DASH-OPD: DISCREPANCY-AWARE SWITCHING WITH HYSTERESIS FOR ON-POLICY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

While on-policy distillation (OPD) reduces exposure bias by training student language models on their own rollouts, early student errors in long-horizon agentic scenarios can lead to contexts unfamiliar to the teacher. To improve trajectory quality, recent work on agentic OPD introduces teacher intervention into training rollouts by switching the executor between the student and the teacher. However, existing methods determine how much teacher intervention is needed—but not when. To address this limitation, we propose DASH-OPD (Discrepancy-Aware Switching with Hysteresis for OPD), the first agentic OPD method to perform adaptive, bidirectional executor switching. At each turn, DASH-OPD measures teacher–student discrepancy using a mean log-likelihood ratio over action tokens. High student-to-teacher ratios on student turns serve as drift signals, while low teacher-to-student ratios on teacher turns serve as recovery signals. These signals are accumulated over multiple turns to form drift and recovery evidence, respectively. DASH-OPD switches executors when either type of evidence exceeds its corresponding switching threshold, introducing hysteresis that prevents rapid switching triggered by transient discrepancy fluctuations. Across three benchmarks and two student model scales, DASH-OPD outperforms five baselines in all 14 task-performance comparisons, while requiring the fewest interaction turns in nine of ten efficiency comparisons. Code and trained models are available in this anonymous repository: `https://anonymous.4open.science/r/DASH-OPD-7C3D`.[1]

1 INTRODUCTION

Deploying capable large language models (LLMs) can be computationally expensive, motivating knowledge distillation to transfer the capabilities from teacher models to more efficient student models (Hinton et al., 2015; Gu et al., 2024). Conventional distillation trains the student on teacher-generated (Peng et al., 2023) or human-provided (Wu et al., 2025) trajectory data, whereas during inference the student must condition on self-generated contexts, thereby creating a training–inference mismatch (Ross et al., 2011), known as exposure bias (Ranzato et al., 2015). On-policy distillation (OPD) (Agarwal et al., 2024) mitigates exposure bias by training students on self-generated trajectories (Song & Zheng, 2026; Zhang, 2026). However, this benefit can diminish in long-horizon agentic tasks: accumulated student errors steer trajectories into contexts unfamiliar to the teacher (Zhou et al., 2026a), thereby degrading the reliability of teacher supervision (Zhong et al., 2026; Fu et al., 2026; Zhu et al., 2026). In our experiments, vanilla OPD marginally improves the 4B student's success rate on the ScienceWorld benchmark (Wang et al., 2022), from 16.21% to 16.36% (see Table 1).

Teacher-generated turns can recover trajectories from student-induced drift, but excessive teacher intervention reintroduces exposure bias. Recent work on agentic OPD uses predefined curricula (Bengio et al., 2009) to balance teacher and student control, as illustrated in Fig. 1. TCOD (Wang et al., 2026) follows a deterministic curriculum that progressively lengthens a contiguous student-controlled segment in each trajectory. Guided-OPD (Li et al., 2026) follows a probabilistic curriculum to sample the executor at each turn, with the probability of selecting the teacher gradually decreasing over training. However, existing methods determine *how much* teacher intervention is needed, but not *when*, even though its necessity may vary within and across trajectories (Gao et al., 2026; Cai et al., 2025; Hoque et al., 2021).

[1]Training logs for DASH-OPD and all baselines will also be publicly released after the review process.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

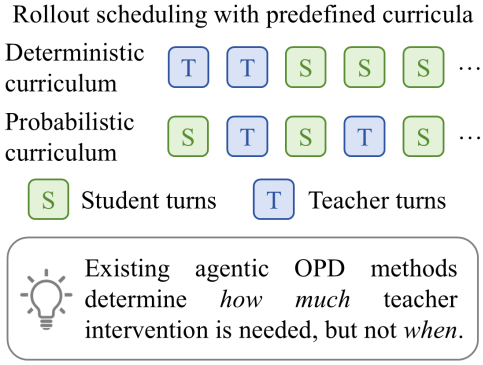

Figure 1: A research gap in agentic OPD.

Teacher–student discrepancy can indicate when the teacher should intervene and when the student can resume control. We therefore propose DASH-OPD (Discrepancy-Aware Switching with Hysteresis for OPD), as illustrated in Fig. 2. At each turn, either the student or the teacher generates a response, and the other model scores the action tokens in this response under the same context, yielding a mean log-likelihood ratio that quantifies teacher–student discrepancy. On student turns, high student-to-teacher ratios serve as drift signals. On teacher turns, low teacher-to-student ratios serve as recovery signals. These two types of signals are accumulated separately across turns to form drift and recovery evidence. Drift evidence tracks the need for teacher intervention, while recovery evidence tracks whether the student can resume control. DASH-OPD switches control to the teacher when drift evidence exceeds an intervention threshold and back to the student when recovery evidence exceeds a return threshold. Following Guided-OPD, DASH-OPD trains the student on all response tokens using reverse-KL distillation on student turns and forward-KL distillation on teacher turns.

DASH-OPD achieves hysteretic switching by acting on evidence accumulated across multiple turns rather than on individual signals. Requiring sufficient evidence before switching makes DASH-OPD robust to transient fluctuations in teacher–student discrepancy, thereby preventing rapid oscillation between the two models.

To the best of our knowledge, DASH-OPD is the *first* agentic OPD method to perform adaptive, bidirectional executor switching. This mechanism enables DASH-OPD to invoke teacher intervention when needed. We evaluate DASH-OPD with 1.7B and 4B student models on WebShop (Yao et al., 2022a), ALFWorld (Shridhar et al., 2020), and ScienceWorld (Wang et al., 2022). DASH-OPD outperforms five baselines in all 14 task-performance comparisons, while requiring the fewest interaction turns in nine of ten efficiency comparisons (second fewest on ScienceWorld at 1.7B).

2 Related Work

To balance trajectory quality with the student's exposure to self-generated contexts during training, recent OPD methods schedule rollouts at either turn or token granularity. Across both granularities, some methods include teacher-generated segments in training rollouts, whereas others keep the rollouts entirely student-generated.

Turn-level rollout scheduling. In agentic OPD, each turn forms a complete decision unit, making turn-level rollout scheduling a natural choice. TCOD (Wang et al., 2026) and Guided-OPD (Li et al., 2026) both include teacher-generated turns and regulate intervention through predefined curricula. TCOD follows a deterministic curriculum that gradually expands the student rollout horizon. Guided-OPD follows a probabilistic curriculum that samples either the teacher or the student to act at each turn, with the probability of choosing the teacher gradually decreasing over training. TurnOPD (Zhou et al., 2026b) uses student-only rollouts and adapts the rollout horizon based on turn-level statistics.

Token-level rollout scheduling. For general text-generation tasks such as coding and mathematical reasoning, rollout scheduling naturally operates at the token level. AdaSwitch (Peng et al., 2025) uses a context-dependent threshold to trigger a student-to-teacher switch, after which the teacher completes the response. POPD and TOPD (Zhang et al., 2026) both use student-generated trajectories without teacher intervention. They implement two forms of rollout scheduling: POPD follows a predetermined curriculum that progressively expands the student rollout horizon over training, whereas TOPD keeps the horizon fixed. ESR (Ziheng et al., 2026) likewise limits each student rollout to a fixed number of tokens.

However, it remains an open challenge to determine when the teacher should intervene and when the student should resume control.

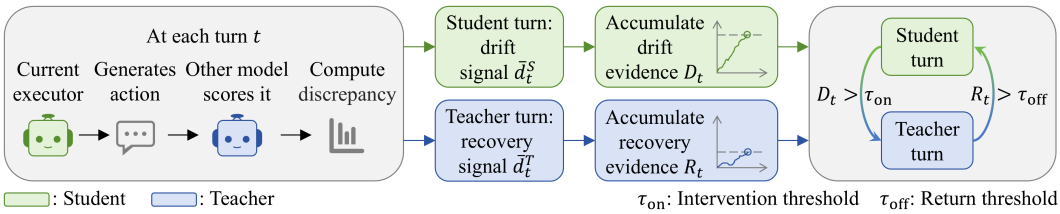

Figure 2: Pipeline of DASH-OPD's switching mechanism. DASH-OPD adaptively switches control between the student and teacher based on their discrepancy.

3 PRELIMINARIES

Large language model (LLM) agents. Rather than generating a single response to a fixed prompt, an LLM agent interacts with an environment over multiple turns (Yao et al., 2022b; Liu et al., 2024). At turn t, the agent conditions its response y_t on the interaction history $h_t = (x, o_1, y_1, \ldots, o_t)$, where x denotes the task prompt and o_t denotes the agent's observation. Executing the action parsed from y_t yields the next observation o_{t+1}. The response y_t is a token sequence $(y_{t,1}, \ldots, y_{t,L_t})$, where L_t is the sequence length. The agent is modeled as a conditional policy $\pi(y_t \mid h_t)$:

$$\pi(y_t \mid h_t) = \prod_{i=1}^{L_t} \pi(y_{t,i} \mid h_t, y_{t,<i}), \quad y_{t,<i} = (y_{t,1}, \ldots, y_{t,i-1}). \tag{1}$$

On-policy distillation (OPD). Conventional knowledge distillation trains the student on trajectories provided by the teacher (Peng et al., 2023) or humans (Wu et al., 2025). During inference, however, the student must condition on self-generated contexts. This training–inference mismatch, known as exposure bias (Ranzato et al., 2015), limits the student's ability to recover from its own errors.

OPD mitigates exposure bias by performing distillation on student-generated trajectories (Agarwal et al., 2024). Given a trainable student policy π_θ and a frozen teacher policy π_T, two common distillation objectives are the forward- and reverse-KL divergences between the two policies. The forward-KL objective is:

$$\mathcal{L}_{\text{forward}} = \sum_t \mathbb{E}_{h_t \sim \rho^t_{\pi_\theta}} D_{\text{KL}}(\pi_T(\cdot \mid h_t) \,\|\, \pi_\theta(\cdot \mid h_t)) \,. \tag{2}$$

The reverse-KL objective is:

$$\mathcal{L}_{\text{reverse}} = \sum_t \mathbb{E}_{h_t \sim \rho^t_{\pi_\theta}} D_{\text{KL}}(\pi_\theta(\cdot \mid h_t) \,\|\, \pi_T(\cdot \mid h_t)) \,. \tag{3}$$

Here, $\rho^t_{\pi_\theta}$ denotes the student-induced context distribution at turn t. In practice, training trajectories are generated by a frozen student snapshot $\pi_{\bar{\theta}}$ to keep the rollout distribution fixed while π_θ is updated. The forward-KL objective is mode-covering, encouraging the student to capture diverse teacher behaviors, whereas the reverse-KL objective is mode-seeking, concentrating the student on high-probability teacher behaviors. At inference time, only the student policy is deployed.

