{\text{end}}$, and they differ only in their curvature inside the curriculum. The linear schedule decays at a constant rate, and the exponential schedule decays multiplicatively. We adopt the cosine schedule by default because the cosine schedule changes slowly near both endpoints and faster in between, which matches the intuition of retaining strong guidance early on while exiting gently near the end.

3.3 Final Objective

Combining the per-turn loss of §3.1 with the β schedule of §3.2 gives the full Guided-OPD objective. At training step t , we sample a trajectory τ from the turn-level mixed rollout distribution $\pi_{\theta, \phi}^{\beta_t}$

induced by Eq. 3 and accumulate the corresponding per-turn losses,

$$\mathcal{L}_{\text{Guided-OPD}}(\theta; t) = \mathbb{E}_{\tau \sim \pi_{\theta, \phi}^{\beta_t}} \left[\sum_{u=0}^{T-1} \left((1 - z_u) \mathcal{L}_{\text{stu}}^{(u)}(\theta) + z_u \mathcal{L}_{\text{tea}}^{(u)}(\theta) \right) \right]. \quad (9)$$

where the role indicator $z_u \in \{0, 1\}$ is sampled together with a_u according to Eq. 3, and the two per-turn losses $\mathcal{L}_{\text{stu}}^{(u)}$ and $\mathcal{L}_{\text{tea}}^{(u)}$ are defined in Eq. 4 and Eq. 5.

Taking expectations turn by turn, Eq. 9 admits the equivalent form

$$\mathcal{L}_{\text{Guided-OPD}}(\theta; t) = (1 - \beta_t) \mathcal{J}_{\text{rKL}}^{\pi_{\theta}}(\theta) + \beta_t \mathcal{J}_{\text{fKL}}^{\pi_{\phi}}(\theta). \quad (10)$$

which decomposes Guided-OPD into reverse-KL distillation on the student-induced state distribution and forward-KL distillation on the teacher-induced state distribution, weighted by $1 - \beta_t$ and β_t respectively. As $\beta_t \rightarrow 0$, the objective recovers standard OPD (Eq. 2) and is fully consistent with the student-only rollout used at inference. As $\beta_t \rightarrow 1$, it reduces to forward-KL distillation on teacher trajectories, a safe imitation regime that anchors the student in the teacher’s state distribution. The curriculum schedule starts training from the latter regime and transitions smoothly to the former, jointly addressing the trajectory drift and the training-inference mismatch identified in §1. The complete procedure is summarized in Algorithm 1.

4 Experiments

4.1 Experimental Setup

Implementation. We distill three Qwen3 students of size 0.6B, 1.7B, and 4B from the same teacher Qwen3-30B-A3B (Yang et al., 2025). For Guided-OPD, we use a cosine β schedule (Eq. 7) with $\beta_{\text{start}} = 1$, $\beta_{\text{end}} = 0$, and curriculum ratio $\rho = 0.8$. All methods are implemented on Trinity-RFT (Pan et al., 2025) and run on $8 \times$ NVIDIA H20 GPUs. Full hyperparameters are deferred to Appendix A.4.

Baselines. We compare Guided-OPD against four baselines spanning training-free evaluation, standard OPD, teacher guidance, and curriculum learning: (i) *Zero-shot*, the untrained student; (ii) *Vanilla OPD* (Agarwal et al., 2024; Lu and Thinking Machines Lab, 2025), the standard on-policy distillation of §2; (iii) *AdaSwitch* (Peng et al.,

Algorithm 1: Guided-OPD step

Input: Teacher π_ϕ , student π_θ , environment $\mathcal{E}$, step t .
Hyperparameters: ρ , T_{total} , β_{start} , schedule $\mathcal{S}$.

```
1  $T_{\text{decay}} \leftarrow \rho T_{\text{total}}$ 
2  $\beta_t \leftarrow \mathcal{S}(t, T_{\text{decay}})$  // default cosine
3 Mixed rollout.
4  $h_0 \leftarrow \mathcal{E}.\text{RESET}(); \tau \leftarrow []$ 
5 for  $u = 0, 1, \dots$  until terminal do
6   Sample  $z_u$  with  $\Pr(z_u = \text{tea}) = \beta_t$ 
7   if  $z_u = \text{tea}$  then
8      $a_u \sim \pi_\phi(\cdot | h_u)$ 
9   else
10     $a_u \sim \pi_\theta(\cdot | h_u)$ 
11     $o_{u+1} \leftarrow \mathcal{E}.\text{STEP}(a_u)$ 
12     $h_{u+1} \leftarrow (h_u, a_u, o_{u+1})$ 
13    Append  $(h_u, a_u, z_u)$  to  $\tau$ 
14 Asymmetric distillation.
15  $\mathcal{L}_{\text{stu}} \leftarrow \sum_{u: z_u = \text{stu}} \mathcal{L}_{\text{stu}}^{(u)}(\theta)$ 
16  $\mathcal{L}_{\text{tea}} \leftarrow \sum_{u: z_u = \text{tea}} \mathcal{L}_{\text{tea}}^{(u)}(\theta)$ 
17  $\mathcal{L}(\theta) \leftarrow \mathcal{L}_{\text{stu}} + \mathcal{L}_{\text{tea}}$  // Eqs. (4), (5)
18 Optimizer step.
19  $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}(\theta)$ 
```

2025), which hands the rest of a generation to the teacher the first time the token-level divergence exceeds an adaptive threshold; and (iv) *TCOD* (Wang et al., 2026b), a curriculum method that linearly grows the number of turns k participating in the KL computation. Full implementation details are in Appendix A.3.

Evaluation. We evaluate on three multi-turn agent benchmarks covering embodied navigation, scientific reasoning, and e-commerce: ALFWorld (Shridhar et al., 2021), ScienceWorld (Wang et al., 2022), and WebShop (Yao et al., 2022); for ALFWorld we use its official *seen/unseen* splits as IID and OOD evaluations. We report three metrics: *Score* $\uparrow$ (continuous task score in $[0, 100]$, unavailable on ALFWorld), *Success Rate* (SR $\uparrow$, binary completion rate), and *Round* $\downarrow$ (average interaction steps per task). All methods are evaluated on the full validation set of each benchmark under the same decoding configuration. See Appendix A.1 for benchmark details (including Table 3) and Appendix A.2 for metric definitions.

