Exploring Temporal Curriculum in On-Policy Distillation for Multi-turn Autonomous Agents

Jiaqi Wang[†], **Wenhao Zhang**, **Weijie Shi**, **Yaliang Li**, **James Cheng**[†]

Tongyi Lab, Alibaba Group

Abstract

On-policy distillation (OPD) has shown strong potential for transferring reasoning ability from frontier or domain-specific models to smaller students. While effective on static single-turn tasks, its behavior in multi-turn agent settings remains underexplored. In this work, we identify a key limitation of vanilla OPD in such settings, which we term *Trajectory-Level KL Instability*. Specifically, we observe that KL divergence increases together with a drop in success rate, and even after convergence, the KL remains high, leading to unstable training. This instability arises from inter-turn error compounding: as errors accumulate, the student is driven beyond the teacher's effective support, rendering the supervision signal unreliable. To address this, we propose **TCOD** (**T**emporal **C**urriculum **O**n-Policy **D**istillation), a simple yet effective framework that controls the trajectory depth exposed to the student and progressively expands it from short to long with a curriculum schedule. Experimental results across four student–teacher pairs on three multi-turn agent benchmarks (ALFWorld, WebShop, ScienceWorld) show that *TCOD mitigates KL escalation and enhances KL stability throughout training, improving agent performance by up to 18 points over vanilla OPD.* Further evaluations show that TCOD can even *surpass the teacher's performance and generalize to tasks on which the teacher fails.*

1 Introduction

On-policy distillation has recently emerged as a primary paradigm for transferring complex reasoning capabilities from frontier models to their smaller counterparts (Agarwal et al., 2024; Lu & Lab, 2025). These methods have demonstrated remarkable success in mathematical or question answering tasks by minimizing token-level KL divergence over student-generated rollouts (Jang et al., 2026; Ko et al., 2026; Jin et al., 2026). However, these approaches are inherently designed for static, single-turn reasoning. Consequently, they leave the more challenging *multi-turn* agent setting underexplored, where the model must continuously reason and act based on a growing history of sequential interactions. It remains a critical open question whether the stability of vanilla OPD can safely generalize to such dynamic, long-horizon environments.

In this work, we present empirical evidence that naively applying vanilla OPD in this multi-turn regime leads to a fundamental failure mode, which we term **Trajectory-Level KL Instability**. Through experiments on ALFWorld (Shridhar et al., 2020), we find that ***(i)*** the student models suffer from simultaneous KL escalation and success rate collapse, and ***(ii)*** although they eventually converge, they begin with prohibitively high KL divergence, both of which induce training instability. Crucially, as shown in Figure 1(left), we reveal the underlying mechanism: Compounding errors across turns progressively push the student into states outside the teacher's effective support. As a result, the teacher assigns lower probabilities to tokens in student-generated responses, indicating increasing KL divergence at each turn and rendering its supervision signal unreliable.

[†] The Chinese University of Hong Kong.

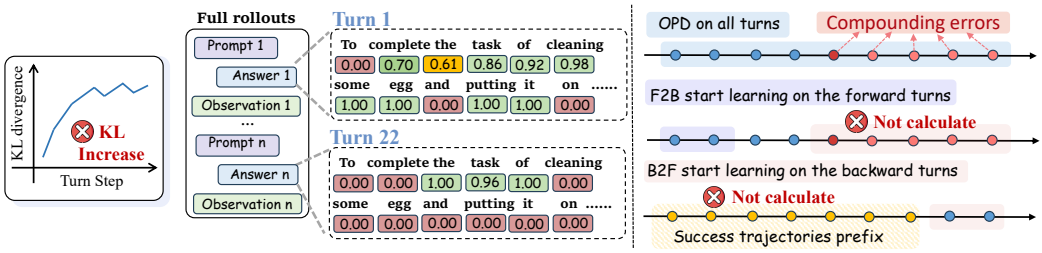


Figure 1: **(left)** In OPD for multi-turn agents, as the number of turns increases, the teacher assigns progressively lower probabilities to tokens in student-generated responses, indicating increasing KL divergence at each turn, rendering the supervision signal unreliable. **(right)** OPD uses all turns and thus includes compounding errors, whereas TCOD-F2B/B2F progressively expands from short to long trajectories, alleviating calculating the error turns.

To address this, we propose **TCOD** (**T**emporal **C**urriculum **O**n-Policy **D**istillation), a simple yet effective framework that controls the trajectory depth exposed to the student and progressively expands it from short to long via a pacing strategy governed by a configurable curriculum growth rate. Based on this core idea, as shown in Figure 1(right), we introduce two practical variants with only minimal code modifications: **Forward-to-Backward (TCOD-F2B)**, which restricts the student to the early steps of the trajectory and progressively extends it to the maximum explore horizon; and **Backward-to-Forward (TCOD-B2F)**, which leverages the teacher to navigate the agent to near-terminal states, alleviating error accumulation in the early steps, while gradually extending the student's rollout horizon backward to the initial stages.

Built on top of TCOD-F2B/B2F, we evaluate four student-teacher pairs on three multi-turn agent benchmarks: ALFWorld (Shridhar et al., 2020), WebShop (Yao et al., 2022a), and ScienceWorld (Wang et al., 2022). Overall, **TCOD alleviates KL instability and improves performance** by recovering Qwen3-1.7B from near-zero success rates and boosting the larger one (e.g., Qwen2.5-7B) by up to 15.71 success rate points while reducing action rounds by an average of 2.97 steps. Moreover, TCOD does not merely imitate the teacher—on the hard split of ALFWorld where the teacher fails under pass@10 sampling, TCOD-B2F surpasses the teacher's success rate by up to 14 points, **demonstrating generalization beyond the teacher's own capability boundary.** Finally, TCOD-F2B/B2F are robust to the curriculum growth rate with less than 2% performance variation, and reduce total training time by up to 32% compared to vanilla OPD.

2 Related Work

LLM-based Multi-turn Agents. Large language models have demonstrated strong capabilities as multi-turn agents (OpenAI et al., 2024; Yang et al., 2025). A common paradigm interleaves reasoning and action generation via frameworks such as ReAct (Yao et al., 2022b), enabling agents to solve tasks in embodied planning, web navigation, and other interactive environments (Shridhar et al., 2020; Yao et al., 2022a; Wang et al., 2022; Merrill et al., 2026; HKUDS, 2026; Li et al., 2026). Recent systems such as OpenClaw (Wang et al., 2026; Contributors, 2026) further demonstrate the potential of LLM-based agents for long-horizon tasks, motivating increasing interest in general-purpose agentic frameworks. Despite these advances, training multi-turn agents remains challenging due to long-horizon credit assignment (Guo et al., 2025), memory management (Shi et al., 2026), and the sample inefficiency of reinforcement learning in sparse-reward settings (Feng et al., 2025; Penaloza et al., 2026).