4 METHOD

4.1 OVERVIEW

At each turn t, DASH-OPD selects either the student or the teacher to generate the response. Let $m_t \in \{S, T\}$ (S for student and T for teacher) indicate which model controls turn t. The response y_t is generated as follows:

$$y_t \sim \begin{cases} \pi_{\bar{\theta}}(\cdot \mid h_t), & m_t = S, \\ \pi_T(\cdot \mid h_t), & m_t = T. \end{cases} \tag{4}$$

The non-executing model scores the action tokens in y_t to compute a directional teacher–student discrepancy. DASH-OPD switches control between the two models based on the evidence accumulated from these discrepancies. The pipeline of DASH-OPD's switching mechanism is illustrated in Fig. 2. Appendix A.1 provides the pseudocode for DASH-OPD.

4.2 DIRECTIONAL TEACHER–STUDENT DISCREPANCY

If turn t is student-controlled, the teacher measures the discrepancy by computing a student-to-teacher mean log-likelihood ratio (i.e., a sample-based analogue of $D_{\mathrm{KL}}(\pi_{\bar{\theta}} \| \pi_T)$):

$$d_t^S = \frac{1}{N_t^A} \sum_i A_{t,i} [\log \pi_{\bar{\theta}}(y_{t,i} \mid h_t, y_{t,<i}) - \log \pi_T(y_{t,i} \mid h_t, y_{t,<i})], \quad y_t \sim \pi_{\bar{\theta}}, \tag{5}$$

where $A_{t,i}$ is an action mask (1 for tokens in the action content; 0 otherwise), and $N_t^A = \sum_i A_{t,i}$ is the number of action tokens. If turn t is teacher-controlled, the student measures the discrepancy in the opposite direction by computing a teacher-to-student mean log-likelihood ratio (i.e., a sample-based analogue of $D_{\mathrm{KL}}(\pi_T \| \pi_{\bar{\theta}})$):

$$d_t^T = \frac{1}{N_t^A} \sum_i A_{t,i} [\log \pi_T(y_{t,i} \mid h_t, y_{t,<i}) - \log \pi_{\bar{\theta}}(y_{t,i} \mid h_t, y_{t,<i})], \quad y_t \sim \pi_T. \tag{6}$$

The teacher–student discrepancy is measured only over the action portion of the response, focusing on decision-making rather than reasoning patterns (see Appendix A.10 for robustness to action extraction failures). Both d_t^S and d_t^T are directly computed from the log-probabilities collected during rollout scoring, requiring no further model forward passes for switching. Appendix A.3 provides a detailed rationale for the directionality of teacher–student discrepancy.

4.3 FROM DRIFT AND RECOVERY SIGNALS TO EVIDENCE

DASH-OPD detects a drift signal when d_t^S rises above its historical level, and detects a recovery signal when d_t^T falls below its historical level. To track these historical levels, DASH-OPD maintains running statistics $\mathcal{R}_t^S = (\mu_t^S, v_t^S)$ for d_t^S and $\mathcal{R}_t^T = (\mu_t^T, v_t^T)$ for d_t^T, updated on student and teacher turns, respectively. Here, $\mu_t^{m_t}$ and $v_t^{m_t}$ denote the mean and variance of observed $d_t^{m_t}$. These statistics are retained throughout each trajectory without resetting. To obtain drift and recovery signals, DASH-OPD standardizes $d_t^{m_t}$ using the corresponding statistics:

$$\bar{d}_t^{m_t} = \frac{d_t^{m_t} - \mu_{t-1}^{m_t}}{\sqrt{v_{t-1}^{m_t}}}, \quad m_t \in \{S, T\}. \tag{7}$$

Here, $\bar{d}_t^S$ denotes the drift signal, and $\bar{d}_t^T$ denotes the recovery signal. On the first turn of each executor in a trajectory, $\bar{d}_t^{m_t}$ is set to zero.

DASH-OPD accumulates $\bar{d}_t^S$ and $\bar{d}_t^T$ separately across turns to form drift evidence D_t and recovery evidence R_t. On student turns, DASH-OPD updates D_t as follows:

$$D_t = \max(D_{t-1} + \bar{d}_t^S, 0), \quad m_t = S. \tag{8}$$

On teacher turns, DASH-OPD updates R_t as follows:

$$R_t = \max(R_{t-1} - \bar{d}_t^T, 0), \quad m_t = T. \tag{9}$$

Both D_t and R_t are initialized to zero for each trajectory. Positive $\bar{d}_t^S$ increases D_t, whereas negative $\bar{d}_t^T$ increases R_t. Signals in the opposite directions reduce the respective evidence toward zero.

4.4 SWITCHING MECHANISM

DASH-OPD switches control to the teacher when the drift evidence D_t exceeds an intervention threshold τ_{on}, and returns control to the student when the recovery evidence R_t exceeds a return threshold τ_{off}. To account for differences in rollout horizon across benchmarks, we scale both thresholds by $H_{\max}$, the maximum number of turns per trajectory:

$$\tau_{\mathrm{on}} = q_{\mathrm{on}} H_{\max}, \quad \tau_{\mathrm{off}} = q_{\mathrm{off}} H_{\max}, \tag{10}$$

where q_{on} and q_{off} are the intervention and return threshold ratios, respectively. In addition to the evidence-based switching, DASH-OPD also triggers teacher intervention when the student stagnates and enforces a minimum length of teacher-controlled segments.

Specifically, after a student turn, the condition for teacher intervention is:

$$\text{intervention}_t = (D_t > \tau_{\text{on}}) \vee g_t. \tag{11}$$

Here, g_t denotes a stagnation flag that is activated when the same action is taken or the same observation is encountered for K consecutive student turns. If intervention_t holds, DASH-OPD sets $m_{t+1} = T$ and resets the drift evidence D_t to zero. After a teacher turn, the condition for returning control to the student is:

$$\text{return}_t = (R_t > \tau_{\text{off}}) \wedge (\ell_t \geq \ell_{\min}). \tag{12}$$

Here, ℓ_t denotes the length of the current teacher-controlled segment, and $\ell_{\min}$ specifies the minimum length for teacher segments. If return_t holds, DASH-OPD sets $m_{t+1} = S$ and resets the recovery evidence R_t to zero. See Appendix A.11 for handling sudden increases in task difficulty.

The switching threshold ratios q_{on} and q_{off}, the minimum teacher segment length $\ell_{\min}$, and the stagnation threshold K are held fixed across all benchmarks and student model scales.

Rollout initialization. For each trajectory, DASH-OPD randomly selects either the student or the teacher to control the first turn. The teacher-start probability decreases linearly as training progresses:

$$p_T(s) = 1 - s/S_{\max}, \tag{13}$$

where s is the current training step and $S_{\max}$ is the total number of training steps. This initialization encourages more teacher intervention early in training and gradually reduces it as the student improves.

4.5 Distillation objective

Following Guided-OPD (Li et al., 2026), we train the student using sampled reverse-KL distillation on student-controlled turns and sampled forward-KL distillation on teacher-controlled turns. Let $\mathcal{L}^S_{t,i}$ and $\mathcal{L}^T_{t,i}$ denote the respective training losses for token i at turn t (detailed in Appendix A.2), and define $\mathbb{I}^S_t = \mathbb{I}[m_t = S]$ and $\mathbb{I}^T_t = \mathbb{I}[m_t = T]$. The overall objective is:

$$\mathcal{L}_{\text{overall}} = \frac{\sum_{t,i} M_{t,i} \left(\mathbb{I}^S_t \mathcal{L}^S_{t,i} + \mathbb{I}^T_t \mathcal{L}^T_{t,i} \right)}{\max\left(\sum_{t,i} M_{t,i}, 1 \right)}, \tag{14}$$

where $M_{t,i}$ equals 1 for valid response tokens and 0 otherwise. The distillation is applied to the full response, including both reasoning and action tokens, whereas the executor switching relies on the action-only discrepancy.

5 Experiments

5.1 Experimental setup

Baseline methods. We compare DASH-OPD against five baselines: vanilla OPD (Agarwal et al., 2024); a general-purpose OPD method, AdaSwitch (Peng et al., 2025); and three agentic OPD methods, TCOD (Wang et al., 2026), Guided-OPD (Li et al., 2026), and TurnOPD (Zhou et al., 2026b). For TCOD, we report the average performance of its B2F and F2B variants (individual results are provided in Appendix A.5). We adapt AdaSwitch to agentic OPD by applying its token-level switching rule independently within each turn. We exclude POPD/TOPD (Zhang et al., 2026) and ESR (Ziheng et al., 2026), as their token-level rollout scheduling may truncate a turn before its final action block. We distill a frozen Qwen3-30B-A3B teacher into Qwen3-1.7B and Qwen3-4B students (Yang et al., 2025). The zero-shot results for the teacher and student models are reported for reference but excluded from the rankings. To ensure a fair comparison, all baselines and DASH-OPD use the same settings for shared training and evaluation components (see Table 5 in Appendix A.13). For most settings specific to each baseline, we follow the original papers or official implementations (see Table 6 in Appendix A.13). For the few baseline settings that require adaptation to our experimental setup, we use the same tuning budget as for DASH-OPD.

Table 1: Task performance and deployment efficiency across three benchmarks and two student model scales. For ALFWorld, "Overall" reports results over all tasks in the "IID" and "OOD" test splits. Higher "Score" and "SR" indicate better task performance, while fewer "Turns" indicate greater efficiency. Results for the zero-shot teacher and students are shown for reference and excluded from the rankings. Within each model scale, **blue bold** and underline mark the best and second-best results, respectively (ties share the same style). DASH-OPD leads all 14 task-performance comparisons and requires the fewest turns in 9 of 10 efficiency comparisons (ranking second in the remaining one).