4.2 Main Results

Table 1 reports the full results on ALFWorld, ScienceWorld, and WebShop. Across the 0.6B, 1.7B, and 4B students, Guided-OPD strictly outperforms all baselines on the three aggregate metrics (average Score, SR, Round) within every student-model group and is best on the vast majority of fine-

grained metrics. Against Vanilla OPD, the most direct counterpart, it improves average Score by +35.3%, +17.6%, and +10.3%, and average SR by +32.8%, +23.7%, and +20.1% on the 0.6B, 1.7B, and 4B students, while reducing average Round by 7.9%, 7.3%, and 2.8%. The simultaneous gain in task quality, success, and interaction efficiency indicates that turn-level guidance combined with a curriculum schedule both corrects long-horizon trajectory drift and yields a more compact policy.

Neither teacher guidance nor curriculum learning alone is sufficient. AdaSwitch provides guidance, but once its switch is triggered it hands the remainder of the generation to the teacher and never returns to the student: the tail of the trajectory is always produced by the teacher during training, yet has to be completed by the student alone at inference. This training-inference mismatch at the trajectory tail leaves AdaSwitch above Vanilla OPD only on 0.6B (26.49 vs. 25.16 in average Score), and below it on both 1.7B (31.91 vs. 32.27) and 4B (37.08 vs. 38.66). TCOD provides a curriculum but still samples from the student’s own rollouts and therefore cannot repair a state distribution that has already drifted; its average Score gain over Vanilla OPD shrinks rapidly as the student grows (+23.4%, +6.2%, and +2.2% on 0.6B, 1.7B, and 4B). Guided-OPD combines the strengths of both and consequently attains a higher average Score and SR than either at every student size.

The relative gain of Guided-OPD grows as the student shrinks, from +10.3% in Score on the 4B student to +35.3% on the 0.6B student. This matches the drift analysis of Section 1: smaller students push trajectories further out of the teacher’s familiar distribution, so turn-level anchoring brings a larger marginal benefit. Notably, the 4B student even surpasses the $7.5\times$ larger 30B-A3B teacher in average Score (42.64 vs. 41.40), and independently on ScienceWorld (29.50 vs. 27.73) and WebShop (55.78 vs. 55.07). On the ALFWorld OOD split, Guided-OPD also attains the best SR at every student size; although its IID–OOD gap is not always the smallest, the consistently stronger OOD SR shows that teacher guidance helps the student inherit the teacher’s decision structure without sacrificing generalization to unseen layouts and object combinations.

Table 1: Main results on three multi-turn agent benchmarks (ALFWORLD, SCIENCEWORLD, WEBSHOP). For ALFWORLD we report Success Rate (SR $\uparrow$) and average interaction Round (Round $\downarrow$) under both IID and OOD evaluation settings. For SCIENCEWORLD and WEBSHOP we additionally report task Score $\uparrow$. The teacher (Qwen3-30B-A3B) is shown for reference. Within each student-model group, the best result is highlighted in **bold**.

Params	Method	ALFWorld				ScienceWorld			WebShop			Avg		
		IID		OOD		Score↑	SR↑	Round↓	Score↑	SR↑	Round↓	Score↑	SR↑	Round↓
		SR↑	Round↓	SR↑	Round↓									
Teacher Models from the Qwen3 Series														
30B-A3B	Zero-Shot	43.57	22.46	40.80	22.78	27.73	7.25	21.62	55.07	19.25	4.96	41.40	27.72	17.96
Student Models from the Qwen3 Series														
0.6B	Zero-shot	0.24	30.00	0.25	29.75	5.19	2.25	20.26	10.37	3.75	14.03	7.78	1.62	23.51
	Vanilla OPD	19.29	26.24	16.42	26.67	10.42	1.50	26.87	39.89	8.00	6.95	25.16	11.30	21.68
	AdaSwitch	25.11	24.93	17.91	26.27	9.56	1.25	29.43	43.41	7.75	5.08	26.49	13.01	21.43
	TCOD	24.29	25.32	14.93	27.14	15.71	3.00	27.48	46.37	8.25	3.83	31.04	12.62	20.94
	Guided-OPD	27.14	24.50	19.40	25.90	19.63	3.75	24.63	48.44	9.75	4.80	34.04	15.01	19.96
1.7B	Zero-shot	8.10	28.35	7.10	27.93	6.48	0.75	24.10	28.71	10.25	9.11	17.60	6.55	22.37
	Vanilla OPD	23.57	25.45	26.87	24.54	16.94	3.00	25.27	47.59	11.00	5.54	32.27	16.11	20.20
	AdaSwitch	25.71	24.83	23.13	25.26	17.02	2.75	26.45	46.79	10.50	5.81	31.91	15.52	20.59
	TCOD	27.86	23.87	24.63	25.01	19.08	3.25	25.26	49.47	14.25	4.39	34.28	17.50	19.63
	Guided-OPD	32.86	23.34	29.85	23.60	23.24	5.25	23.84	52.63	11.75	4.14	37.94	19.93	18.73
4B	Zero-shot	28.57	24.25	27.86	24.73	16.21	3.75	15.97	20.59	8.25	12.23	18.40	17.11	19.30
	Vanilla OPD	31.43	23.36	32.09	23.34	26.45	7.50	23.88	50.87	13.50	4.53	38.66	21.13	18.78
	AdaSwitch	29.29	23.59	29.10	23.96	25.24	6.25	24.48	48.91	12.35	5.57	37.08	19.25	19.40
	TCOD	32.86	23.18	31.34	24.04	25.86	9.25	23.36	53.13	19.00	4.26	39.50	23.11	18.71
	Guided-OPD	37.14	22.76	35.08	23.15	29.50	11.50	22.86	55.78	17.75	4.28	42.64	25.37	18.26

Table 2: Ablation study on Qwen3-1.7B distilled from Qwen3-30B-A3B. All variants share the same training configuration as Guided-OPD; the three blocks ablate the asymmetric loss design, the curriculum schedule, and the decay shape, respectively. Score, SR, and Round are averaged across ALFWORLD, ScienceWorld, and WebShop following the protocol of Table 1.