On-Policy Distillation and its Limitations. On-Policy Distillation (Agarwal et al., 2024; Lu & Lab, 2025) has emerged as a compelling alternative to on-policy post-training by replacing sparse scalar rewards with dense distillation signals and thereby improving sample efficiency. Existing work has improved OPD through several design choices, including objective design (Jang et al., 2026; Jin et al., 2026), optimization heuristics (Ko et al., 2026), and alternative supervision sources (Ye et al., 2026; Zhao et al., 2026). These methods, such as balancing forward and backward KL terms (Jang et al., 2026; Jin et al., 2026) and incorpo-

rating RL-style heuristics such as reward clipping (Ko et al., 2026), improve training stability and convergence. However, these approaches are primarily designed for single-turn settings and do not directly address multi-turn agent environments.

Curriculum Learning. Curriculum learning (Bengio et al., 2009) is a training strategy where a model is exposed to progressively more difficult examples as its competence grows. Recent works (Zhang et al., 2026; Wang et al., 2025b) apply this to the pre-training and post-training of LLMs, respectively. Shi et al. (2025); Wang & Ammanabrolu (2025); Gong et al. (2026) further apply curriculum learning to reinforcement learning methods, such as GRPO Guo et al. (2025), but still rely on an external model to measure difficulty. Lauffer et al. (2025) trains the student only on the expert's subsequent corrective actions, breaking the on-policy setting. Our approach avoids both by defining difficulty through increasing trajectory depth, using only student-generated data, keeping training simple, on-policy, and more stable.

3 Preliminary

In this paper, we consider multi-turn autonomous agents interacting with an environment over a finite horizon. Let $t \in \{0, \ldots, T-1\}$ denote the turn index within a trajectory, where T is the maximum number of interaction steps. At each turn t, the agent receives an observation o_t, generates a response a_t, and the environment returns the next observation o_{t+1}. Following the recent agent frameworks (Wang et al., 2025a), each response a_t consists of a chain-of-thought reasoning trace followed by an executable action.

History State for Multi-turn Agent. Since the environment is generally partially observable, we define the agent state as the full interaction history up to the current observation:

$$h_t = (o_0, a_0, o_1, a_1, \ldots, o_{t-1}, a_{t-1}, o_t). \tag{1}$$

A complete trajectory is then $\tau = (h_0, a_0, h_1, a_1, \ldots, h_{T-1}, a_{T-1})$, which terminates either when a termination action is taken or when the horizon T is reached.

On-Policy Distillation for Multi-turn Agent. Given a teacher policy π_ϕ and a student policy π_θ, the goal of on-policy distillation is to align the student with the teacher under the student's own state distribution. The objective is:

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

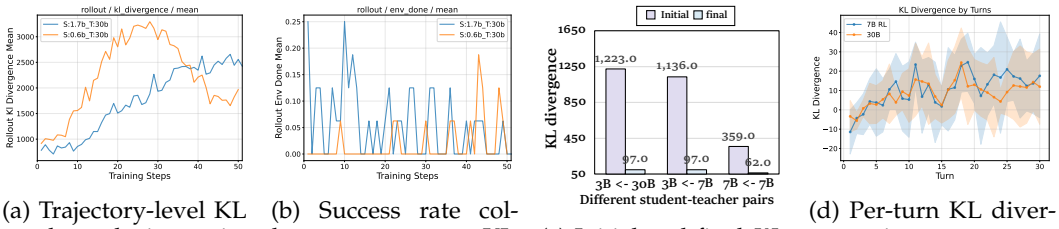
where $\mathcal{D}_{\text{KL}}(\pi_\phi \parallel \pi_\theta) = \sum_{a_t} \pi_\phi(a_t \mid h_t) \log \frac{\pi_\phi(a_t|h_t)}{\pi_\theta(a_t|h_t)}$ is the KL divergence measuring the discrepancy between the teacher policy π_ϕ and the student policy π_θ.

4 TCOD: Temporal Curriculum On-Policy Distillation

In this section, we observe a key limitation of OPD in multi-turn agent settings, termed trajectory-level KL instability. Through empirical analysis, we show that OPD exhibits instability in long-horizon interactions, where compounding errors lead to escalating KL divergence and degraded performance. Motivated by these findings, we propose TCOD, a temporal curriculum strategy that progressively controls trajectory depth during training to improve stability and effectiveness in multi-turn distillation.

4.1 Trajectory-Level KL Instability in Multi-Turn On-Policy Distillation

In this section, we conduct a pilot study on ALFWorld to examine the behavior of OPD in multi-turn settings. We systematically evaluate student–teacher pairs across the Qwen3 and Qwen2.5 model families, including both larger-scale and domain-adapted teachers. For Qwen3, we use Qwen3-30B-A3B-Instruct as the teacher and Qwen3-{0.6, 1.7, 4}B as students. For Qwen2.5, we adopt a GRPO-trained Qwen2.5-7B model as the teacher and Qwen2.5-{0.5, 1.5, 3, 7}B as students.



(a) Trajectory-level KL escalates during training.

(b) Success rate collapses to zero as KL spikes.

(c) Initial and final *KL* during OPD training.

(d) Per-turn KL divergence increases as errors accumulate.

Figure 2: **Trajectory-level KL analysis across different teacher–student pairs on ALFWorld.** (a)(b) show that the KL divergence escalates throughout training and task completion rates collapse. (c) shows the large gap between the initial and converged KL divergence during OPD training. (d) reveals the underlying reason: the KL divergence grows with the turn index, indicating compounding error amplification over the trajectory.

Observation 1: KL escalation and success rate collapse co-occur during training. Unlike prior work on single-turn settings such as mathematics or question answering, where the KL divergence consistently converges and decreases throughout training, we observe that the KL divergence escalates with the number of training steps in multi-turn agent scenarios. As shown in Figure 2a&2b, when the student model (Qwen3-{0.6,1.7}B) is trained under vanilla OPD with a strong teacher (Qwen3-30B-A3B-Instruct), the trajectory-level KL divergence escalates rapidly, and the task success rate collapses to near-zero.

Observation 2: Although KL divergence converges, it suffers from a prohibitively high initial value. Moreover, we conduct experiments on different student models and observe that although their KL divergence eventually converges, they start with a prohibitively high value. As shown in Figure 2c, across different student-teacher pairs (Qwen3-3B distilled from Qwen3-30B-A3B-Instruct, and Qwen2.5-{3,7}B distilled from a GRPO-trained Qwen2.5-7B model), we consistently observe the initial KL divergence (~ 1000) is typically orders of magnitude larger than its converged value (~ 60), indicating severe instability during the training for multi-turn OPD. More details refer to Appendix B.

The underlying mechanism: Compounding error amplification over the trajectory. The above two observations motivated us to investigate why directly applying OPD to agents leads to such KL escalation and training instability. To this end, in Figure 2d, we visualize the per-turn KL divergence for Qwen2.5-3B distilled from a GRPO-trained Qwen2.5-7B and Qwen3-30B-A3B-Instruct, and observe a consistent increase with the turn index.

Regardless of whether the increasing KL divergence reflects the student's inability to imitate the teacher or is a consequence of the student entering out-of-distribution states where the teacher becomes uncertain, the underlying issue remains the same—*error accumulation over turns*. This is an inherent property of long-horizon multi-turn agents: student-generated actions and observations are appended to the history h_t, inducing causal coupling across turns and resulting in an increasing trend in KL divergence. For small students, this is catastrophic; for larger ones, it is partially tolerated but remains highly inefficient.