Method	WebShop			ALFWorld						ScienceWorld		
				IID		OOD		Overall				
	Score ↑	SR ↑	Turns ↓	SR ↑	Turns ↓	SR ↑	Turns ↓	SR ↑	Turns ↓	Score ↑	SR ↑	Turns ↓
Qwen3-30B-A3B teacher												
Zero-shot	40.67	17.00%	8.06	45.00%	23.13	36.57%	24.27	40.88%	23.69	51.86	23.01%	21.23
Qwen3-1.7B student												
Zero-shot	3.33	2.00%	14.68	2.86%	29.28	0.00%	30.00	1.46%	29.63	10.92	1.76%	17.01
Vanilla OPD	32.03	10.00%	8.40	23.57%	25.47	29.10%	25.01	26.28%	25.25	27.23	7.65%	25.58
AdaSwitch	34.43	12.00%	8.05	27.14%	25.58	26.12%	25.54	26.64%	25.56	35.85	11.01%	23.96
TCOD	36.41	11.00%	7.68	25.71%	25.42	32.09%	24.99	28.83%	25.21	30.45	5.93%	**22.80**
Guided-OPD	33.19	10.00%	8.34	26.43%	25.74	30.60%	24.99	28.47%	25.37	32.14	10.24%	25.23
TurnOPD	37.29	12.00%	7.96	27.14%	25.41	31.34%	25.01	29.20%	25.22	32.52	9.33%	24.91
DASH-OPD	**43.53**	**13.00%**	**6.25**	**31.43%**	**24.98**	**34.33%**	**24.69**	**32.85%**	**24.84**	**38.46**	**12.77%**	23.60
Qwen3-4B student												
Zero-shot	22.74	9.00%	11.41	31.43%	24.49	36.57%	23.83	33.94%	24.16	38.39	16.21%	19.08
Vanilla OPD	35.10	12.00%	8.44	34.29%	24.38	38.81%	24.40	36.50%	24.39	43.71	16.36%	23.19
AdaSwitch	34.85	14.00%	8.36	35.00%	23.75	38.06%	23.92	36.50%	23.83	46.19	20.34%	21.36
TCOD	38.05	14.50%	8.02	37.86%	23.64	34.70%	24.20	36.31%	23.91	47.30	19.69%	21.98
Guided-OPD	35.82	13.00%	8.42	35.00%	23.79	32.09%	24.72	33.58%	24.24	46.05	19.65%	21.12
TurnOPD	36.26	14.00%	8.24	35.71%	23.88	39.55%	23.88	37.59%	23.88	45.26	17.20%	22.45
DASH-OPD	**42.77**	**16.00%**	**7.15**	**41.43%**	**22.86**	**41.04%**	**23.46**	**41.24%**	**23.15**	**50.97**	**22.40%**	**20.62**

Benchmarks and metrics. We evaluate all methods on three benchmarks for long-horizon agentic tasks with text-based interactions: WebShop (Yao et al., 2022a), which requires searching for and selecting products on a simulated shopping website to satisfy user requirements; ALFWorld (Shridhar et al., 2020), which involves moving between rooms and manipulating objects to complete household tasks; and ScienceWorld (Wang et al., 2022), which requires conducting experiments in a virtual laboratory to achieve specified scientific goals. Appendix A.12 provides the prompt templates used for each benchmark. Following the official implementation of TCOD (Wang et al., 2026), we use the 100-task WebShop evaluation set and the 1,308-task ScienceWorld evaluation set. The evaluation on ALFWorld covers the full official test set of 140 IID and 134 OOD tasks. Following Guided-OPD (Li et al., 2026), we report task performance using mean score and mean success rate (SR) for WebShop and ScienceWorld, and SR alone for ALFWorld. We also report the mean number of interaction turns on each benchmark to measure deployment efficiency.

Training and evaluation setup. Most training and evaluation settings follow the official implementation of TCOD (Wang et al., 2026). We train all methods using the Trinity-RFT framework (Pan et al., 2025) with the AdamW optimizer (Loshchilov et al., 2017). We evaluate the final checkpoint using one rollout per task. Following TCOD and Guided-OPD, we use random seed 42 for all stochastic processes. All experiments are conducted on eight NVIDIA A100 GPUs with 80 GB of memory each. Detailed training and evaluation settings are provided in Appendix A.13 (Table 5).

DASH-OPD settings. We set the intervention threshold ratio to $q_{\text{on}} = 0.1$ and the return threshold ratio to $q_{\text{off}} = q_{\text{on}}/3$. The minimum teacher segment length $\ell_{\min}$ is set to 2, and the stagnation threshold K is set to 3. In Eq. (7), we lower-bound the denominator by 10^{-3} and clip $\bar{d}_t^{m_t}$ to $[-10, 10]$. These hyperparameters are held fixed across all benchmarks and model scales.

5.2 Main Results

Task performance. Table 1 reports task performance across the three benchmarks and two model scales. DASH-OPD outperforms all five baselines in all 14 task-performance comparisons. On

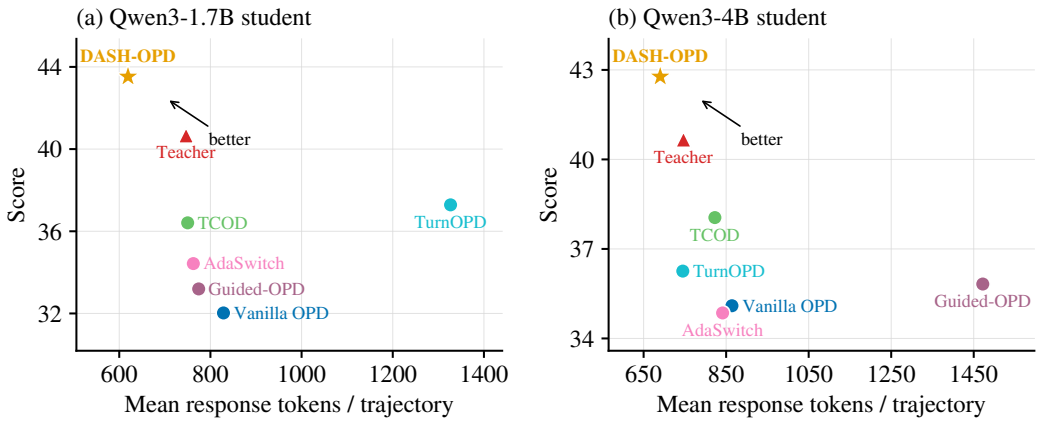

Figure 3: Performance–efficiency Pareto analysis on WebShop (mean score versus mean response tokens per task). Higher scores and fewer response tokens are better (upper left). At both model scales, DASH-OPD strictly Pareto-dominates all baselines and the zero-shot teacher. On ALFWorld and ScienceWorld, DASH-OPD is Pareto-optimal among baselines at 1.7B, and strictly Pareto-dominates all baselines (as well as the teacher on ALFWorld) at 4B (see Figs. 8 and 9 in Appendix A.6).

WebShop, the 1.7B DASH-OPD student achieves a score of 43.53 and an SR of 13.00%, exceeding the best baseline, TurnOPD, by 6.24 points and 1.00 percentage point, respectively. The 4B DASH-OPD student achieves a score of 42.77 and an SR of 16.00%, with corresponding gains of 4.72 points and 1.50 percentage points over the strongest baseline, TCOD. The 1.7B and 4B DASH-OPD students also exceed the zero-shot teacher's score by 2.86 and 2.10 points. On ALFWorld, DASH-OPD outperforms all baselines on both the IID and OOD splits at both model scales. The 1.7B and 4B DASH-OPD students achieve overall SRs of 32.85% and 41.24%, each exceeding the best baseline at the same scale by 3.65 percentage points. The 4B DASH-OPD student also exceeds the teacher's overall SR by 0.36 percentage points. On ScienceWorld, the 1.7B DASH-OPD student achieves a score of 38.46 and an SR of 12.77%, improving over the best baseline, AdaSwitch, by 2.61 points and 1.76 percentage points, respectively. The 4B DASH-OPD student achieves a score of 50.97 and an SR of 22.40%, with corresponding gains of 3.67 points and 2.06 percentage points over the strongest baseline for each metric.

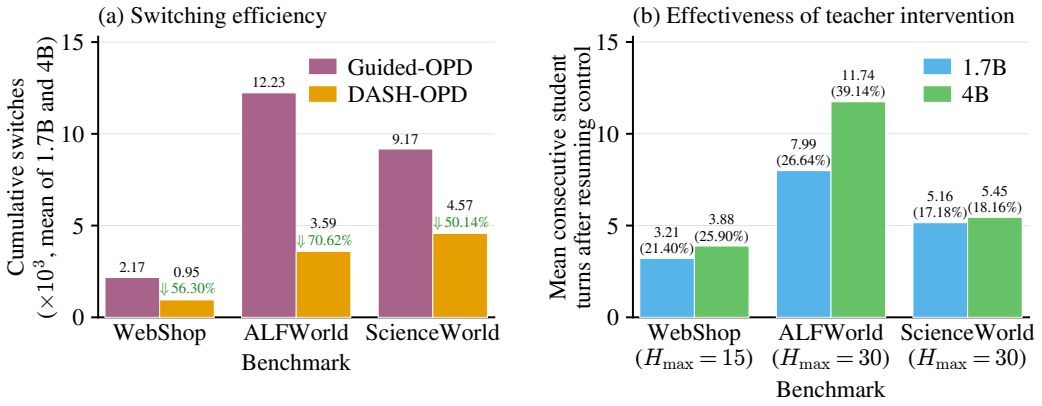

Figure 4: **(a)** Cumulative number of executor switches averaged over the 1.7B and 4B models. Fewer executor switches can reduce the overhead of control transfer. Across all benchmarks, DASH-OPD requires fewer switches than Guided-OPD. **(b)** Mean number of consecutive student turns after the teacher returns control during DASH-OPD training. Percentages indicate the fraction of the turn limit $H_{\max}$. Across all benchmarks and model scales, students sustain multi-turn execution after teacher intervention, indicating restored autonomy.

Table 2: Ablation results for DASH-OPD. Reporting conventions follow Table 1. "w/o teacher execution (OPD)": use student-only rollouts without teacher intervention (i.e., vanilla OPD). "w/o student resumption": prevent the student from resuming control after teacher intervention. "w/o signal accumulation": use single-turn standardized signals instead of accumulated evidence and scale both switching threshold ratios by 0.4. "w/o action mask": compute teacher–student discrepancy over all response tokens instead of action tokens only. "w/o stagnation trigger": disable teacher intervention triggered by stagnation in student actions or observations. "w/o minimum teacher span": remove the minimum-length constraint on teacher-controlled segments.