Variant	Score $\uparrow$	SR $\uparrow$	Round $\downarrow$
Guided-OPD (Ours)	37.94	19.93	18.73
(a) Asymmetric loss design			
fKL on stu, rKL on tea	33.32	15.47	23.30
rKL on both turns	36.58	17.83	20.77
fKL on both turns	34.56	17.02	21.13
(b) Curriculum vs fixed mixing			
Fixed $\beta = 0.25$	34.41	16.09	19.97
Fixed $\beta = 0.50$	35.17	15.62	20.05
Fixed $\beta = 0.75$	35.85	14.82	19.36
(c) Decay shape			
Linear	36.12	17.01	20.45
Exp	34.44	15.79	19.58

4.3 Ablation Study

To quantify the contribution of each design choice in Guided-OPD, we ablate the three core components introduced in §3 on the Qwen3-1.7B student, keeping the teacher, benchmarks, hyperparameters, and evaluation script identical to the main experiments. The three blocks of Table 2 mirror the

bottom-up construction of the method: block (a) ablates the asymmetric loss assignment from Turn-level Guidance (§3.1), block (b) ablates whether β must follow a curriculum at all (§3.2), and block (c) ablates the shape of the decay curve. We discuss the three blocks in turn.

Reverse and forward KL are not interchangeable. We first examine the asymmetric loss assignment within a turn (Table 2(a)). Swapping the two KL roles—forward KL on student turns and reverse KL on teacher turns—drops Score and SR sharply from 37.94/19.93 to 33.32/15.47 and raises Round from 18.73 to 23.30, the worst Score and Round across all variants. This swap misuses both divergences at once: forward KL on student turns asks the student to mass-cover the teacher while it is actively driving the rollout, encouraging it to emit unfamiliar teacher actions and accelerating the trajectory drift identified in §1; reverse KL on teacher turns lets the student re-weight the demonstration through its current preferences, partially discarding the high-quality behaviors it is meant to inject. Reverse KL on both turns (36.58 Score, 17.83 SR) clearly beats forward KL on both (34.56, 17.02), confirming that tightening the student with mode-seeking reverse KL on its own trajectories matters more than forcing it to cover

the full teacher behavior. The default asymmetric pairing outperforms both symmetric variants, showing that matching each divergence to the role that drives generation is what makes the loss design effective.

The curriculum is not replaceable by fixed mixing. We then examine whether β needs to decay with training. Table 2(b) shows that all fixed- β variants trail the cosine schedule (Score 37.94) by 2.1–3.5 points, with an even larger SR gap: the best, fixed- $\beta = 0.25$, reaches only 16.09, 3.84 points below our 19.93. Strikingly, as fixed β grows from 0.25 to 0.75, Score keeps rising (34.41 $\rightarrow$ 35.17 $\rightarrow$ 35.85) while SR moves in the opposite direction (16.09 $\rightarrow$ 15.62 $\rightarrow$ 14.82). More teacher intervention is thus not always better: a persistent teacher exposure trains the student into an imitator that reproduces intermediate teacher behaviors—scoring higher under the partial-credit Score—but never operates on its own rollouts long enough to learn how to recover without a teacher hint, which costs it on the stricter binary SR and is mirrored by a Round that stays near 20 for every fixed β . The cosine curriculum resolves this dilemma by concentrating guidance in the early stage when the student is weakest and OOD drift most likely, and gradually withdrawing it as the student matures, achieving the best Score, SR, and Round simultaneously.

The cosine shape strikes the best trade-off. Finally, we compare the three decay shapes while keeping all other configurations fixed. As shown in Table 2(c), cosine (37.94 Score) strictly outperforms linear (36.12) and exp (34.44), with exp suffering the largest drop on both Score and SR. The cosine curve decays slowly near both endpoints and faster in the middle, matching the intuition of §3.2: retain a strong teacher anchor while the student is still fragile, transfer the bulk of supervision in the middle phase, and exit smoothly so that late training is indistinguishable from pure OPD. The linear schedule preserves moderate guidance throughout and is a reasonable but suboptimal backup. The exponential schedule, in contrast, decays β too aggressively in the early stage and removes teacher support precisely at the phase where OOD errors accumulate most easily, which explains its largest drop.

5 Related Work

On-Policy Distillation for Multi-turn Agents. Standard OPD (Agarwal et al., 2024; Lu and Thinking Machines Lab, 2025) excels on single-turn reasoning (Jin et al., 2026; Xu et al., 2026; Yang et al., 2026b) but suffers in multi-turn settings as early errors compound and degrade subsequent supervision (Ross et al., 2011; Fu et al., 2026; Li et al., 2026b); recent work either augments the token-level KL with reward-aligned signals from verifiable rewards or reward models (Zhong et al., 2026; Wang et al., 2026a; Li et al., 2026a), which depends on a reliable verifier or critic (Guo et al., 2025; Shao et al., 2024; Yu et al., 2025a), or schedules trajectory depth via a temporal curriculum (Wang et al., 2026b) that operates only at the macro level and does not repair the state distribution along trajectories the student has already entered. We instead intervene inside the rollout through explicit teacher guidance (Ross et al., 2011), pulling the state distribution back into the teacher-familiar region without any extra reward or critic.

Teacher Guidance during Distillation Rollout. A more direct line of work injects teacher guidance into the rollout itself (Xu et al., 2025; Peng et al., 2025), but existing methods sit at two unbalanced extremes: token-level bidirectional switching (Koo et al., 2025; Xu et al., 2025) alternates at every token and produces chimeric trajectories (Peng et al., 2025), while sequence-tail one-shot handover (Peng et al., 2025) transfers the remainder of generation to the teacher upon a single deviation and prevents the student from recovering from local errors at later decoding steps. Both keep the teacher present throughout training and leave a persistent training-inference mismatch when the teacher is removed at inference (Fu et al., 2026). Guided-OPD intervenes at the turn level, the granularity aligned with the environment’s state transitions (Xue et al., 2026; Zhou et al., 2024), and decays this intervention to zero by the end of training, so that the late stage returns to pure OPD (Agarwal et al., 2024) and the mismatch is eliminated.