Remark 1. *Note that Long-CoT increases the response length on the same environment state. However, multi-turn agents update the environment state at each interaction by incorporating new observations and actions, thereby amplifying compounding errors over the trajectory.*

The above observations and analysis pose a challenge: how can we retain the benefits of OPD's dense signal while avoiding destabilization from accumulated errors in long-horizon interactions? To address this, we turn to curriculum learning, where the model is first trained on easy problems and progressively exposed to hard ones.

4.2 Our Proposal: Temporal Curriculum On-Policy Distillation

Building on the observations and insights from the previous section, we propose **T**emporal **C**urriculum **O**n-Policy **D**istillation (TCOD), a principled approach that controls the trajectory depth of agent interactions during the training process. Specifically, we introduce two

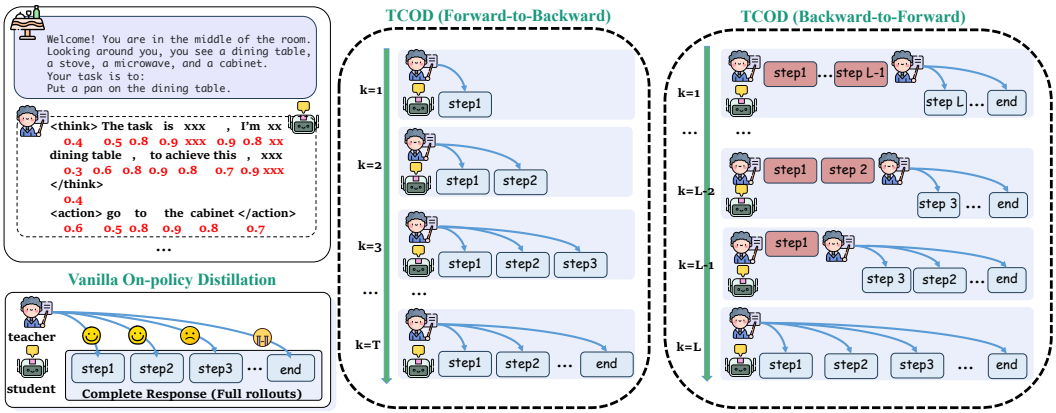


Figure 3: **Overview of our method TCOD-F2B/B2F.** Comparison of vanilla on-policy distillation and TCOD. **Left** is the OPD, **middle** is the illustration of TCOD-F2B, and **right** is TCOD-B2F. k is the linear pacing control the trajectory length. The blue step is executed by the student, and the red step is executed by the teacher with a stop gradient.

variants: TCOD-F2B and TCOD-B2F, which explicitly impose step constraints in forward and reverse curricula, respectively.

Forward-to-Backward Induced Temporal Curriculum On-Policy Distillation (TCOD-F2B). We implement a "shallow-to-deep" curriculum by restricting the maximum interaction steps of a trajectory during the training process. As shown in Figure 3(middle), in our TCOD-F2B, the student policy π_θ rolls out for maximum k steps to finish the task, where k starts from a small number and progressively increases to a larger one, the objective is as follows:

$$\mathcal{L}_{TCOD_F2B}(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{k-1} \mathcal{D}_{KL}\left(\pi_\phi(a_t|h_t) \parallel \pi_\theta(a_t|h_t)\right)\right], \tag{3}$$

where the student first focuses on early-turn learning signals and then progressively completes the task end-to-end, mitigating compounding errors and preventing horizon-induced KL collapse. However, determining the optimal step size and starting point is challenging, as different environments and models exhibit varying reasoning capabilities. To address this, we begin with linear pacing across the training step:

$$k = k_{\text{start}} + \lfloor n/\eta \rfloor, \quad n \in 1, \ldots, N, \tag{4}$$

where n represents the current training step and N is the total number of training steps, k_{start} defines the initial number of interaction steps and η controls the curriculum's growth rate. This approach requires only minimal code modifications:

Algorithm 1 Code Change for `TCOD (F2B)`

```
def tcod_f2b_env_step(max_env_step, n, k_start, eta):
    k = k_start + n/eta
    return min(max_env_step, k)
```

Furthermore, to better exploit the teacher model, we propose TCOD-B2F, which leverages the teacher to avoid early-turn error accumulation.

Backward-to-Forward Induced Temporal Curriculum On-Policy Distillation (TCOD-B2F). In this variant, the teacher policy π_ϕ acts as a "navigator." We initialize the environment to an intermediate state obtained by executing the initial prefix of a pre-collected successful trajectory τ^* using the teacher policy π_ϕ, and let the agent start interaction from this state. Specifically, as shown in Figure 3, the teacher executes the first $L-k$ steps of its successful trajectory τ^* in the environment, after which the student policy π_θ takes over from this immediate state to continue planning and execution. The objective is as follows:

$$\mathcal{L}_{\text{TCOD_B2F}}(\theta) = \mathbb{E}_{\tau\sim(\pi_\phi,\pi_\theta)}\left[\sum_{t=L-k+1}^{T-1} \mathcal{D}_{KL}\left(\pi_\phi(a_t|h_t) \parallel \pi_\theta(a_t|h_t)\right)\right], \tag{5}$$

where L denotes the length of the successful trajectory τ^* for a given task, and k is defined as in Equation 4, monotonically expanding until the student completes the task end-to-end throughout training. This implementation is similarly lightweight, requiring only a simple warmup loop as shown in Algorithm 2.

Algorithm 2 Code Change for `TCOD (B2F)`

```
def tcod_b2f_env(env, teacher_actions, n, k_start, eta):
    k = k_start + n/eta
    L = len(teacher_actions)
    if L-k < 0: return env
    for i in range(L-k):
        env.step(teacher_actions[i])
    return env
```

This mechanism effectively **bypasses compounding action errors** by ensuring the student only optimizes on trajectories initiated from successful, teacher-vetted prefixes. Crucially, the teacher steps of the trajectory do not contribute to the gradient, serving only to place the student on the "doorstep of success." Detailed algorithms are provided in Appendix C.

Discussion of the train-test mismatch in TCOD-B2F. During training, the student starts from a teacher-navigated checkpoint, whereas at test time it must act end-to-end from scratch. To this end, we gradually reduce the teacher's prefix from $L-1$ steps down to zero, ensuring that by the end of training the student executes the full trajectory from the initial state with no teacher intervention, fully aligning the training and test distributions. As shown in Appendix D.6, the end-to-end success rate on the test set increases steadily with training steps, confirming that the smooth curriculum transition effectively prevents catastrophic distribution shift in practice.

5 Experiments

In this section, we conduct experiments on various benchmarks to evaluate our approach. Mainly, we design the experiments to study the following key questions:

Q1: Compared to vanilla OPD, how does TCOD alleviate KL escalation and recover the performance for small student models, and how does it enhance training stability and the performance for the larger ones?

Q2: Can TCOD enable the student to generalize effectively to tasks beyond the teacher's own capability boundary?