Method	WebShop			ALFWorld						ScienceWorld		
				IID		OOD		Overall				
	Score ↑	SR ↑	Turns ↓	SR ↑	Turns ↓	SR ↑	Turns ↓	SR ↑	Turns ↓	Score ↑	SR ↑	Turns ↓
Qwen3-1.7B student												
w/o teacher execution (OPD)	32.03	10.00%	8.40	23.57%	25.47	29.10%	25.01	26.28%	25.25	27.23	7.65%	25.58
w/o student resumption	32.26	**13.00%**	8.39	24.29%	25.84	31.34%	24.95	27.74%	25.40	32.14	9.63%	25.42
w/o signal accumulation	35.90	10.00%	7.47	27.86%	25.19	29.85%	25.18	28.83%	25.18	34.01	10.32%	24.46
w/o action mask	38.47	11.00%	7.16	27.14%	25.45	31.34%	24.63	29.20%	25.05	37.75	11.39%	24.14
w/o stagnation trigger	39.83	11.00%	6.71	**32.86%**	**24.74**	32.84%	**24.57**	**32.85%**	**24.65**	37.04	12.69%	**23.19**
w/o minimum teacher span	41.15	11.00%	6.52	26.43%	25.42	**34.33%**	24.81	30.29%	25.12	37.59	12.16%	23.92
DASH-OPD	**43.53**	**13.00%**	**6.25**	31.43%	24.98	**34.33%**	24.69	**32.85%**	24.84	**38.46**	**12.77%**	23.60
Qwen3-4B student												
w/o teacher execution (OPD)	35.10	12.00%	8.44	34.29%	24.38	38.81%	24.40	36.50%	24.39	43.71	16.36%	23.19
w/o student resumption	35.74	13.00%	8.35	37.86%	23.35	34.33%	24.54	36.13%	23.93	46.92	20.11%	21.66
w/o signal accumulation	34.13	11.00%	8.87	33.57%	24.62	36.57%	24.05	35.04%	24.34	45.37	19.42%	21.75
w/o action mask	37.85	15.00%	7.87	37.14%	24.37	**41.79%**	23.40	39.42%	23.89	49.06	21.71%	21.02
w/o stagnation trigger	40.39	**16.00%**	7.63	40.71%	23.74	37.31%	24.60	39.05%	24.16	46.14	**22.40%**	**20.61**
w/o minimum teacher span	39.58	**16.00%**	7.57	**41.43%**	23.42	38.06%	**23.27**	39.78%	23.35	47.70	21.10%	21.59
DASH-OPD	**42.77**	**16.00%**	**7.15**	**41.43%**	**22.86**	41.04%	23.46	**41.24%**	**23.15**	**50.97**	**22.40%**	20.62

Deployment efficiency. Table 1 also reports deployment efficiency measured by the mean number of interaction turns per task. Among trained students, DASH-OPD requires the fewest turns in nine of the ten efficiency comparisons: on WebShop and ALFWorld (IID, OOD, and overall) at both model scales, and on ScienceWorld at 4B. On ScienceWorld at 1.7B, DASH-OPD ranks second with 23.60 turns, behind TCOD (22.80).

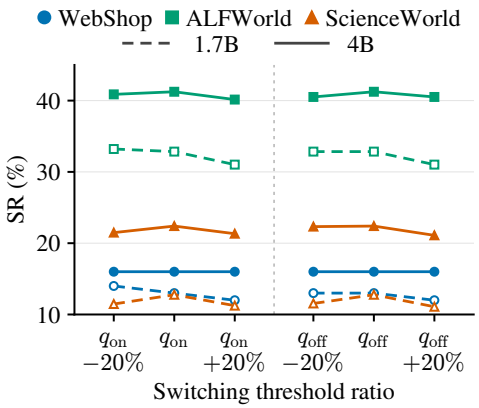

Figure 5: Sensitivity to switching threshold ratios. Each ratio is perturbed by ±20% relative to its default while the other is held fixed. ALFWorld uses overall SR. Varying either hyperparameter changes SR by at most 1.82 percentage points, suggesting local robustness.

Performance–efficiency Pareto analysis. Figure 3 compares mean score and mean response tokens per task on WebShop. At both model scales, DASH-OPD strictly Pareto-dominates all baselines and the zero-shot teacher, achieving higher scores with fewer response tokens. The 1.7B and 4B DASH-OPD students achieve scores of 43.53 and 42.77 with 619.24 and 690.24 response tokens, respectively. Compared with the teacher, they improve scores by 2.86 and 2.10 points while reducing response tokens by 127.37 and 56.37. The corresponding comparisons on ALFWorld and ScienceWorld are provided in Appendix A.6 (Figs. 8 and 9). On both benchmarks, the 1.7B DASH-OPD students are Pareto-optimal among baselines, while the 4B students strictly Pareto-dominate all baselines. On ALFWorld, the 4B student also Pareto-dominates the teacher. Appendix A.7 reports detailed token counts for all methods, including zero-shot students.

Number of executor switches. Figure 4(a) compares the cumulative number of executor switches for DASH-OPD and Guided-OPD (averaged over the 1.7B and 4B models). Fewer executor switches can reduce the overhead of transferring control between the student and teacher. DASH-OPD consistently requires fewer switches than Guided-OPD across all three benchmarks. This advantage

holds at both model scales. On WebShop, DASH-OPD reduces the total switches from 2,186 to 901 at 1.7B and from 2,155 to 996 at 4B, corresponding to reductions of 58.78% and 53.78%, respectively. On ALFWorld, DASH-OPD records 9,172 fewer switches (73.88%) at 1.7B and 8,105 fewer (67.25%) at 4B. On ScienceWorld, the corresponding reductions are 4,625 (49.02%) and 4,574 (51.32%).

DASH-OPD's lower switch count stems from its hysteretic switching mechanism, which suppresses frequent alternation between the teacher and student. Appendix A.8 provides further evidence of the hysteresis.

Effectiveness of teacher intervention. Figure 4(b) reports the mean number of consecutive student turns after the teacher returns control during DASH-OPD training. The turn limit per trajectory ($H_{\max}$) is 15 for WebShop and 30 for ALFWorld and ScienceWorld. On WebShop, the 1.7B and 4B students remain in control for an average of 3.21 and 3.88 turns, respectively, corresponding to 21.40% and 25.90% of the turn limit. The corresponding spans are 7.99 and 11.74 turns on ALFWorld (26.64% and 39.14% of its turn limit) and 5.16 and 5.45 turns on ScienceWorld (17.18% and 18.16%). Across all benchmarks, students at both scales execute multiple consecutive turns after regaining control, suggesting that teacher intervention helps restore sustained student autonomy. The 4B students maintain control longer than the 1.7B students, likely because of their greater capacity. Appendix A.9 complements this analysis with student survival rates after the teacher returns control.

5.3 Ablation study

Core components. We ablate three core components of DASH-OPD: teacher intervention during training rollouts, student resumption of control, and cross-turn signal accumulation. The results are reported in the "w/o teacher execution (OPD)", "w/o student resumption", and "w/o signal accumulation" rows in Table 2, respectively. In the "w/o signal accumulation" ablation, single-turn signals are typically smaller in magnitude than accumulated evidence, making the original switching thresholds harder to reach. We therefore scale both q_{on} and q_{off} by 0.4 (selected using the same tuning budget as for full DASH-OPD). Removing any of these core components substantially reduces task performance at both model scales: scores drop by 7.03–11.50 points on WebShop and by 4.05–11.23 points on ScienceWorld, while overall SR drops by 4.02–6.57 percentage points on ALFWorld. Appendix A.4 shows decreasing trends in teacher–student discrepancy and distillation losses over the course of training, supporting the effectiveness of DASH-OPD training.

Additional components. We also ablate three additional components of DASH-OPD: action masking in teacher–student discrepancy computation, stagnation-triggered teacher intervention, and the minimum-length constraint on teacher-controlled segments. The results are reported in the "w/o action mask", "w/o stagnation trigger", and "w/o minimum teacher span" rows in Table 2, respectively. Removing these components generally incurs smaller performance losses than the core ablations: scores drop by 2.38–5.06 points on WebShop and by 0.71–4.83 points on ScienceWorld, while overall SR drops by 0.00–3.65 percentage points on ALFWorld.

Hyperparameter sensitivity. We assess sensitivity to the switching threshold ratios by perturbing q_{on} and q_{off} individually by $\pm 20\%$ relative to their default values, while holding the other ratio fixed. As shown in Fig. 5, across all benchmarks and model scales, SR differs from the default setting by at most 1.82 percentage points. On WebShop, the 4B model maintains 16.00% SR under every perturbation. These results suggest robustness to local changes in both hyperparameters.

6 Conclusion

We propose DASH-OPD, the first agentic OPD method to perform adaptive, bidirectional executor switching. It uses directional teacher–student discrepancy to build drift and recovery evidence, then compares the evidence with thresholds to guide switching between the teacher and student. This hysteretic switching mechanism prevents rapid alternation between the two models. Across three benchmarks and two model scales, DASH-OPD outperforms five baselines in all task-performance comparisons and requires the fewest interaction turns in nine of ten efficiency comparisons.

STATEMENTS

AI USE STATEMENT

In this work, we used generative AI tools for generating synthetic data sets, implementing methods, assisting with translation, and interpreting results. We have not used generative AI tools for helping develop theoretical models or conceptual frameworks, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, cleaning and reformatting dataset, or supporting qualitative and thematic data analysis. Formulating mathematical claims, providing critical ingredients for proving mathematical claims, and assisting in the writing of proofs are not applicable to this work. Additionally, we used generative AI tools for creating or modifying scientific figures or images, drafting parts of a research paper, summarizing or analysing existing literature, sourcing/searching for information, editing a research paper to improve readability, and identifying relevant literature. We have reviewed all AI-assisted work. We checked LLM-generated code for potential mistakes through manual debugging. LLM-generated result analysis was verified for correctness. We reviewed AI-assisted draft text for accuracy and readability. We manually reviewed the references suggested by LLMs to verify their accuracy. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies on-policy distillation for language model agents using open-source Qwen3 models and the publicly available WebShop, ALFWorld, and ScienceWorld benchmarks. All agent interactions occur in simulated environments. Our experiments do not involve human participants, collect new personal data, make real purchases, or deploy agents on physical systems. As with other distillation methods, student models may inherit biases, factual errors, and harmful behaviors from their teachers, while downstream applications may pose risks of misuse or unintended consequences. These considerations are particularly relevant when extending agents beyond the controlled environments studied here. We therefore encourage application-specific safety and fairness evaluations before real-world deployment, along with appropriate safeguards and ongoing human oversight during use.