6 Conclusion

We studied On-Policy Distillation in the multi-turn agent setting and identified a characteristic failure mode in which small student errors compound across turns and push the rollout out of the teacher’s familiar state distribution, so that the teacher’s su-

pervision becomes least reliable where the student needs it most. We addressed this with Guided-OPD, a simple yet effective algorithm that mixes teacher- and student-generated turns within a rollout and schedules the teacher’s intervention along a curriculum decaying smoothly to zero. Turn-level mixing aligns role transitions with environment state transitions, the asymmetric pairing of reverse and forward KL matches each divergence to the role driving generation, and the curriculum preserves the training-inference consistency. Across ALFWorld, ScienceWorld, and WebShop, and across Qwen3 students of three scales distilled from Qwen3-30B-A3B, Guided-OPD consistently outperforms vanilla OPD and strong teacher-guidance and curriculum baselines, with the largest gains on the smallest student. We hope the rollout-drift perspective and turn-level curriculum view will inform future work on transferring agentic capabilities to compact, deployable models.

Limitations

Our study has several limitations that point to natural directions for future work. First, the evaluation suite covers only three multi-turn agent benchmarks, namely ALFWorld, ScienceWorld, and WebShop, all of which expose the agent through purely text-based observations and a discrete action set. Although these environments already span embodied navigation, scientific reasoning, and e-commerce, they leave out a number of regimes that are increasingly important for practical agents, including tool-use agents with open-ended API calls, software engineering agents that operate on real codebases, GUI agents that perceive screenshots, and embodied agents that act in continuous physical environments. Whether Guided-OPD remains effective under these richer observation and action spaces is an open question. Second, all distillation experiments are conducted within a single model family. We distill Qwen3 students of three different scales from the same Qwen3-30B-A3B teacher, so the conclusions reflect intra-family transfer with aligned tokenizers and similar inductive biases. Cross-family distillation, for example from a Qwen teacher to a Llama or Mistral student, may introduce tokenizer mismatches and distributional gaps that further amplify the rollout-drift problem and possibly require adjustments to the curriculum schedule; combining Guided-OPD with recent cross-tokenizer distillation techniques (Sun

et al., 2026) is a promising direction. Third, the curriculum hyperparameters, including the initial mixing probability β_{start} , the curriculum ratio ρ , and the cosine decay shape, are chosen once and shared across all student sizes and benchmarks. In contrast, a schedule that adapts β to a measured training signal rather than following a predefined curve may reduce this tuning cost and raise the performance ceiling. Finally, our experiments are run on a single 8-GPU node with students up to 4B parameters and a 30B-A3B teacher. Verifying the trends at substantially larger student or teacher scales, and on longer-horizon tasks where rollout drift is likely to be more severe, requires resources beyond those available for this study and is left to future work.

Acknowledgments

This work was supported by the Prevention and Control of Emerging and Major Infectious Diseases–National Science and Technology Major Project (No. 2025ZD01901600) and the Beijing Natural Science Foundation (No. L252035).

We disclose that generative AI assistants (Cursor and Claude) were used in two ways during this work: to assist with coding while implementing the training and evaluation pipeline, and to polish the language of the manuscript.

References

- Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. 2024. [On-policy distillation of language models: Learning from self-generated mistakes](#). In *The Twelfth International Conference on Learning Representations*.
- Kevin Chen, Marco Cusumano-Towner, Brody Huval, Aleksei Petrenko, Jackson Hamburger, Vladlen Koltun, and Philipp Krähenbühl. 2025. Reinforcement learning for long-horizon interactive llm agents. *arXiv preprint arXiv:2502.01600*.
- Yuqian Fu, Haohuan Huang, Kaiwen Jiang, Jiacai Liu, Zhuo Jiang, Yuanheng Zhu, and Dongbin Zhao. 2026. [Revisiting on-policy distillation: Empirical failure modes and simple fixes](#). In *Third Conference on Language Modeling*.
- Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. [MiniLLM: Knowledge distillation of large language models](#). In *The Twelfth International Conference on Learning Representations*.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang,

- Shirong Ma, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Wenjin Hou, Shangpin Peng, Weinong Wang, Zheng Ruan, Yue Zhang, Zhenglin Zhou, Mingqi Gao, Yifei Chen, Kaiqi Wang, Hongming Yang, et al. 2026. Uni-opd: Unifying on-policy distillation with a dual-perspective recipe. *arXiv preprint arXiv:2605.03677*.
- Woogyel Jin, Taywon Min, Yongjin Yang, Dennis Wei, Yi Zhou, Swanand Ravindra Kadhe, Nathalie Baracaldo, and Kimin Lee. 2026. [Entropy-aware on-policy distillation of language models](#). In *Forty-third International Conference on Machine Learning*.
- Minki Kang, Jongwon Jeong, Seanie Lee, Jaewoong Cho, and Sung Ju Hwang. 2025. [Distilling LLM agent into small models with retrieval and code tools](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Jahyun Koo, Yerin Hwang, Yongil Kim, Taegwan Kang, Hyunkyung Bae, and Kyomin Jung. 2025. [SWITCH: Studying with teacher for knowledge distillation of large language models](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 3733–3746, Albuquerque, New Mexico. Association for Computational Linguistics.
- Gengsheng Li, Tianyu Yang, Junfeng Fang, Mingyang Song, Mao Zheng, Haiyun Guo, Dan Zhang, Jinqiao Wang, and Tat-Seng Chua. 2026a. [Unifying group-relative and self-distillation policy optimization via sample routing](#). In *Third Conference on Language Modeling*.
- Weizhen Li, Jianbo Lin, Zhuosong Jiang, Jingyi Cao, Xinpeng Liu, Jiayu Zhang, Zhenqiang Huang, Qianben Chen, Weichen Sun, Qiexiang Wang, et al. 2025. Chain-of-agents: End-to-end agent foundation models via multi-agent distillation and agentic rl. *arXiv preprint arXiv:2508.13167*.
- Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan-ang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. 2026b. [Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe](#). In *ICML 2026 Workshop on Foundations of Deep Generative Models: Understanding Memorization, Generalization, and Reasoning*.
- Jun Liu, Zhenglun Kong, Peiyan Dong, Changdi Yang, Tianqi Li, Hao Tang, Geng Yuan, Wei Niu, Wenbin Zhang, Pu Zhao, et al. 2025. Structured agent distillation for large language model. *arXiv preprint arXiv:2505.13820*.
- Kevin Lu and Thinking Machines Lab. 2025. [On-policy distillation](#). *Thinking Machines Lab: Connectionism*. <https://thinkingmachines.ai/blog/on-policy-distillation>.
- Feng Luo, Yu-Neng Chuang, Guanchu Wang, Zicheng Xu, Xiaotian Han, Tianyi Zhang, and Vladimir Braverman. 2026. [Demystifying OPD: Length inflation and stabilization strategies for large language models](#). In *Third Conference on Language Modeling*.
- Xuchen Pan, Yanxi Chen, Yushuo Chen, Yuchang Sun, Daoyuan Chen, Wenhao Zhang, Yuexiang Xie, Yilun Huang, Yilei Zhang, Dawei Gao, et al. 2025. Trinity-rft: A general-purpose and unified framework for reinforcement fine-tuning of large language models. *arXiv preprint arXiv:2505.17826*.
- Jingyu Peng, Maolin Wang, Hengyi Cai, Yuchen Li, Kai Zhang, Shuaiqiang Wang, Dawei Yin, and Xiangyu Zhao. 2025. Adaswitch: Balancing exploration and guidance in knowledge distillation via adaptive switching. *arXiv preprint arXiv:2510.07842*.
- Jiahao Qiu, Xinzhe Juan, Yimin Wang, Ling Yang, Xuan Qi, Tongcheng Zhang, Jiacheng Guo, Yifu Lu, Zixin Yao, Hongru Wang, et al. 2025. Agentdistill: Training-free agent distillation with generalizable mcp boxes. *arXiv preprint arXiv:2506.14728*.
- Stephane Ross, Geoffrey Gordon, and Drew Bagnell. 2011. [A reduction of imitation learning and structured prediction to no-regret online learning](#). In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 627–635, Fort Lauderdale, FL, USA. PMLR.
- Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.
- Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Cote, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2021. [ALFWorld: Aligning text and embodied environments for interactive learning](#). In *International Conference on Learning Representations*.
- Joykirat Singh, Raghav Magazine, Yash Pandya, and Akshay Nambi. 2025. Agentic reasoning and tool integration for llms via reinforcement learning. *arXiv preprint arXiv:2505.01441*.
- Mingyang Song and Mao Zheng. 2026. A survey of on-policy distillation for large language models. *arXiv preprint arXiv:2604.00626*.
- Jie Sun, Mao Zheng, Mingyang Song, Qiyong Zhong, Yilin Cheng, Bichuan Feng, Pengfei Liu, Junfeng Fang, and Xiang Wang. 2026. Simct: Recovering lost supervision for cross-tokenizer on-policy distillation. *arXiv preprint arXiv:2605.07711*.