Q3: How sensitive is TCOD to the growth rate of the curriculum, and how does it compare to vanilla OPD in terms of training efficiency?

5.1 Experimental Setup

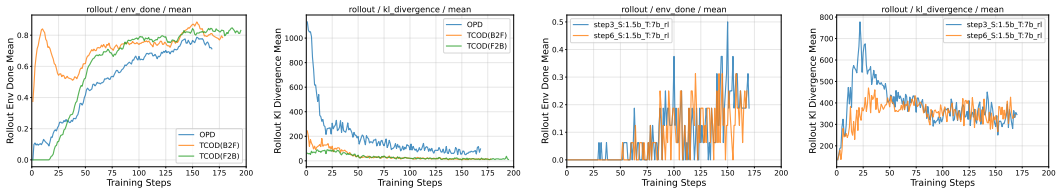
Benchmarks. We conduct experiments on three benchmarks including the embodied navigation environment ALFWorld (Shridhar et al., 2020), e-commerce platform WebShop (Yao et al., 2022a), and scientific reasoning ScienceWorld (Wang et al., 2022) as illustrated in Table 1, spanning a spectrum of reasoning levels from simple to complex. Max turns means the maximum exploration steps for each task. For ALFWorld, we evaluate on both the *seen* and *unseen* splits, where the unseen split contains novel room layouts and object combinations not encountered during training, serving as our OOD evaluation. We additionally construct a **Hard** set comprising tasks where the teacher fails under pass@10 sampling on the train split, to test whether TCOD can generalize beyond the teacher's own capability boundary. More benchmark details refer to Appendix D.1.

Table 1: **Summary of the benchmarks used**.

Benchmark	OOD	Type	Difficulty	Max Turns
ALFWorld-seen		Embodied	Easy	30
ALFWorld-unseen	✓	Embodied	Medium	30
ALFWorld-hard	✓	Embodied	Hard	30
Webshop		E-commerce	Medium	15
ScienceWorld		Scientific Reason	Hard	30

Table 2: **Out-of-domain and hard set performance comparison** between TCOD and OPD on ALFWorld. SR is success rate (%) and Rounds is averge action rounds per task. The best result is **bolded** and the second-best is underlined. Green and red subscripts indicate improvement and degradation over Vanilla OPD, respectively.

Method	Valid Seen		Valid Unseen		Hard	
	SR ↑	Rounds ↓	SR ↑	Rounds ↓	SR ↑	Rounds ↓
Qwen2.5-7B-RL (Teacher)	85.71	10.61	76.87	13.06	6.61	27.31
Qwen2.5-3B (Student)						
Zero-Shot	7.86	28.73	2.24	29.63	0.83	29.88
SFT	32.14	22.85	25.37	24.16	4.96	29.12
Vanilla OPD	65.72	14.73	60.45	16.21	10.74	28.64
TCOD (B2F)	77.86$_{\uparrow 12.14}$	12.57$_{\downarrow 2.16}$	70.90$_{\uparrow 10.45}$	14.56$_{\downarrow 1.65}$	**13.22**$_{\uparrow 2.48}$	**28.16**$_{\downarrow 0.48}$
TCOD (F2B)	**81.43**$_{\uparrow 15.71}$	**11.76**$_{\downarrow 2.97}$	**79.19**$_{\uparrow 18.74}$	**12.47**$_{\downarrow 3.74}$	9.92$_{\downarrow 0.82}$	28.57$_{\downarrow 0.07}$
Qwen2.5-7B (Student)						
Zero-Shot	9.29	28.34	8.96	28.46	1.65	29.77
SFT	54.29	18.92	48.73	20.11	8.26	28.73
Vanilla OPD	75.37	13.18	72.14	13.37	13.22	27.89
TCOD (B2F)	**86.43**$_{\uparrow 11.06}$	**11.06**$_{\downarrow 2.12}$	**77.61**$_{\uparrow 5.47}$	**13.16**$_{\downarrow 0.21}$	**20.66**$_{\uparrow 7.44}$	**27.07**$_{\downarrow 0.82}$
TCOD (F2B)	82.14$_{\uparrow 6.77}$	13.22$_{\uparrow 0.04}$	76.12$_{\uparrow 3.98}$	13.22$_{\downarrow 0.15}$	18.18$_{\uparrow 4.96}$	27.37$_{\downarrow 0.52}$



(a) Success Rates (b) KL Divergence (c) Success Rates (d) KL Divergence

Figure 4: **Training Dynamics comparison of TCOD and OPD on ALFWorld.** (a) and (b) show the success rate and KL divergence, respectively, for Qwen2.5-7B as the student. TCOD maintains a higher success rate and more stable KL divergence. (c) and (d) show the success rate and KL divergence, respectively, for Qwen2.5-1.5B as the student model. TCOD-F2B under $\eta = 3, 6$ mitigate the success rate collapse and kl escalation.

Training Details. For the main experiments on ALFWorld, we use Qwen2.5-3B and Qwen2.5-7B as student models, with Qwen2.5-7B fine-tuned via GRPO on the ALFWorld domain serving as the teacher. For the cross-benchmark evaluation, we adopt Qwen3-1.7B and Qwen3-4B as students, with Qwen3-30B-A3B-Instruct as the teacher. All experiments are conducted on 8× NVIDIA H20 GPUs. We implement TCOD based on the Reinforcement Fine-Tuning framework Trinity-RFT (Pan et al., 2025). For expert trajectory collection for TCOD-B2F initialization, we adopt a pass@10 sampling strategy using the teacher model, retaining only successful trajectories. For simplicity, we fix $k_{\text{start}} = 1$ and $\eta = 2$ and examine the impact of different η from $\{2, 4, 6\}$ in Sec 5.4.

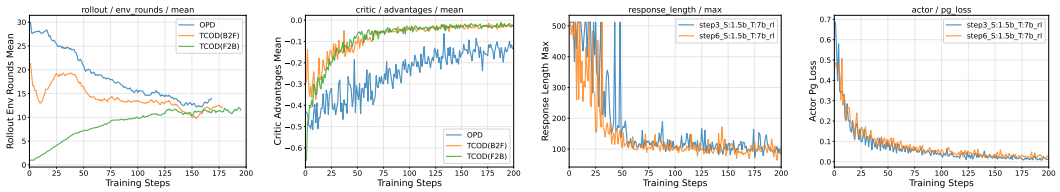
For baselines, we report the zero-shot student as the empirical lower bound and the teacher policy as the theoretical upper bound (Oracle). Moreover, we compare TCOD with standard knowledge transfer paradigms, including supervised fine-tuning (SFT) and vanilla on-policy distillation (OPD). For evaluation, we test all the benchmarks using **success rate (SR)**, which measures the percentage of tasks completed successfully, where task completion is treated as a binary outcome. More details are provided in Appendix D.