REPRODUCIBILITY STATEMENT

Our experiments use open-source Qwen3 models and the publicly available WebShop, ALFWorld, and ScienceWorld benchmarks. Section 4 describes DASH-OPD, with pseudocode in Appendix A.1. Section 5.1 specifies the benchmarks, model pairs, evaluation metrics, hardware, and DASH-OPD hyperparameters. Appendix A.13 details the shared training and evaluation settings, as well as baseline-specific configurations. Appendix A.12 provides the prompt templates used for training and evaluation. The code and trained models are available in this anonymous repository:

`https://anonymous.4open.science/r/DASH-OPD-7C3D`.

Training logs for DASH-OPD and all baselines will also be publicly released after the review process.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, 2009.

Haoyuan Cai, Zhenghao Peng, and Bolei Zhou. Robot-gated interactive imitation learning with adaptive intervention mechanism. *arXiv preprint arXiv:2506.09176*, 2025.

Yuqian Fu, Haohuan Huang, Kaiwen Jiang, Jiacai Liu, Zhuo Jiang, Yuanheng Zhu, and Dongbin Zhao. Revisiting on-policy distillation: Empirical failure modes and simple fixes. *arXiv preprint arXiv:2603.25562*, 2026.

Wenbo Gao, Renxi Liu, Xian Wang, Fang Guo, Shuai Yang, Xi Chen, Hui-Ling Zhen, Hanting Chen, Weizhe Lin, Xiaosong Li, et al. Agentcollab: A self-evaluation-driven collaboration paradigm for efficient llm agents. *arXiv preprint arXiv:2603.26034*, 2026.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations*, volume 2024, pp. 32694–32717, 2024.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Ryan Hoque, Ashwin Balakrishna, Ellen Novoseller, Albert Wilcox, Daniel S Brown, and Ken Goldberg. Thriftydagger: Budget-aware novelty and risk gating for interactive imitation learning. *arXiv preprint arXiv:2109.08273*, 2021.

Sam Ade Jacobs, Masahiro Tanaka, Chengming Zhang, Minjia Zhang, Shuaiwen Leon Song, Samyam Rajbhandari, and Yuxiong He. Deepspeed ulysses: System optimizations for enabling training of extreme long sequence transformer models. *arXiv preprint arXiv:2309.14509*, 2023.

Gengsheng Li, Mao Zheng, Mingyang Song, Ruiqi Liu, Tianyu Yang, Jie Sun, Qiyong Zhong, Haiyun Guo, Junfeng Fang, Dan Zhang, et al. On-policy distillation with curriculum turn-level guidance for multi-turn agents. *arXiv preprint arXiv:2606.15912*, 2026.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. Agentbench: Evaluating llms as agents. In *International Conference on Learning Representations*, volume 2024, pp. 52989–53046, 2024.

Ilya Loshchilov, Frank Hutter, et al. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*, 5(5):5, 2017.

Xuchen Pan, Yanxi Chen, Yushuo Chen, Yuchang Sun, Daoyuan Chen, Wenhao Zhang, Yuexiang Xie, Yilun Huang, Yilei Zhang, Dawei Gao, et al. Trinity-rft: A general-purpose and unified framework for reinforcement fine-tuning of large language models. *arXiv preprint arXiv:2505.17826*, 2025.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*, 2023.

Jingyu Peng, Maolin Wang, Hengyi Cai, Yuchen Li, Kai Zhang, Shuaiqiang Wang, Dawei Yin, and Xiangyu Zhao. Adaswitch: Balancing exploration and guidance in knowledge distillation via adaptive switching. *arXiv preprint arXiv:2510.07842*, 2025.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. 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*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

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

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*, 2020.

Mingyang Song and Mao Zheng. A survey of on-policy distillation for large language models. *arXiv preprint arXiv:2604.00626*, 2026.

Jiaqi Wang, Wenhao Zhang, Weijie Shi, Yaliang Li, and James Cheng. Tcod: Exploring temporal curriculum in on-policy distillation for multi-turn autonomous agents. *arXiv preprint arXiv:2604.24005*, 2026.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. Scienceworld: Is your agent smarter than a 5th grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11279–11298, 2022.

Taiqiang Wu, Chaofan Tao, Jiahao Wang, Runming Yang, Zhe Zhao, and Ngai Wong. Rethinking kullback-leibler divergence in knowledge distillation for large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 5737–5755, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

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

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022b.

Deheng Ye, Zhao Liu, Mingfei Sun, Bei Shi, Peilin Zhao, Hao Wu, Hongsheng Yu, Shaojie Yang, Xipeng Wu, Qingwei Guo, et al. Mastering complex control in moba games with deep reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 6672–6679, 2020.

Bowen Zhang. A formula-driven survey and research agenda for on-policy distillation. *arXiv preprint arXiv:2606.22793*, 2026.

Yaocheng Zhang, Jiajun Chai, Yuqian Fu, Songjun Tu, Xiaohan Wang, Wei Lin, Guojun Yin, Qichao Zhang, Yuanheng Zhu, and Dongbin Zhao. Are full rollouts necessary for on-policy distillation? *arXiv preprint arXiv:2605.31490*, 2026.

Qiyong Zhong, Mao Zheng, Mingyang Song, Xin Lin, Jie Sun, Houcheng Jiang, Xiang Wang, and Junfeng Fang. Sod: Step-wise on-policy distillation for small language model agents. *arXiv preprint arXiv:2605.07725*, 2026.

Yuhang Zhou, Lizhu Zhang, Yifan Wu, Mingyi Wang, Bo Peng, Jiayi Liu, Xiangjun Fan, and Zhuokai Zhao. Sage-opd: Selective agent-guided intervention for multi-turn on-policy distillation. *arXiv preprint arXiv:2606.19659*, 2026a.

Yuhang Zhou, Kai Zheng, Haoling Li, Dengyun Peng, Can Xu, and Jingjing Chen. Turnopd: Making on-policy distillation turn-aware for efficient long-horizon agent training. *arXiv preprint arXiv:2607.05804*, 2026b.

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

Zhou Ziheng, Jiaqi Li, Huacong Tang, Ying Nian Wu, and Demetri Terzopoulos. Less is more: Early stopping rollout for on-policy distillation. *arXiv preprint arXiv:2605.27028*, 2026.

A Appendix

A.1 DASH-OPD algorithmic details

Algorithm 1 presents the pseudocode for DASH-OPD. For improved training effectiveness, DASH-OPD does not follow TCOD and Guided-OPD in gradually reverting to vanilla OPD. Given the teacher–student discrepancy, DASH-OPD updates its statistics and switching state in $O(1)$ time per turn and uses $O(1)$ memory per trajectory.

Algorithm 1 DASH-OPD

Require: Student π_θ, rollout student $\pi_{\bar{\theta}}$, teacher π_T; environment $\mathcal{E}$ and initial history h_1; training step s and total steps $S_{\max}$; turn limit $H_{\max}$; $q_{\text{on}}, q_{\text{off}}, \ell_{\min}, K$

Set $\tau_{\text{on}}, \tau_{\text{off}}$ by Eq. (10); set $D = R = \ell = 0$ and initialize running statistics $\mathcal{R}^S, \mathcal{R}^T$
Sample $m_1 = T$ with probability $p_T(s)$ (Eq. (13)); otherwise set $m_1 = S$
for $t = 1, \ldots, H_{\max}$ **do**
  Generate response y_t under executor m_t given h_t (Eq. (4))
  Compute action-token discrepancy $d_t^{m_t}$ (Eqs. (5) and (6))
  Compute $\bar{d}_t^{m_t}$ using prior $\mathcal{R}^{m_t}$ (Eq. (7)); update $\mathcal{R}^{m_t}$ with $d_t^{m_t}$
  Execute the action in y_t in $\mathcal{E}$ and update h_{t+1}; **break** if the episode ends
  $m_{t+1} \leftarrow m_t$
  if $m_t = S$ **then**
    $D \leftarrow \max(D + \bar{d}_t^S, 0)$; compute stagnation flag g_t (Eq. (11))
    if $D > \tau_{\text{on}} \vee g_t$ **then** $m_{t+1} \leftarrow T$; $D, \ell \leftarrow 0$
  else
    $R \leftarrow \max(R - \bar{d}_t^T, 0)$; $\ell \leftarrow \ell + 1$
    if $R > \tau_{\text{off}} \wedge \ell \geq \ell_{\min}$ **then** $m_{t+1} \leftarrow S$; $R, \ell \leftarrow 0$
  end if
end for
Update π_θ on the collected responses: sampled reverse KL on student turns and sampled forward KL on teacher turns (Eq. (14))

A.2 Detailed training objectives for Guided-OPD and DASH-OPD

Following Guided-OPD, DASH-OPD uses the same role-specific token losses and clipping settings. Let $z_{t,i}^j = \log \pi_j(y_{t,i} \mid h_t, y_{t,<i})$ for $j \in \{\theta, \bar{\theta}, T\}$. On a student turn, we optimize a PPO-clipped objective (Schulman et al., 2017) using the advantage $a_{t,i} = z_{t,i}^T - z_{t,i}^{\bar{\theta}}$. Define the policy ratio $r_{t,i} = \exp\big(\text{clip}(z_{t,i}^\theta - z_{t,i}^{\bar{\theta}}, -20, 20)\big)$, its clipped value $\tilde{r}_{t,i} = \text{clip}(r_{t,i}, 1-\epsilon, 1+\epsilon)$, and $u_{t,i} = \max\{-a_{t,i} r_{t,i}, -a_{t,i} \tilde{r}_{t,i}\}$. The student-turn token loss in Eq. (14) can be expressed as the following dual-clipped PPO loss (Ye et al., 2020):

$$\mathcal{L}_{t,i}^S = \begin{cases} u_{t,i}, & a_{t,i} \geq 0, \\ \min\{u_{t,i}, -c\, a_{t,i}\}, & a_{t,i} < 0. \end{cases} \tag{15}$$

On a teacher turn, the teacher's log-probability is independent of θ. Minimizing the sampled forward KL is therefore equivalent to minimizing the following token-level negative log-likelihood (NLL):

$$\mathcal{L}_{t,i}^T = -z_{t,i}^\theta = -\log \pi_\theta(y_{t,i} \mid h_t, y_{t,<i}). \tag{16}$$

Both methods use $\epsilon = 0.2$, a dual-clip coefficient $c = 3$, and a KL coefficient of 1. Sequence masking and policy-gradient fallback are disabled (see Table 6). The inner clip in $r_{t,i}$ limits the log ratio for numerical stability. Equation (14) averages these losses over valid response tokens.