- Hao Wang, Guozhi Wang, Han Xiao, Yufeng Zhou, Yue Pan, Jichao Wang, Ke Xu, Yafei Wen, Xiaohu Ruan, Xiaoxin Chen, et al. 2026a. Skill-sd: Skill-conditioned self-distillation for multi-turn llm agents. *arXiv preprint arXiv:2604.10674*.
- Jiaqi Wang, Wenhao Zhang, Weijie Shi, Yaliang Li, and James Cheng. 2026b. [Exploring temporal curriculum in on-policy distillation for multi-turn autonomous agents](#). In *Third Conference on Language Modeling*.
- Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. [Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2609–2634, Toronto, Canada. Association for Computational Linguistics.
- Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. 2022. [ScienceWorld: Is your agent smarter than a 5th grader?](#) In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11279–11298, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. 2025. The rise and potential of large language model based agents: A survey. *Science China information sciences*, 68(2):121101.
- Bangjun Xiao, Bingquan Xia, Bo Yang, Bofei Gao, Bowen Shen, Chen Zhang, Chenhong He, Chiheng Lou, Fuli Luo, Gang Wang, et al. 2026. Mimo-v2-flash technical report. *arXiv preprint arXiv:2601.02780*.
- Jiajun Xu, Zhiyuan Li, Wei Chen, Qun Wang, Xin Gao, Qi Cai, and Ziyuan Ling. 2024a. On-device language models: A comprehensive review. *arXiv preprint arXiv:2409.00088*.
- Wenda Xu, Rujun Han, Zifeng Wang, Long Le, Dhruv Madeka, Lei Li, William Yang Wang, Rishabh Agarwal, Chen-Yu Lee, and Tomas Pfister. 2025. [Speculative knowledge distillation: Bridging the teacher-student gap through interleaved sampling](#). In *The Thirteenth International Conference on Learning Representations*.
- Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. 2024b. A survey on knowledge distillation of large language models. *arXiv preprint arXiv:2402.13116*.
- Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, Zhipeng Wang, and Alborz Geramifard. 2026. Tip: Token importance in on-policy distillation. *arXiv preprint arXiv:2604.14084*.
- Zhenghai Xue, Longtao Zheng, Qian Liu, Yingru Li, Xiaosen Zheng, Zejun MA, and Bo An. 2026. [SimpleTIR: End-to-end reinforcement learning for multi-turn tool-integrated reasoning](#). In *The Fourteenth International Conference on Learning Representations*.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. 2026a. Self-distilled rlvr. *arXiv preprint arXiv:2604.03128*.
- Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. 2026b. Learning beyond teacher: Generalized on-policy distillation with reward extrapolation. *arXiv preprint arXiv:2602.12125*.
- Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757.
- 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*.
- Qiyong Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gao-hong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, and 17 others. 2025a. [DAPO: An open-source LLM reinforcement learning system at scale](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Zhaochen Yu, Ling Yang, Jiaru Zou, Shuicheng Yan, and Mengdi Wang. 2025b. Demystifying reinforcement learning in agentic reasoning. *arXiv preprint arXiv:2510.11701*.
- Qiyong Zhong, Mao Zheng, Mingyang Song, Xin Lin, Jie Sun, Houcheng Jiang, Xiang Wang, and Junfeng Fang. 2026. Sod: Step-wise on-policy distillation for small language model agents. *arXiv preprint arXiv:2605.07725*.
- Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. 2024. [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.

Siqi Zhu, Xuyan Ye, Hongyu Lu, Weiye Shi, and Ge Liu. 2026. The many faces of on-policy distillation: Pitfalls, mechanisms, and fixes. *arXiv preprint arXiv:2605.11182*.

A Experimental Setup Details

This appendix provides the full experimental details that complement §4.1. The five subsections describe the benchmark environments (§A.1), the evaluation metrics (§A.2), the baselines (§A.3), the training and evaluation hyperparameters (§A.4), and the prompt templates used for each environment (§A.5).