5.2 Q1: Alleviating KL Escalation and Improving Performance

In Table 2, we present results of TCOD on ALFWorld with students (Qwen2.5-3B, Qwen2.5-7B) and a GRPO-trained Qwen2.5-7B teacher, reporting both success rate (SR) and average action steps. We find that TCOD-F2B and B2F substantially outperform vanilla OPD and SFT across model scales. Notably, TCOD reduces the average number of action steps

Table 3: **Performance comparison between TCOD-F2B/B2F and OPD.** We report Success Rate (%) on validation sets. η is the curriculum's growth rate. The best result is **bolded** and the second-best is underlined. Green and red subscripts indicate improvement and degradation over Vanilla OPD, respectively.

Method	WebShop ↑	ALFWorld ↑	ScienceWorld ↑	Avg ↑
Qwen3-30B (Teacher)	32.84	39.57	18.42	30.28
Qwen3-1.7B (Student)				
Vanilla OPD	0.14	0.32	0.05	0.17
TCOD (B2F) η=2	20.54$_{\uparrow 20.40}$	24.55$_{\uparrow 24.23}$	10.82$_{\uparrow 10.77}$	18.64$_{\uparrow 18.47}$
TCOD (B2F) η=4	**21.12**$_{\uparrow 20.98}$	23.87$_{\uparrow 23.55}$	**11.34**$_{\uparrow 11.29}$	**18.78**$_{\uparrow 18.61}$
TCOD (B2F) η=6	20.33$_{\uparrow 20.19}$	**24.91**$_{\uparrow 24.59}$	10.65$_{\uparrow 10.60}$	18.63$_{\uparrow 18.46}$
TCOD (F2B) η=2	21.15$_{\uparrow 21.01}$	24.12$_{\uparrow 23.80}$	10.45$_{\uparrow 10.40}$	18.57$_{\uparrow 18.40}$
TCOD (F2B) η=4	20.44$_{\uparrow 20.30}$	**25.03**$_{\uparrow 24.71}$	9.22$_{\uparrow 9.17}$	18.23$_{\uparrow 18.06}$
TCOD (F2B) η=6	**21.78**$_{\uparrow 21.64}$	23.65$_{\uparrow 23.33}$	**11.08**$_{\uparrow 11.03}$	**18.84**$_{\uparrow 18.67}$
Qwen3-4B (Student)				
Vanilla OPD	30.12	36.85	15.95	27.64
TCOD (B2F) η=2	**32.15**$_{\uparrow 2.03}$	38.62$_{\uparrow 1.77}$	17.46$_{\uparrow 1.51}$	**29.41**$_{\uparrow 1.77}$
TCOD (B2F) η=4	29.21$_{\downarrow 0.91}$	37.84$_{\uparrow 0.99}$	**16.73**$_{\uparrow 0.78}$	27.93$_{\uparrow 0.29}$
TCOD (B2F) η=6	29.05$_{\downarrow 1.07}$	**39.35**$_{\uparrow 2.50}$	15.88$_{\downarrow 0.07}$	28.09$_{\uparrow 0.45}$
TCOD (F2B) η=2	**31.81**$_{\uparrow 1.69}$	**38.95**$_{\uparrow 2.10}$	**17.85**$_{\uparrow 1.90}$	**29.54**$_{\uparrow 1.90}$
TCOD (F2B) η=4	30.54$_{\uparrow 0.42}$	37.62$_{\uparrow 0.77}$	17.23$_{\uparrow 1.28}$	28.46$_{\uparrow 0.82}$
TCOD (F2B) η=6	29.87$_{\downarrow 0.25}$	37.88$_{\uparrow 1.03}$	17.12$_{\uparrow 1.17}$	28.29$_{\uparrow 0.65}$



(a) Action Rounds (b) Advantages (c) Max Response length (d) Policy Gradient Loss

Figure 5: **Further Analysis of TCOD-F2B/B2F on ALFWorld.** (a)(b) is the average action rounds, advantages during training for Qwen2.5-7B as the student. TCOD effectively reduces the action rounds and achieves faster advantage convergence. (c)(d) is the maximum response length, policy gradient loss during training for Qwen2.5-1.5B as the student model. TCOD mitigates redundant responses while maintaining training stability.

by 2.97 steps, while improving SR by up to 15.71 over OPD, suggesting that **curriculum learning from the teacher over trajectories leads to better performance**. Figure 4a&4b&5b further show that TCOD achieves faster convergence in success rate and advantage, while maintaining more stable KL divergence than vanilla OPD.

Different Benchmarks and Model Sizes. In Table 3, we evaluate TCOD across three benchmarks using students Qwen3-1.7B and Qwen3-4B with a Qwen3-30B-A3B-Instruct teacher. Overall, TCOD-F2B and TCOD-B2F achieve comparable performance to vanilla OPD. Moreover, as illustrated in Figure 4c&4d, TCOD-F2B under both $\eta = \{3, 6\}$ **maintains stable KL throughout training and achieves an increasing success rate, effectively mitigating KL escalation and improving average success rate by 18.67.** Furthermore, Figure 5c&5d shows additional training metrics, where TCOD can recover from an explosion in response length, while the policy gradient loss decreases smoothly.

5.3 Q2: Generalizing Beyond the Teacher's Capability Boundary

Beyond the performance gains and KL stability achieved by TCOD, we further investigate whether TCOD can enable the student to surpass the teacher itself. Table 2 reports performance on both the unseen environment split and the hard split. Specifically, the

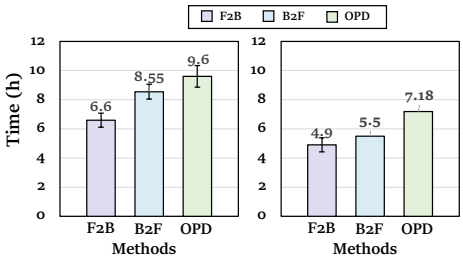
hard split comprises 121 challenging tasks from ALFWorld where the teacher performs poorly. On the unseen split, TCOD already outperforms the teacher by up to 2.5 points in SR. More surprisingly, on the Train Hard split, both TCOD-B2F and TCOD-F2B substantially exceed the teacher's SR of 6.61, with TCOD-B2F achieving a gain of up to 14 points. This demonstrates that **TCOD does not merely imitate the teacher, but develops a more robust policy that generalizes beyond the teacher's capability boundary.**

5.4 Q3: Robustness, Sensitivity, and Efficiency Analysis of TCOD

Curriculum's Growth Rate η Ablation. Table 3 reports the effect of varying the curriculum's growth rate $\eta \in \{2, 4, 6\}$ across different benchmarks. Performance remains consistently stronger than vanilla OPD across settings, with less than 2% variation in success rate, demonstrating that TCOD-F2B/B2F is not sensitive to the specific choice of η. This robustness makes TCOD easy to deploy in practice without extensive hyperparameter tuning. Nonetheless, as shown in Figure 4d, a larger η leads to more stable KL divergence during training, as the student spends more iterations mastering the current trajectory depth before the curriculum advances to longer horizons. In practice, we recommend starting with a small η to allow the curriculum to progress quickly in the early stages, and increasing η if the KL divergence instability during training is observed.