A.3 Rationale for the directionality of teacher–student discrepancy

In Eqs. (5) and (6), we place the executing policy in the numerator of the log-likelihood ratio. Consequently, the student-to-teacher ratio d_t^S is a sample-based analogue of $D_{\text{KL}}(\pi_{\bar{\theta}} \| \pi_T)$, whereas the teacher-to-student ratio d_t^T is a sample-based analogue of $D_{\text{KL}}(\pi_T \| \pi_{\bar{\theta}})$. Both ratios therefore share a common interpretation: larger values indicate greater discrepancy with the non-executing model. Their directions match the sampled reverse- and forward-KL objectives on student- and teacher-generated responses, respectively, and both ratios are directly computed from the log-probabilities collected during rollout scoring without further model forward passes.

A.4 Training dynamics

As shown in Fig. 6, the aggregate directional teacher–student discrepancy decreases over training at both model scales. From the first to the last 5% training-progress interval, the benchmark-mean

student-to-teacher ratio falls from 0.72 to 0.41 (42.65%) for 1.7B and from 0.39 to 0.26 (32.80%) for 4B. The teacher-to-student ratio falls from 0.99 to 0.25 (74.20%) at 1.7B and from 0.70 to 0.17 (75.60%) at 4B. These reductions provide evidence for the distillation process.

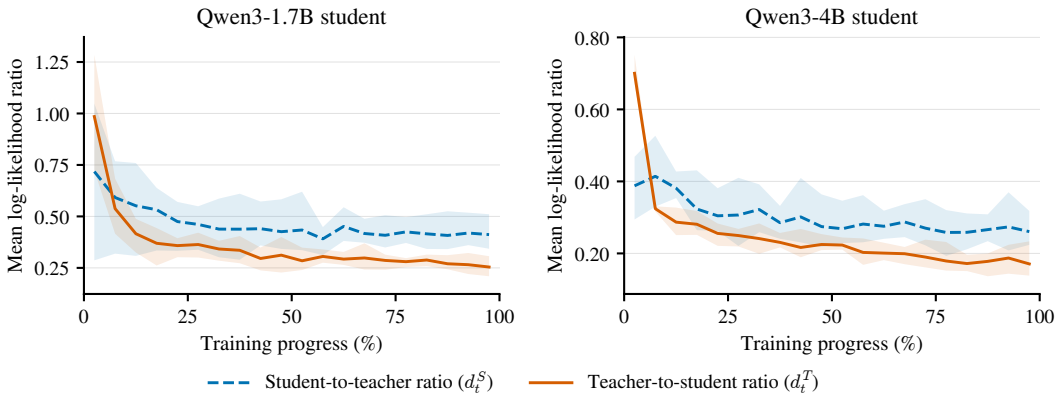

Figure 6: Directional teacher–student discrepancy during DASH-OPD training. These discrepancies are measured by the student-to-teacher and teacher-to-student mean log-likelihood ratios in Eqs. (5) and (6). The results are aggregated across all three benchmarks. "Training progress" is measured relative to the planned 150 training steps on WebShop and 250 steps on ALFWorld and ScienceWorld, then divided into 20 equal intervals (5% each). For each interval and model scale, we take the median batch estimate within each benchmark. The line shows the mean across benchmarks, and the shading spans the minimum and maximum. The decreasing discrepancies in both directions suggest the effectiveness of the distillation.

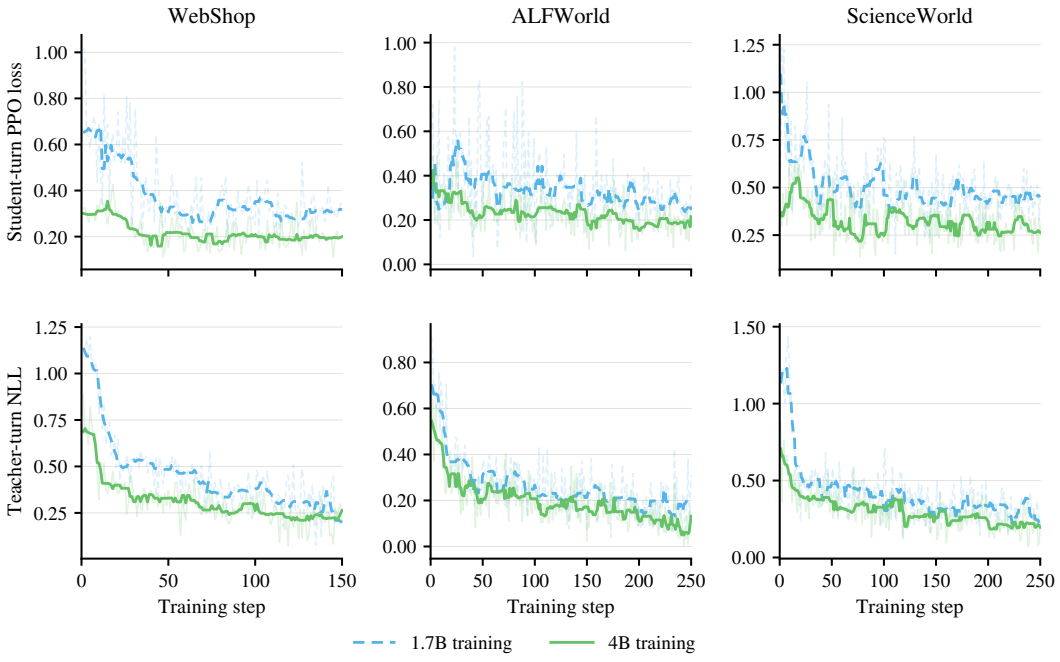

Figure 7: Distillation losses during DASH-OPD training. The top and bottom rows show the logged, token-averaged values corresponding to the student-turn PPO loss in Eq. (15) and the teacher-turn negative log-likelihood (NLL) in Eq. (16), respectively. Thin lines show logged step values, and thick lines show centered nine-step running medians. The decreasing losses indicate the effectiveness of the training process.

Figure 7 traces the two distillation losses detailed in Section A.2. Their decreasing trends suggest that the model is being effectively optimized throughout training.

A.5 Comparison with TCOD variants

Table 3 reports TCOD-B2F and TCOD-F2B separately. DASH-OPD exceeds both variants in all 14 task-performance comparisons across the three benchmarks and two model scales. Against the stronger variant on each metric, DASH-OPD's WebShop score gains are 6.03 and 3.99 points at 1.7B and 4B, and its ScienceWorld score gains are 7.48 and 3.29 points. The corresponding gains in ALFWorld overall SR are 2.19 and 4.01 percentage points. In addition, DASH-OPD uses the fewest turns in nine of ten efficiency comparisons (second fewest on ScienceWorld at 1.7B).

Table 3: Task performance and mean interaction turns for TCOD-B2F, TCOD-F2B, and DASH-OPD across three benchmarks and two student scales. B2F and F2B mean backward-to-forward and forward-to-backward. ALFWorld "Overall" combines the IID and OOD test splits. Higher scores and SRs are better; fewer turns are better. Within each model scale, **blue bold** and underline mark the best and second-best results, respectively (ties share the same style). DASH-OPD leads all 14 task-performance comparisons and requires the fewest turns in 9 of 10 efficiency comparisons (ranking second in the remaining one).

Method	WebShop			ALFWorld						ScienceWorld		
				IID		OOD		Overall				
	Score ↑	SR ↑	Turns ↓	SR ↑	Turns ↓	SR ↑	Turns ↓	SR ↑	Turns ↓	Score ↑	SR ↑	Turns ↓
Qwen3-1.7B student												
TCOD-B2F	35.33	11.00%	7.80	22.86%	25.64	31.34%	24.99	27.01%	25.32	29.91	3.44%	**20.15**
TCOD-F2B	37.50	11.00%	7.56	28.57%	25.20	32.84%	24.98	30.66%	25.09	30.98	8.41%	25.45
DASH-OPD	**43.53**	**13.00%**	**6.25**	**31.43%**	**24.98**	**34.33%**	**24.69**	**32.85%**	**24.84**	**38.46**	**12.77%**	23.60
Qwen3-4B student												
TCOD-B2F	37.32	14.00%	8.34	37.86%	23.59	32.84%	24.29	35.40%	23.93	47.68	19.27%	22.13
TCOD-F2B	38.78	15.00%	7.70	37.86%	23.70	36.57%	24.10	37.23%	23.90	46.92	20.11%	21.82
DASH-OPD	**42.77**	**16.00%**	**7.15**	**41.43%**	**22.86**	**41.04%**	**23.46**	**41.24%**	**23.15**	**50.97**	**22.40%**	**20.62**

A.6 Performance–efficiency Pareto analysis on ALFWorld and ScienceWorld

Figure 8 compares overall SR and mean response tokens per task on the full 274-task ALFWorld test set. The 1.7B DASH-OPD student is Pareto-optimal among baselines, achieving 32.85% overall SR with 1,433.13 response tokens per task. Compared with TurnOPD, the strongest baseline by SR, it gains 3.65 percentage points and uses 29.37 fewer tokens. The 4B student strictly Pareto-dominates all baselines and the zero-shot teacher, achieving 41.24% SR with 1,218.21 tokens. Relative to TurnOPD, it gains 3.65 percentage points and uses 50.11 fewer tokens. Relative to the teacher, it gains 0.36 percentage points and uses 68.46 fewer tokens.

Figure 9 compares mean score and mean response tokens per task on the 1,308-task ScienceWorld test set. The 1.7B DASH-OPD student is Pareto-optimal among baselines, scoring 38.46 with 1,981.40 tokens per task. Relative to AdaSwitch, the strongest baseline by score, it gains 2.61 points at a cost of 245.89 more tokens. The 4B student strictly Pareto-dominates all baselines, scoring 50.97 with 1,406.16 tokens. Relative to TCOD, the strongest baseline by score, it gains 3.67 points while using 127.72 fewer tokens.

A.7 Detailed response-token results

Table 4 gives the token counts underlying Figs. 3, 8 and 9 and includes the zero-shot students in Table 1 for reference. For each task, we sum response tokens across interaction turns, then average the task totals over the test set.