A.1 Benchmark Environments

We evaluate Guided-OPD on three text-based multi-turn interactive environments that cover three complementary task families: embodied navigation, scientific reasoning, and e-commerce. The dataset-level statistics are summarized in Table 3.

ALFWorld. ALFWorld (Shridhar et al., 2021) is a text-based embodied environment that requires the agent to complete six categories of household tasks: picking a target object and placing it at a destination; picking and placing after cleaning, heating, or cooling the object; examining an object under a light source; and picking and placing two objects of the same type. The agent perceives the environment through natural-language observations and acts through high-level commands such as moving to a location, opening containers, taking and putting objects, heating, cooling, and examining items, with a maximum of 30 interaction steps per episode. ALFWorld provides two official evaluation splits: the *seen* split shares room layouts and object combinations with the training distribution and serves as our **IID** evaluation, while the *unseen* split contains novel room layouts and object combinations never observed during training and serves as our **OOD** evaluation. The OOD split therefore probes whether the policy learned by Guided-OPD generalizes beyond the specific environments encountered during distillation.

ScienceWorld. ScienceWorld (Wang et al., 2022) is a text-based interactive environment that tests scientific reasoning across 30 task types aligned with an elementary-school science curriculum, covering topics such as thermodynamics, electric circuits, chemistry, biology, and matter classification. The environment exposes a large action space (e.g., focus on, activate, mix, wait, move . . . to

Table 3: Summary of the benchmarks used. “Max Turns” is the maximum number of agent–environment interaction rounds per task.

Benchmark	Type	Difficulty	Max Turns
ALFWorld (IID)	Embodied	Easy	30
ALFWorld (OOD)	Embodied	Medium	30
ScienceWorld	Scientific Reasoning	Hard	30
WebShop	E-commerce	Medium	15

. . .) together with a structured observation that lists the agent’s current location, visible objects, and inventory. Each episode allows at most 30 interaction steps, and at the end of each episode the environment returns a continuous score in $[0, 100]$ that reflects the fraction of subgoals completed. ScienceWorld is the most challenging benchmark in our suite because successful trajectories typically require long-horizon planning, multi-step tool use, and a non-trivial chain of cause–effect reasoning.

WebShop. WebShop (Yao et al., 2022) is a simulated e-commerce environment in which the agent must satisfy a natural-language shopping instruction (e.g., specific product type, color, size, and price range) by interacting with a structured web interface. The agent observes a rendered HTML page that includes a search bar, product listings, and product detail pages, and acts through two action primitives: `search[query]` for issuing search queries and `click[button_name]` for navigating links, applying filters, or completing a purchase. Each episode allows at most 15 interaction steps. After purchase, the environment returns a continuous score in $[0, 100]$ that measures how well the purchased product matches the requested attributes, and a binary success signal when the score reaches 100.

A.2 Metrics

We report three metrics in a unified manner across the three benchmarks.

Score ($\uparrow$). The continuous task score natively provided by ScienceWorld and WebShop, bounded in $[0, 100]$. On ScienceWorld it is computed from the fraction of subgoals completed during the episode, and on WebShop it is computed from the attribute-level match between the purchased product and the requested specification. Score offers a partial-credit, fine-grained measurement that lets us distinguish between failures that are close to completion and failures that are entirely off the mark. ALFWorld only provides a binary success signal, so

Score is not reported on it.

Success Rate (SR $\uparrow$). The task completion rate, treated as a binary outcome on every task and bounded above by $\text{SR} = 100\%$ and below by 0% . On ALFWorld an episode is counted as successful when the environment emits a terminal success flag; on ScienceWorld and WebShop, where an episode may terminate without full credit, an episode is counted as successful only when it receives a Score of 100.

Round ($\downarrow$). The average number of environment interaction steps consumed per task, which reflects the efficiency of the learned policy. When SR is similar across methods, a smaller Round is preferred because it indicates that the policy completes the task with fewer interactions.

For ALFWorld we additionally report SR and Round separately on the IID and OOD splits, so that in-distribution performance and OOD generalization are measured at the same time. All metrics are averaged over the full validation set of each benchmark, and all methods use the same decoding configuration to ensure a fair comparison.

A.3 Baselines

We benchmark Guided-OPD against four baselines that span training-free evaluation, the standard OPD paradigm, the teacher-guidance family, and the curriculum-learning family. All baselines share the same teacher (Qwen3-30B-A3B), the same training hyperparameters (Table 4), and the same evaluation script as Guided-OPD to ensure a fair comparison.

Zero-Shot Student (Lower Bound). The untrained student π_θ evaluated directly on the interactive tasks without any task-specific fine-tuning or distillation. This establishes the absolute starting point of the student’s capability in each environment, and serves as a sanity-check lower bound for all distillation methods.

Vanilla OPD. The standard multi-turn adaptation of recent OPD methods (Agarwal et al., 2024; Lu and Thinking Machines Lab, 2025), defined in §2. The student samples the entire trajectory by itself and the teacher KL is applied at every token of the agent-generated segments, with no teacher intervention during rollout. This serves as the direct counterpart against which we measure the effect of introducing turn-level teacher guidance and a curriculum schedule.

AdaSwitch. AdaSwitch (Peng et al., 2025) is a recent representative of the teacher-guidance family. It maintains a sliding-window average $\bar{d}$ of the token-level student–teacher divergence over the most recent L steps and, the first time the instantaneous divergence exceeds the adaptive threshold $\tau = K\bar{d}$, hands the remainder of the current generation to the teacher; the handover is one-shot and is never reverted, so no further divergence checks are performed. We follow its recommended settings, with threshold multiplier $K = 3$ and sliding window length $L = 10$. The switching is fully triggered by sample-level divergence signals and is independent of training progress, so there is no curriculum tied to the training step.

TCOD. TCOD (Wang et al., 2026b) is a recent curriculum-based method tailored to multi-turn agent OPD. It uses the rollout depth k , that is, the maximum number of interaction turns the student is allowed to take before the trajectory is truncated, as a proxy for difficulty, and increases k linearly with the training step n via the fixed pacing schedule $k_n = \min(T_{\max}, k_{\text{start}} + \lfloor n/\eta \rfloor)$, so that the student gradually expands from short-horizon decisions to full trajectories and the KL is accordingly accumulated over these k turns. We follow its recommended fixed pacing with $\eta = 2$ and $k_{\text{start}} = 1$, and adopt its F2B variant because it does not require pre-collected successful teacher trajectories. The curriculum acts purely on trajectory depth, and no turn-level teacher intervention is introduced inside the rollout.