Domain-Specific *vs.* Larger Teacher. Comparing Table 2 and Table 3, we find that teacher quality strongly affects the upper bound of TCOD. In Table 2, the teacher is a GRPO-tuned Qwen2.5-7B on ALFWorld, reaching 85.71% success rate. Under this setting, TCOD-B2F with the same 7B backbone even slightly surpasses the teacher by 0.7 points. In Table 3, the teacher is Qwen3-30B-A3B-Instruct, a general model with weaker performance on the target domain. In this case, both vanilla OPD and TCOD fail to exceed the teacher, with about a 2-point gap. This suggests that the teacher's performance on the target domain matters more than model scale alone in enabling student improvement.

TCOD is computationally efficient. Figure 6 compares the total training cost of TCOD and vanilla OPD on ALFWorld and ScienceWorld. On both benchmarks, TCOD-F2B and TCOD-B2F reduce total training time by nearly 32% compared to vanilla OPD. This gain comes from the step-based curriculum in TCOD: early in training, the student takes fewer steps, producing shorter trajectories and faster data collection. Notably, TCOD-F2B is more efficient than TCOD-B2F. This is because TCOD-F2B limits the maximum interaction steps to k, while TCOD-B2F, though starting from intermediate states, still leads the student to take extra exploratory actions, producing longer trajectories. Figure 5a further verifies that TCOD-F2B uses fewer rollout action steps than TCOD-B2F, and both require fewer steps than vanilla OPD.



(a) The training time comparison based on Qwen3-4B on ScienceWorld.

(b) The training time comparison based on Qwen2.5-7B on ALFWorld.

Figure 6: **Training time comparison**.

6 Conclusion

In this work, we identify a fundamental failure mode of vanilla OPD in multi-turn agents, termed Trajectory-Level KL Instability, where compounding errors across turns lead to escalating KL divergence and unreliable teacher supervision. Building on this insight, we propose TCOD, a simple and principled framework that controls the trajectory depth exposed to the student during training, instantiated through two practical variants: Forward-to-Backward (F2B) and Backward-to-Forward (B2F). Extensive experiments demonstrate that TCOD consistently stabilizes training, recovers small models from collapse, improves success rates for larger models, and reduces total training time compared to vanilla OPD. Beyond practical improvements, TCOD opens new directions for curriculum-guided training of long-horizon autonomous agents.

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 *The twelfth international conference on learning representations*, 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.

OpenClaw Contributors. Openclaw: Your own personal ai assistant. https://github.com/openclaw/openclaw, 2026. GitHub repository.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. *arXiv preprint arXiv:2505.10978*, 2025.

Zhaoyan Gong, Zhiqiang Liu, Songze Li, Xiaoke Guo, Yuanxiang Liu, Xinle Deng, Zhizhen Liu, Lei Liang, Huajun Chen, and Wen Zhang. Temp-r1: A unified autonomous agent for complex temporal kgqa via reverse curriculum reinforcement learning. *arXiv preprint arXiv:2601.18296*, 2026.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: incentivizes reasoning in llms through reinforcement learning. *nature*, 645:633–638, 2025.

HKUDS. Cli-anything. https://github.com/HKUDS/CLI-Anything, 2026. GitHub repository.

Ijun Jang, Jewon Yeom, Juan Yeo, Hyunggu Lim, and Taesup Kim. Stable on-policy distillation through adaptive target reformulation. *arXiv preprint arXiv:2601.07155*, 2026.

Woogyeol Jin, Taywon Min, Yongjin Yang, Swanand Ravindra Kadhe, Yi Zhou, Dennis Wei, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models. *arXiv preprint arXiv:2603.07079*, 2026.

Jongwoo Ko, Sara Abdali, Young Jin Kim, Tianyi Chen, and Pashmina Cameron. Scaling reasoning efficiently via relaxed on-policy distillation. *arXiv preprint arXiv:2603.11137*, 2026.

Niklas Lauffer, Xiang Deng, Srivatsa Kundurthy, Brad Kenstler, and Jeff Da. Imitation learning for multi-turn lm agents via on-policy expert corrections. *arXiv preprint arXiv:2512.14895*, 2025.

Xiangyi Li, Wenbo Chen, Yimin Liu, Shenghan Zheng, Xiaokun Chen, Yifeng He, Yubo Li, Bingran You, Haotian Shen, Jiankai Sun, et al. Skillsbench: Benchmarking how well agent skills work across diverse tasks. *arXiv preprint arXiv:2602.12670*, 2026.

Kevin Lu and Thinking Machines Lab. On-policy distillation. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20251026. https://thinkingmachines.ai/blog/on-policy-distillation.

Mike A Merrill, Alexander G Shaw, Nicholas Carlini, Boxuan Li, Harsh Raj, Ivan Bercovich, Lin Shi, Jeong Yeon Shin, Thomas Walshe, E Kelly Buchanan, et al. Terminal-bench: Benchmarking agents on hard, realistic tasks in command line interfaces. *arXiv preprint arXiv:2601.11868*, 2026.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2024.

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

Emiliano Penaloza, Dheeraj Vattikonda, Nicolas Gontier, Alexandre Lacoste, Laurent Charlin, and Massimo Caccia. Privileged information distillation for language models. *arXiv preprint arXiv:2602.04942*, 2026.

Taiwei Shi, Yiyang Wu, Linxin Song, Tianyi Zhou, and Jieyu Zhao. Efficient reinforcement finetuning via adaptive curriculum learning. *arXiv preprint arXiv:2504.05520*, 2025.

Weijie Shi, Yanxi Chen, Zexi Li, Xuchen Pan, Yuchang Sun, Jiajie Xu, Xiaofang Zhou, and Yaliang Li. R3l: Reflect-then-retry reinforcement learning with language-guided exploration, pivotal credit, and positive amplification. *arXiv preprint arXiv:2601.03715*, 2026.

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.

Jiaqi Wang, Kevin Qinghong Lin, James Cheng, and Mike Zheng Shou. Think or not? selective reasoning via reinforcement learning for vision-language models. *arXiv preprint arXiv:2505.16854*, 2025a.

Ruiyi Wang and Prithviraj Ammanabrolu. A practitioner's guide to multi-turn agentic reinforcement learning. *arXiv preprint arXiv:2510.01132*, 2025.

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.

Yinjie Wang, Xuyang Chen, Xiaolong Jin, Mengdi Wang, and Ling Yang. Openclaw-rl: Train any agent simply by talking. *arXiv preprint arXiv:2603.10165*, 2026.

Zhenting Wang, Guofeng Cui, Yu-Jhe Li, Kun Wan, and Wentian Zhao. Dump: Automated distribution-level curriculum learning for rl-based llm post-training. *arXiv preprint arXiv:2504.09710*, 2025b.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, 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 R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*, 2022b.

Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. On-policy context distillation for language models. *arXiv preprint arXiv:2602.12275*, 2026.

Yang Zhang, Amr Mohamed, Hadi Abdine, Guokan Shang, and Michalis Vazirgiannis. Beyond random sampling: Efficient language model pretraining via curriculum learning. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5776–5794, 2026.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026.