Among trained students, DASH-OPD uses the fewest tokens in four of six response-token comparisons (second fewest on ALFWorld at 1.7B): WebShop at both scales, plus ALFWorld and ScienceWorld at 4B. At 1.7B, it uses 25.78 more tokens than Guided-OPD on ALFWorld and 655.99 more than

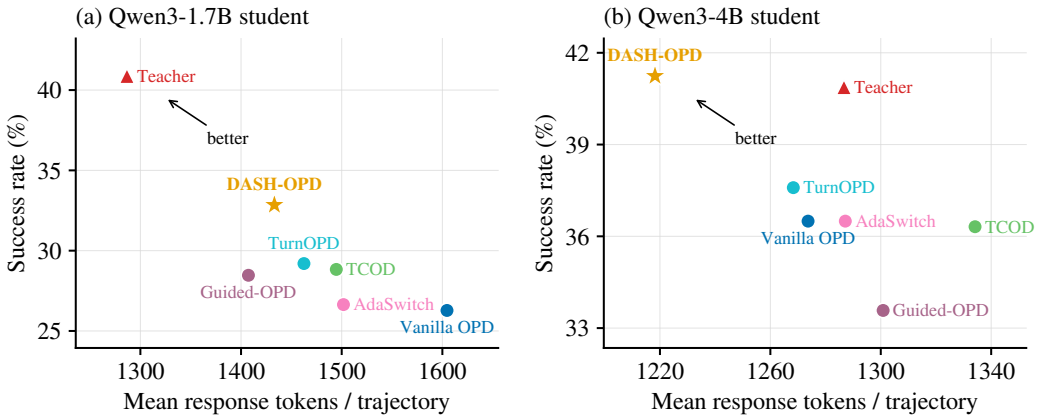

Figure 8: Performance–efficiency Pareto analysis on ALFWorld (overall SR versus mean response tokens per task). Higher SR and fewer response tokens are better (upper left). The 1.7B DASH-OPD student is Pareto-optimal among baselines, and the 4B student strictly Pareto-dominates all baselines and the zero-shot teacher.

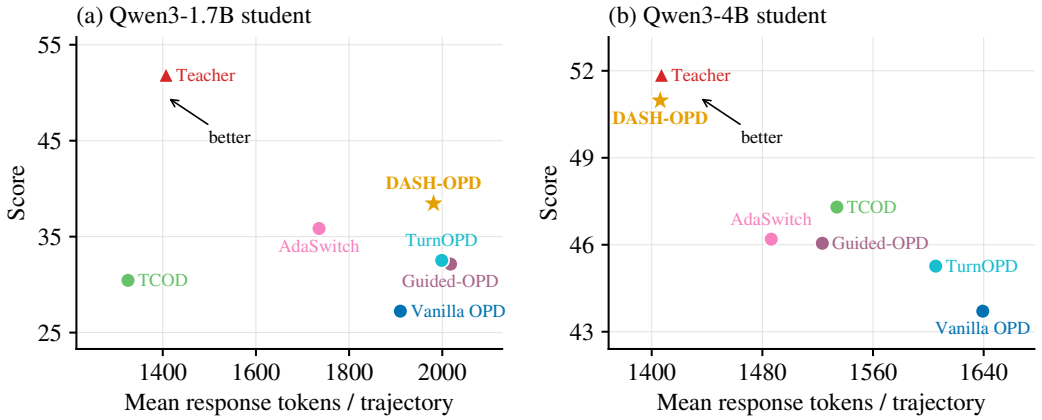

Figure 9: Performance–efficiency Pareto analysis on ScienceWorld (mean score versus mean response tokens per task). Higher scores and fewer response tokens are better (upper left). The 1.7B DASH-OPD student is Pareto-optimal among baselines, and the 4B student strictly Pareto-dominates all baselines.

TCOD on ScienceWorld, while gaining 4.38 percentage points in overall SR and 8.01 points in mean score, respectively.

A.8 Hysteretic switching

We sample rollouts every 50 training steps from the six DASH-OPD training runs in Table 1 and count the consecutive student or teacher turns before each executor switch, as shown in Fig. 10. The sample contains 386 student-to-teacher and 436 teacher-to-student switches. Before teacher intervention, the student remains in control for an average of 4.43 turns at 1.7B and 5.10 turns at 4B. Three consecutive student turns precede 41.75% and 39.58% of student-to-teacher switches, respectively. Before returning control to the student, the teacher remains in control for an average of 4.09 turns at 1.7B and 3.61 turns at 4B. Two consecutive teacher turns precede 45.45% and 53.24% of teacher-to-student switches, respectively. At 4B, the student remains in control longer and the teacher remains in control for fewer turns than at 1.7B. Most switches occur after multiple consecutive turns under one executor, providing further evidence of hysteretic switching.

Table 4: Mean response tokens per task across three benchmarks and two student scales. ALFWorld includes all 274 IID and OOD test tasks. TCOD entries average its B2F and F2B variants. Fewer tokens are better. Zero-shot teacher and student results are shown for reference and excluded from rankings. Within each model scale, **blue bold** and underline mark the best and second-best results, respectively (ties share the same style).

Method	WebShop Tokens ↓	ALFWorld Tokens ↓	ScienceWorld Tokens ↓
Qwen3-30B-A3B teacher			
Zero-shot	746.61	1,286.67	1,407.07
Qwen3-1.7B student			
Zero-shot	456.36	510.76	186.65
Vanilla OPD	828.66	1,604.59	1,910.01
AdaSwitch	762.52	1,501.75	1,735.51
TCOD	750.11	1,494.63	**1,325.41**
Guided-OPD	774.27	**1,407.35**	2,017.77
TurnOPD	1,326.80	1,462.50	1,998.78
DASH-OPD	**619.24**	1,433.13	1,981.40
Qwen3-4B student			
Zero-shot	1,113.30	1,301.18	1,428.46
Vanilla OPD	864.06	1,273.66	1,639.35
AdaSwitch	841.39	1,287.15	1,486.43
TCOD	822.71	1,334.16	1,533.88
Guided-OPD	1,471.92	1,300.85	1,523.34
TurnOPD	744.95	1,268.32	1,605.32
DASH-OPD	**690.24**	**1,218.21**	**1,406.16**

A.9 Student survival after the teacher returns control

Figure 11 complements the mean post-return student spans in Fig. 4(b) with survival rates at two horizons. At a three-turn horizon, the 1.7B and 4B rates are 84% and 91% on ALFWorld, 83% and 87% on ScienceWorld, and 43% and 53% on WebShop. At a five-turn horizon, they are 54% and 73%, 47% and 58%, and 31% and 41%, respectively. WebShop's 15-turn episode limit, versus 30 turns on the other benchmarks, can shorten its mean post-return student spans. At both horizons and on every benchmark, survival is higher for the 4B student, consistent with its longer mean post-return spans.

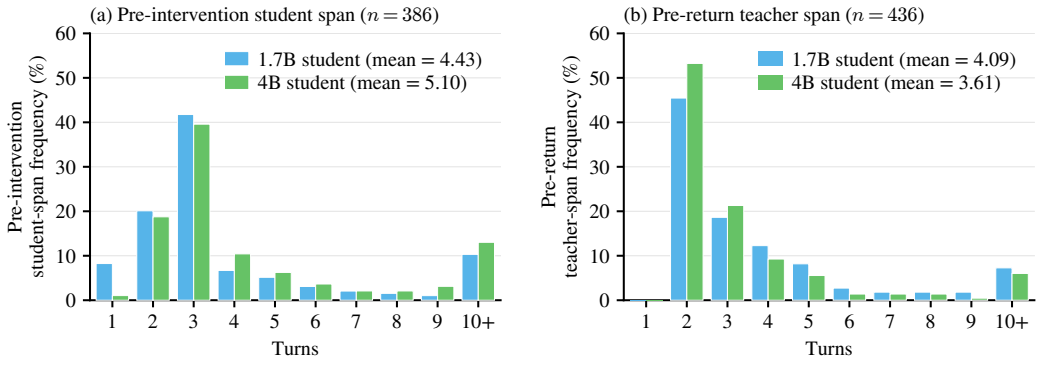

Figure 10: Number of consecutive student or teacher turns before an executor switch. Rollouts are sampled every 50 training steps from the six DASH-OPD training runs in Table 1. **(a)** Student turns before teacher intervention. **(b)** Teacher turns before control returns to the student. The most common spans are three student turns before teacher intervention and two teacher turns before control returns to the student, consistent with DASH-OPD's hysteretic switching mechanism.

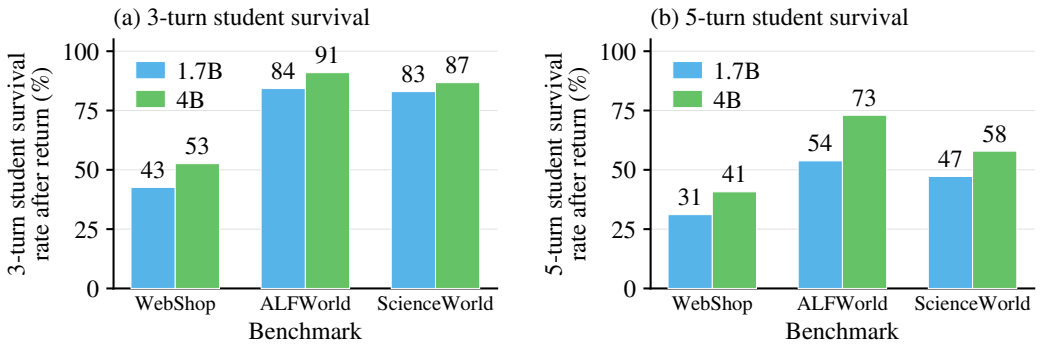

Figure 11: Student survival rates after the teacher returns control during DASH-OPD training at **(a)** three-turn and **(b)** five-turn horizons. Survival means that the student executes the full horizon without another teacher intervention or completes the task successfully before the horizon. The substantial post-return survival rates suggest that teacher intervention helps students continue independently.

Overall, the high post-return survival rates, especially at three turns, further support the effectiveness of teacher intervention in sustaining student autonomy.

A.10 Robustness of action-only discrepancy

If the action extraction for Eqs. (5) and (6) fails, the rollout is discarded without updating the controller, and training continues with other samples. In ALFWorld, such failures affect only 0.60% and 0.15% of trajectories for the 1.7B and 4B students, respectively. After excluding failed rollouts, the 1.7B run collects 3.92 times the number of samples required for training, and the 4B run collects 3.91 times the required number, leaving substantial sample surpluses in both runs. The runs on ScienceWorld and WebShop also collect more samples than required (3.18/2.87 times for 1.7B/4B on ScienceWorld and 1.20/1.34 times for 1.7B/4B on WebShop), leaving a surplus to accommodate rollout failures. Moreover, early in training, the high teacher-start probability in Eq. (13) reduces reliance on student-generated outputs, while full-response distillation helps the student learn the action format. Each DASH-OPD training run completes without termination due to action extraction failures.