A.4 Hyperparameters

We conduct training across the three text-based interactive environments described in §A.1. The training configuration shared by all methods is summarized in Table 4, where the last two blocks list the hyperparameters specific to Guided-OPD and to the two baselines that have method-specific hyperparameters (AdaSwitch, TCOD). Evaluation hyperparameters are summarized in Table 5.

A.5 Prompt Templates

This subsection lists the prompt templates used for the three environments during both training and evaluation. All prompts follow a consistent structure: task description, observation–action history, current observation, admissible actions, and a thinking/action format requirement. The agent is required to first produce its reasoning inside

Table 4: Training hyperparameters across all environments. The left block lists the configuration shared by all methods; the right block lists the system-level setup and the hyperparameters specific to Guided-OPD and to the two baselines that have method-specific hyperparameters (AdaSwitch, TCOD).

Hyperparameter	Value	Hyperparameter	Value
Algorithm		Distributed Training	
Algorithm type	On-Policy Distillation	Number of nodes	1
KL coefficient	1.0	GPUs per node	8 (NVIDIA H20, 96 GB)
Learning rate	1×10^{-6}	GPU allocation	4 act. / 2 tea. / 2 lm.
Optimizer	AdamW	Tensor parallel size	2
Gradient clipping	1.0	Sequence parallel size	2 (Ulysses)
Repeat times	1	Max tokens per GPU	16,384
Sample strategy	Staleness ctrl ($\Delta_{\max} = 2$)	GPU memory utilization	0.7
Training		Data type	BFloat16
Total training steps	250	Guided-OPD (Ours)	
Batch size	16	Init. mixing β_{start}	1.0
Train batch size	64	Term. mixing β_{end}	0
Save interval	250	Curriculum ratio ρ	0.8
Evaluation interval	5 steps	Decay schedule	Cosine (Eq. 7)
Model Configuration		Student-turn loss	rKL (Eq. 4)
Max prompt tokens	10,240	Teacher-turn loss	fKL (Eq. 5)
Max response tokens	512	AdaSwitch (Peng et al., 2025)	
Inference (Rollout)		Sliding window length L	10
Temperature (training)	1.0	Threshold multiplier K	3
Temperature (evaluation)	0.4	TCOD (Wang et al., 2026b)	
Logprobs	Enabled (all tokens)	Variant	F2B
Seed	42	Starting horizon k_{start}	1
Environment-Specific		Pacing factor η	2
ALFWorld max steps	30		
ScienceWorld max steps	30		
WebShop max steps	15		

Table 5: Evaluation hyperparameters across all environments.

Hyperparameter	Value
Generation	
Maximum tokens	4,096
Temperature	0.4
Top-p / Top-k / MinP	1.0 / -1 / 0.0
Environment	
Max env. steps	30 (ALFW./SciW.) / 15 (WebS.)
History length	2 steps
Parallelization	
Number of workers	8
Process timeout	3,600 s
Synchronization style	Dynamic by explorer
Data Splits	
ALFWorld	test / test_unseen
ScienceWorld	Test split
WebShop	Test split

<thought> tags and then emit an executable command inside <action> tags, so that the reasoning and the environment-bound action are syntactically separable.

ALFWorld. ALFWorld is an embodied task that requires the agent to navigate household environ-

ments and complete object-manipulation goals. The prompt emphasizes step-by-step reasoning within <thought> tags followed by an executable action within <action> tags selected from a dynamically updated admissible-action set.

You are an expert agent operating in the ALFRED Embodied Environment. Your task is to: {task_description}

Prior to this step, you have already taken {step_count} step(s). Below are the most recent {history_length} observations and the corresponding actions you took: {action_history}

You are now at step {current_step} and your current observation is: {current_observation}

Your admissible actions of the current situation are: [{admissible_actions}].

Now it's your turn to take an action.

You should first reason step-by-step about the current situation. This reasoning process MUST be enclosed within <thought> tags.

Once you've finished your reasoning, you should choose an admissible action for the current step and present it within <action> </action> tags.

ScienceWorld. ScienceWorld focuses on scientific reasoning tasks in a text-based laboratory environment. The prompt guides the agent through multi-step experiments that require domain knowledge and procedural reasoning.

```
Your ScienceWorld task is:
{task_description}
Prior to this step, you have already
taken {step_count} step(s). Below
are the most recent {history_length}
observations and the corresponding
actions you took: {action_history}
You are now at step {current_step}
and your current observation is:
{current_observation}
Your valid actions of the current
situation are: [{admissible_actions}].

Now it's your turn to take an action.
You should first reason step-by-step
about the current situation. This
reasoning process MUST be enclosed
within <thought> tags.
Once you've finished your reasoning,
you should choose a valid action for
the current step and present it within
<action> </action> tags.
```

WebShop. WebShop presents e-commerce shopping tasks that require the agent to navigate product listings, apply filters, and make purchasing decisions according to natural-language instructions. The prompt emphasizes matching user preferences to available product attributes.

```
You are an expert autonomous agent
operating in the WebShop e-commerce
environment.
Your task is to: {task_description}.
Prior to this step, you have already
taken {step_count} step(s). Below
are the most recent {history_length}
observations and the corresponding
actions you took: {action_history}
You are now at step {current_step}
and your current observation is:
{current_observation}.
Your admissible actions of the current
situation are:
[
{available_actions}
].

Now it's your turn to take one action
for the current step.
You should first reason step-by-step
about the current situation, then think
carefully which admissible action best
advances the shopping goal. This
reasoning process MUST be enclosed
within <thought> tags.
Once you've finished your reasoning, you
should choose an admissible action for
the current step and present it within
<action> </action> tags.
```

Table 6: Teacher-takeover analysis on ALFWorld. Student prefixes of increasing length L are produced by an untrained Qwen3-1.7B. For each L we report the teacher PPL on the L -th student turn and the success rate the Qwen3-30B-A3B teacher attains when it takes over the episode from turn L onwards. All values are averaged over 200 trajectories.