Appendix

Contents

A Limitations and Future Work **13**

B Additional Observation **13**

C Algorithm for TCOD-F2B/B2F **13**

D Experiment Details **15**

D.1 Benchmark Environments 15

D.2 Baselines 15

D.3 Implementation Details for Stability 16

D.4 Training Hyperparameters 16

D.5 Evaluation Hyperparameters 16

D.6 More experiments results 16

E Environment Prompts **17**

E.1 ALFWorld Prompts 18

E.2 ScienceWorld Prompts 19

E.3 WebShop Prompts 20

A Limitations and Future Work

While TCOD offers practical benefits, it also comes with a few limitations that point to interesting future work. TCOD-B2F relies on pre-collected successful teacher trajectories, which may require additional trajectory collection overhead. In such cases, the forward-to-backward variant (TCOD -F2B) provides a drop-in alternative that requires no demonstrations. Although we empirically observe that TCOD's fixed curriculum schedule is robust across the three benchmarks and model sizes we studied, the optimal pace may vary with different environments or student–teacher pairs. An adaptive mechanism that automatically adjusts the horizon based on the student's learning progress—such as through an exponential moving average of the KL divergence—could further improve generality; we consider this a promising direction for future investigation. Our evaluation focuses on three text-based multi-turn benchmarks; extending TCOD to multimodal or physically embodied environments is an important next step to assess its generality. These considerations do not compromise TCOD's practical effectiveness, but instead highlight promising directions for further improvement.

B Additional Observation

We systematically evaluate student–teacher pairs across the Qwen3 and Qwen2.5 model families, including both larger-scale and domain-adapted teachers. For Qwen3, we use Qwen3-30B-A3B-Instruct as the teacher and Qwen3-{0.6, 1.7, 4}B as students. For Qwen2.5, we adopt a GRPO-trained Qwen2.5-7B model as the teacher and Qwen2.5-{0.5, 1.5, 3, 7}B as students.

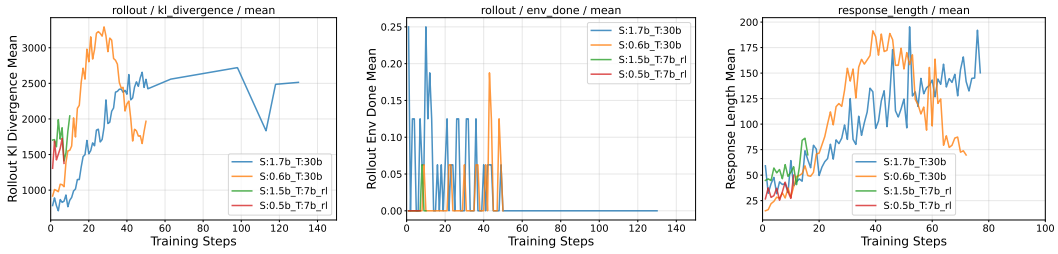
Observation 1: KL escalation and success rate collapse co-occur in small models ($<$3B). Unlike prior work in single-turn settings (e.g., math or QA), where the KL divergence typically decreases and stabilizes during training, we observe a fundamentally different behavior in multi-turn agent environments. As shown in Figure 7, when training small student models (Qwen3-0.6B, 1.7B and Qwen2.5-0.5B, 1.5B) with vanilla OPD, the trajectory-level KL divergence increases sharply as training progresses. This escalation is accompanied by a simultaneous collapse of the success rate to nearly zero. Moreover, response lengths grow steadily across turns, indicating compounding errors and increasingly off-distribution trajectories. Together, these results suggest that, in multi-turn settings, small models fail to maintain alignment with the teacher under their own rollout distribution, leading to unstable training dynamics and ineffective supervision.

Observation 2: Teacher–student matching matters; stronger teachers are not always better. We further examine the impact of teacher–student pairing in Figure 8. For a 3B student, training under both a strong 30B teacher and a 7B RL teacher leads to similar outcomes: the KL divergence decreases steadily and the success rate improves at comparable rates, indicating that increasing teacher strength beyond a certain point does not yield additional benefits. In contrast, when the student capacity matches the teacher more closely (7B student with 7B RL teacher), the KL divergence converges significantly faster and the success rate rises more rapidly, outperforming both 3B student settings. This suggests that an appropriate capacity match between teacher and student is more critical than absolute teacher strength; overly strong teachers do not necessarily improve, and may even limit, distillation efficiency in multi-turn settings.

C Algorithm for TCOD-F2B/B2F

Algorithm 3 and Algorithm 4 present the complete training procedures for TCOD-F2B and TCOD-B2F, respectively, integrating the curriculum pacing strategy and implementation details described in Section 4.2.

In TCOD-F2B (Algorithm 3), the student policy π_θ rolls out the trajectory for k steps at each training iteration, where k is progressively expanded according to the linear pacing schedule

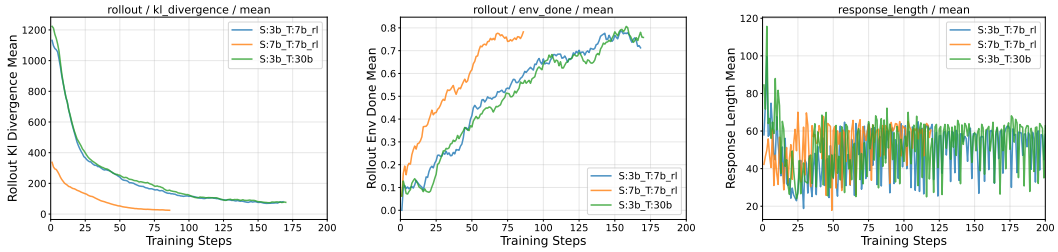


(a) Trajectory-level KL escalates during training.

(b) Success rate collapses to zero as KL spikes.

(c) Response length increases across turns.

Figure 7: **KL Escalation and success rate across Teacher–Student Pairs.** We evaluate Qwen3-{0.6B, 1.7B} (teacher: Qwen3-30B-A3B-Instruct) and Qwen2.5-{0.5B, 1.5B} (teacher: Qwen2.5-7B-RL) under vanilla OPD on ALFWorld.



(a) KL divergence.

(b) Success rate.

(c) Response length.

Figure 8: **Horizon-Induced KL Escalation across Teacher–Student Pairs.** We evaluate Qwen2.5-{3B, 7B} (teacher: Qwen3-30B-A3B-Instruct, Qwen2.5-7B-RL) under vanilla OPD on ALFWorld.

in Equation 4. By concentrating the distillation signal on early-turn states at the beginning of training and gradually extending the horizon, the student builds a robust foundation before being exposed to the full trajectory, effectively mitigating compounding errors and preventing KL collapse.