A.11 Handling sudden increases in task difficulty

DASH-OPD supports teacher intervention when task difficulty suddenly increases after the student resumes control. Such increases can potentially be detected through a large teacher–student discrepancy or high student entropy. As described in Section 4, a sufficiently large teacher–student discrepancy on the first resumed student turn can immediately trigger teacher intervention on the next turn. We also evaluate an entropy-based fallback that discards the first resumed student response and retains teacher control if the response's mean token entropy exceeds 2. This fallback, however, is never triggered in our experiments.

A.12 Prompt templates

All methods use the same environment-specific user-prompt templates in training and evaluation. Given the same interaction history, the teacher and student receive identical prompts and use the same action parser. The full user–assistant conversation remains in context throughout the episode.

The first three turns have no explicit history summary. From the fourth turn onward, `{action_history}` contains the two most recent observation–action pairs, each formatted as `[Observation` i`: '`o_i`', Action` i`: '`a_i`']`.

The prompts ask for reasoning followed by an action. The environment action must appear between `<action>` and `</action>` for parsing. The boxes below reproduce the templates used in the experiments.

ALFWorld. In ALFWorld, the agent navigates and manipulates objects to complete household tasks. Each turn supplies the current observation and admissible actions. For the first three turns, the prompt has no separate task or history field: the task appears in the initial observation, which remains in the conversation history.

```
You are an expert agent operating in the ALFRED Embodied
Environment. 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
about the current situation. Once you've finished your
reasoning, you should choose the best admissible action for
the current step and present it within <action> </action>
tags. Do not output any other text besides your reasoning and
the action.
```

From the fourth turn onward, the prompt also includes the task description, the step count, and the two most recent observation–action pairs:

```
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
about the current situation. Once you've finished your
reasoning, you should choose the best admissible action for
the current step and present it within <action> </action>
tags. Do not output any other text besides your reasoning and
the action.
```

ScienceWorld. In ScienceWorld, the agent conducts experiments over multiple turns. The prompt lists action templates and available objects separately. The agent combines them into a valid action. The first three turns use the following template without an explicit history summary:

```
You are an expert agent operating in the ScienceWorld text
environment. {task_description} Your current observation is:
{current_observation} Available action commands:
[{action_templates}] Available objects you can interact with:
[{objects}]

Now it's your turn to take an action. Combine an action
command with appropriate object(s) to form a valid action. You
should first reason about the current situation. Once you've
finished your reasoning, you should choose the best valid
action for the current step and present it within <action>
</action> tags. Do not output any other text besides your
reasoning and the action.
```

From the fourth turn onward, the prompt also includes the step count and the two most recent observation–action pairs:

```
You are an expert agent operating in the ScienceWorld text
environment. {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}
Available action commands: [{action_templates}] Available
objects you can interact with: [{objects}]

Now it's your turn to take an action. Combine an action
command with appropriate object(s) to form a valid action. You
should first reason about the current situation. Once you've
finished your reasoning, you should choose the best valid
action for the current step and present it within <action>
</action> tags. Do not output any other text besides your
reasoning and the action.
```

WebShop. In WebShop, the agent navigates an e-commerce interface to satisfy a natural-language shopping request. Available actions depend on the current page. They primarily include `search[query]` when a search bar is present and `click[button_name]`, whose argument must match a clickable element. The first three turns use the following template:

```
You are an expert autonomous agent operating in the WebShop
e-commerce environment. Your task is to: {task_description}.
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 about the current situation, then
think which admissible action best advances the shopping goal.
Once you've finished your reasoning, you should choose the
best admissible action for the current step and present it
within <action> </action> tags. Do not output any other text
besides your reasoning and the action.
```

From the fourth turn onward, the prompt also includes the step count and the two most recent observation–action pairs:

```
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 about the current situation, then
think which admissible action best advances the shopping goal.
Once you've finished your reasoning, you should choose the
best admissible action for the current step and present it
within <action> </action> tags. Do not output any other text
besides your reasoning and the action.
```

A.13 DETAILED TRAINING AND EVALUATION SETTINGS

Settings shared by all methods. For each benchmark and student scale, all six methods use the same teacher–student pair, training framework, interaction protocol, and evaluation protocol. Table 5 lists these settings; most follow the official TCOD implementation.

Table 5: Training and evaluation settings shared by all six methods at both model scales. "No sampling truncation" denotes top-$p = 1$, top-$k = -1$, and min-$p = 0$.

Setting	WebShop	ALFWorld	ScienceWorld
Student model / frozen teacher	Qwen3-1.7B or Qwen3-4B / Qwen3-30B-A3B		
Training task pool size	4,096	3,553	2,294
Optimizer updates	150	250	250
Rollout / update batch size	16 / 64		
Rollout staleness	At most two student policy versions behind the current version		
Optimizer	AdamW; lr $= 10^{-6}$; $(\beta_1, \beta_2) = (0.9, 0.999)$; weight decay 0.01		
Learning-rate schedule / warm-up	Constant / none		
Gradient clipping	Global gradient norm capped at 1.0		
Additional RL components	No critic, reference-policy KL penalty, or entropy bonus		
Training decoding	Temperature 1.0; no sampling truncation; at most 512 new tokens per turn		
Maximum prompt length	10,240 tokens		
Maximum turns per task $H_{\max}$	15	30	30
Interaction context	Full conversation history; two latest observation–action pairs summarized from turn 4		
Model thinking mode	Qwen3 thinking mode disabled		
Random seed	42 for all stochastic components in training and evaluation		
Training precision	bfloat16		
Memory settings	Dynamic batching; 16,384 tokens per GPU; Ulysses sequence parallelism (Jacobs et al., 2023) $= 2$		
Hardware	One node with eight NVIDIA A100 GPUs, each with 80 GB of memory		
Evaluated policy	Final student checkpoint; no teacher or rollout controller		
Evaluation set	100 tasks	140 IID + 134 OOD tasks	1,308 tasks
Evaluation rollouts	One rollout per task		
Evaluation batch size	16 tasks		
Evaluation decoding	Temperature 0.4; no sampling truncation; at most 4,096 new tokens per turn		
Reported metrics	Mean score; SR; mean turns	SR and mean turns	Mean score; SR; mean turns

Baseline-specific settings and behavior. Table 6 specifies each baseline's hyperparameters, rollout rules, and training objective. We follow the original papers or official implementations where applicable. Settings that require adaptation to this experimental setup receive the same tuning budget as DASH-OPD.

A.14 LIMITATIONS AND FUTURE WORK

Our experiments cover three text-based benchmarks and one teacher–student model family. Evaluating DASH-OPD across other modalities and model families would clarify whether its performance gains generalize beyond the current experimental settings. In addition, DASH-OPD uses a single fixed teacher during training. Future work could explore multiple domain-specialized teachers and assess whether they improve performance across diverse tasks.

Table 6: Baseline-specific settings for both student model scales. Sampled reverse-KL losses use PPO clipping with $\epsilon = 0.2$ and a dual-clip coefficient of 3. TCOD and Guided-OPD advance their curricula once per rollout batch; TurnOPD numbers updates using the rollout student version plus one. NLL denotes negative log-likelihood, and EMA denotes exponential moving average.

Baseline	Setting	WebShop	ALFWorld	ScienceWorld
Vanilla OPD	Rollout executor	Student at every turn		
	Objective	Sampled reverse KL on student-generated tokens, scored by the teacher		
AdaSwitch	Token-level rollout rule	Begin each turn with the student; after a switch, the teacher completes that turn		
	Switching statistic	Truncated forward-KL proxy: renormalize each model's top-1,024 token probabilities, sum over tokens in both top-1,024 sets, and clamp negative values to zero		
	Window L / multiplier k	10 / 3	10 / 3	10 / 3
	Switching criterion	Switch when the current divergence exceeds k times the mean of the previous L values; require a full window before switching		
	Objective	Token-averaged weighted NLL: weights $\pi_T/\pi_{\bar{\theta}}$ on student-generated tokens and 1 on teacher-generated tokens; no PPO clipping		
TCOD	Reported result	Average of backward-to-forward (B2F) and forward-to-backward (F2B) results		
	B2F / F2B interval (batches)	4 / 4	5 / 6	5 / 6
	B2F curriculum	Replay an expert prefix, then let the student continue; shorten the prefix by one action at each B2F interval until no expert actions remain		
	F2B curriculum	Student horizon $\min\{H_{\max}, 1 + \lfloor s/\Delta_{\mathrm{F2B}} \rfloor\}$, where s is the zero-based rollout batch index and Δ_{F2B} is the F2B interval		
	Objective	Sampled reverse KL on student-generated tokens		
Guided-OPD	Executor selection	Choose the teacher or student independently at each turn (seed 42)		
	Teacher-selection decay	Cosine decay from 1 to 0		
	Decay duration (batches)	120	200	200
	Final student-only batches	30	50	50
	Executor-specific objective	Sampled reverse KL on student turns; NLL on teacher turns (sampled forward KL up to a constant)		
	Sequence masking / fallback	Sequence masking and policy-gradient fallback disabled		
TurnOPD	Depth: min. / requested max. / effective max.	2 / 50 / 15	2 / 50 / 30	2 / 50 / 30
	Probe schedule	Start the first probe during updates 1–3; start each later probe after 8 updates have elapsed since the preceding probe finished		
	Probe batch size	64 full-depth trajectories (4×16)		
	Controller signal	Reverse KL on the teacher's top-50 tokens, with both distributions renormalized		
	Coverage horizon	Nearest-rank 80th percentile of successful completion turns; retain the previous value with fewer than 8 successes (initial value 0)		
	EMA weight on new horizon	0.30	0.30	0.30
	Adaptive rollout depth	Initialize $\bar{H}$ to the effective maximum; use EMA to track the larger of the zero-based KL-mass and coverage horizons; clip $\lfloor \bar{H} + 0.5 \rfloor + 1$ to the depth bounds		
	KL-mass normalization ϵ	10^{-8}	10^{-8}	10^{-8}
	Turn-normalization blend	Linear $0 \to 1$; 150 updates	Linear $0 \to 1$; 250 updates	Linear $0 \to 1$; 250 updates
	Minimum support per turn	$\max\{8, \lceil 0.15N \rceil\}$ trajectories, where N is the number returned in the rollout batch; use uniform token weights if no turn meets this threshold		
	Objective	Sampled reverse KL, progressively shifting from uniform token weighting to equal weighting of valid turns		