Prefix length L	Teacher PPL	Takeover SR (%) $\uparrow$
1	1.24	42.5
5	1.42	38.0
10	1.53	31.5
15	1.57	24.5
20	1.68	16.0

WebShop uses a specific action format with two primary action types: `search[<query>]`, which issues a search using a free-text query and is only available when the search bar is present, and `click[<button_name>]`, which clicks an interactive element such as a product link, a filter button, or a pagination button. The set of available clicks is regenerated from the current page state at every step.

B Further Analyses of the Teacher-PPL Diagnostic

Figure 1(a) reports the teacher PPL on student-generated tokens as a function of the interaction turn, and §1 reads its upward slope as evidence of rollout-level distribution drift. This appendix examines that diagnostic from three angles. §B.1 tests whether a high teacher PPL indicates that the teacher’s supervision is genuinely less useful rather than merely less confident. §B.2 relates the teacher PPL of a trained student to its task performance across all methods in our main comparison. §B.3 evaluates whether, given that PPL grows along the trajectory, teacher guidance should be allocated by turn position instead of drawn uniformly within a training step. All experiments in this appendix use Qwen3-1.7B as the student and Qwen3-30B-A3B as the teacher.

B.1 Usefulness of Teacher Supervision in High-PPL States

To further examine the relationship between the teacher PPL and the quality of teacher supervision, we sample 200 ALFWorld trajectories of at least 20 turns from an untrained student and truncate each of them at prefix lengths $L \in \{1, 5, 10, 15, 20\}$, so that the underlying task set is held fixed across all L and only the amount of student-generated

Table 7: Mean teacher PPL and task performance of Qwen3-1.7B students trained by different methods, measured on ALFWorld. PPL is computed under the Qwen3-30B-A3B teacher over the same tasks for every method.

Student (Qwen3-1.7B)	PPL ↓	SR ↑	Round ↓
Zero-shot	1.51	7.60	28.14
Vanilla OPD	1.40	25.22	25.00
AdaSwitch	1.42	24.42	25.05
TCOD	1.34	26.25	24.44
Guided-OPD	1.27	31.36	23.47

history varies. For every L we record two quantities: the teacher PPL on the L -th student turn, computed in the same way as in Figure 1(a), and the *teacher-takeover success rate*, obtained by letting the teacher continue the episode from turn L until termination. Table 6 reports the results.

The two quantities move in opposite directions. As the student prefix lengthens and the teacher PPL rises, the success rate the teacher can still achieve falls steadily, from 42.5% after a single student turn to 16.0% after twenty. This is because the more turns the student reasons over, the more readily the rollout leaves the region familiar to the teacher, so that the task becomes hard to complete even when the teacher takes over. We therefore read the rising PPL as a symptom of the degrading quality of the teacher’s supervision.

B.2 Teacher PPL Versus Task Performance Across Methods

To further examine whether a lower teacher PPL is reflected in task performance, we re-measure the diagnostic across every method in our main comparison. Taking the Qwen3-1.7B students trained by each method, we run them on the same ALFWorld tasks and compute the mean teacher PPL under Qwen3-30B-A3B, averaging first within each trajectory and then across trajectories. Task metrics are the ALFWorld SR and Round of Table 1. Table 7 reports the results.

Ordering the five models by mean teacher PPL basically reproduces their ordering by SR and by Round, and Guided-OPD is best on all three. This is because a rollout that stays closer to the teacher’s familiar region receives more reliable supervision throughout training, which in turn yields a stronger and more efficient student.

Table 8: Position-reactive allocation of teacher guidance on Qwen3-1.7B. Turn u is assigned to the teacher with probability $\min(\beta_t \gamma_u, 1)$, where γ_u grows linearly with the turn index from γ_{start} to γ_{end} . Score, SR, and Round follow the averaging protocol of Table 2.

Schedule	Score ↑	SR ↑	Round ↓
Vanilla OPD	32.27	16.11	20.20
Guided-OPD (default)	37.94	19.93	18.73
$\gamma : 0.75 \rightarrow 1.25$	38.16	19.47	18.61
$\gamma : 0.50 \rightarrow 1.50$	37.13	18.24	19.97
$\gamma : 0.25 \rightarrow 1.75$	35.92	16.73	21.15

B.3 Position-Reactive Allocation of Teacher Guidance

Within a training step, Eq. 3 assigns every turn to the teacher with the same probability β_t , so guidance follows training progress but not the position of a turn in the trajectory. To further examine whether guidance should instead be concentrated at the later turns where the teacher PPL is higher, we make the mixing probability position-dependent.

Gating on PPL directly is not possible during rollout, since the teacher PPL of a turn is defined over the tokens the student produces and can only be evaluated once that turn has been generated; we therefore use the position of a turn as a proxy, exploiting the empirical regularity that PPL grows with the turn index. Leaving the curriculum and the asymmetric loss of §3 untouched, we assign turn u to the teacher with probability $\min(\beta_t \gamma_u, 1)$, where γ_u takes the value γ_{start} at the first turn, grows linearly with the turn index, and is clipped at γ_{end} once the turn index reaches the mean number of interaction rounds that Guided-OPD attains on that benchmark. Placing γ_{start} and γ_{end} symmetrically about one keeps the average amount of teacher guidance close to that of the default schedule, so that the comparison isolates how guidance is distributed rather than how much of it is spent. Table 8 reports three increasingly aggressive settings.

Reallocation trades one metric against another. The setting $\gamma_{\text{start}} = 0.75$, $\gamma_{\text{end}} = 1.25$ raises Score by 0.22 and shortens Round by 0.12, yet loses 0.46 in SR. This is because extra guidance at the error-prone tail lets the student pick up more of the teacher’s high-quality action patterns, which the partial-credit Score rewards, while at the same time making it more reliant on the teacher and less able to recover from accumulated errors on its own, which the binary SR penalizes; the same seesaw appears for large fixed β in Table 2(b). Pushing the re-

allocation further degrades every metric, since the guidance withdrawn from the early turns is what curbs error accumulation at its root. We therefore retain the position-independent draw of Eq. 3 as the default: it attains the best SR while introducing no additional hyperparameters, which suggests that the benefit of turn-level guidance comes mainly from anchoring the rollout as a whole rather than from targeting individual high-PPL turns.