Algorithm 3 Temporal Curriculum On-Policy Distillation: TCOD-F2B

Input: Student π_θ, Teacher π_ϕ, Environment $\mathcal{E}$, total steps N, curriculum parameters k_{start}, η
Output: Trained student policy π_θ
for $n = 1, 2, \ldots, N$ **do**
  $k \leftarrow \min(k_{\text{start}} + \lfloor n/\eta \rfloor,\ T_{\max})$
  Initialize $s_0 \sim \mathcal{E}$, history $h_0 \leftarrow \varnothing$
  for $t = 0, 1, \ldots, k-1$ **do**
    Sample $a_t \sim \pi_\theta(\cdot \mid h_t)$; execute a_t; update h_{t+1}
  end for
  $\mathcal{L} \leftarrow \sum_{t=0}^{k} \mathcal{D}_{\text{KL}}\big(\pi_\phi(a_t \mid h_t) \,\|\, \pi_\theta(a_t \mid h_t)\big)$
  Update $\theta \leftarrow \theta - \nabla_\theta \mathcal{L}$
end for
return π_θ

In TCOD-B2F (Algorithm 4), the teacher policy π_ϕ first replays the initial $L - k$ steps from a pre-collected successful trajectory τ^* without contributing to the gradient, placing the student at a vetted checkpoint state. The student then takes over for the remaining k steps, learning to complete the task from progressively earlier starting points as k increases. By the end of training, the teacher prefix is fully eliminated ($k = L$), ensuring the student executes the complete trajectory end-to-end and fully closing the train-test distribution gap.

Algorithm 4 Temporal Curriculum On-Policy Distillation: TCOD-B2F

Input: Student π_θ, Teacher π_ϕ, Environment $\mathcal{E}$, total steps N, curriculum parameters k_{start}, η
Output: Trained student policy π_θ
Pre-collect teacher successful trajectories $\mathcal{T}^* \leftarrow \{\tau^*\}$
for $n = 1, 2, \ldots, N$ **do**
  $k \leftarrow \min(k_{\text{start}} + \lfloor n/\eta \rfloor,\ L)$
  Sample $\tau^* \in \mathcal{T}^*$ with length L; initialize $s_0 \sim \mathcal{E}$
  for $t = 0, 1, \ldots, L-k-1$ **do**
    Execute teacher action a_t^* **(stop gradient)**; update h_{t+1}
  end for
  for $t = L-k, \ldots, L$ **do**
    Sample $a_t \sim \pi_\theta(\cdot \mid h_t)$; execute a_t; update h_{t+1}
  end for
  $\mathcal{L} \leftarrow \sum_{t=L-k}^{L} \mathcal{D}_{\text{KL}}\big(\pi_\phi(a_t \mid h_t) \,\|\, \pi_\theta(a_t \mid h_t)\big)$
  Update $\theta \leftarrow \theta - \nabla_\theta \mathcal{L}$
end for
return π_θ

D Experiment Details

D.1 Benchmark Environments

ALFWorld (Shridhar et al., 2020) is a text-based embodied environment requiring navigation and object manipulation across six categories of household tasks. ALFWorld provides *seen* and *unseen* splits: the seen split tests performance in environments present during training, while the unseen split requires the agent to operate in novel room layouts and object combinations, serving as our OOD evaluation. For ALFWorld, we further build a **Hard** set of 121 tasks where the teacher fails under pass@10 sampling on the training split. This set serves as a more challenging OOD evaluation to test whether TCOD can generalize beyond the teacher's own capability boundary.

Webshop (Yao et al., 2022a) is a web-based environment requiring the agent to search and select products that match a given user instruction across multi-turn interactions with a simulated e-commerce platform.

ScienceWorld (Wang et al., 2022) is a text-based environment that tests scientific reasoning across 30 task types aligned with the elementary science curriculum. The agent receives a score between 0 and 100 at the end of each task based on task completion.

D.2 Baselines

To rigorously assess the effectiveness of TCOD, we benchmark against the following paradigms, establishing clear performance boundaries for the student models:

Teacher (Upper Bound): The performance of the expert policy (π_ϕ) is evaluated directly on the environment. In standard distillation, this represents the theoretical upper bound, as the primary goal is to recover this capability within the smaller student model. Notably, our evaluation on the Train Hard split (Sec 5.3) investigates whether TCOD can even generalize beyond this upper limit.

Zero-Shot Student (Lower Bound): The base student model (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 reasoning capability in the agentic environments.

Supervised Fine-Tuning (SFT): The fundamental imitation learning baseline. The student model is fine-tuned via standard negative log-likelihood (NLL) loss strictly on the successful

trajectories (τ^*) pre-collected from the Teacher for 2 epochs, suffering from the well-known exposure bias in multi-turn settings.

Vanilla On-Policy Distillation (OPD): The standard multi-turn adaptation of recent OPD methods. The student is trained to minimize the token-level KL divergence against the teacher's distribution over the student's entire generated trajectory (full rollouts), without any horizon constraints or temporal curriculum. This serves as the direct baseline to demonstrate the Trajectory-Level KL Instability.

D.3 Implementation Details for Stability

While the core TCOD framework is conceptually straightforward, several practical design choices significantly impact training stability and efficiency in real-world deployments. All experiments were conducted on 8× NVIDIA H20 (96GB) GPUs. We describe our key implementation strategies:

Asynchronous Rollout and Training. To maximize GPU utilization, we decouple trajectory collection and model optimization into separate asynchronous processes. We use a pool of **actor** processes for rollout to continuously sample trajectories, while a central **learner** process for training uses these trajectories from a shared buffer and performs gradient updates. We use a lock-free ring buffer to minimize synchronization overhead. In our experiments, we allocate 4×H20 GPUs for actors and 2×H20 GPUs for learners. Moreover, we use the remaining 2×H20 GPUs for the teachers.

Staleness-Aware Sub-trajectory Experience Replay. To maximize sample efficiency in multi-turn environments, we decompose each complete trajectory into a set of recursive sub-trajectories. Specifically, for a trajectory of length n, we store each prefix sequence $\tau_{1:t} = (s_0, a_0, \ldots, s_t)$ as an independent experience entry in the replay buffer for $t \in \{1, \ldots, n\}$. To prevent the input context from exceeding the model's effective memory limit, leading to training instability, we encapsulate the interaction history within the prompt as a structured context. Consequently, the number of rollouts generated per batch is dynamic, depending on the varying lengths of collected trajectories. In our asynchronous setting, each trajectory is tagged with the version number n of the policy π_{θ_n} used for collection. We implement a staleness filter that discards any experience where $n_{\text{current}} - n_{\text{old}} > \Delta_{\max}$. Empirically, we find that $\Delta_{\max} = 2$ provides an optimal balance between sample efficiency and the strictness of the on-policy constraint.

D.4 Training Hyperparameters

We conduct training across three text-based interactive environments: ALFWorld, ScienceWorld, and WebShop. The training configuration is summarized in Table 4.

D.5 Evaluation Hyperparameters

For evaluation, we assess model performance on three test sets: `test_unseen`, `test`, and `train_hard` (ALFWorld only). The evaluation hyperparameters are consistent across all environments as shown in Table 5.

D.6 More experiments results

Detailed success rate for TCOD-B2F As shown in Figure 9 and Figure 10, TCOD-B2F exhibits a characteristic non-monotonic training dynamic. Specifically, the rollout success rate is initially high—since training starts from short horizons—then drops as the curriculum expands to longer trajectories, and finally recovers as the student adapts to the increased difficulty. A similar pattern is observed in the *valid seen* split, where the success rate also decreases mid-training before improving again.

In contrast, the *valid unseen* and *train hard* splits remain relatively stable throughout training, without pronounced drops. This suggests that the intermediate degradation is not due to

Table 4: Training hyperparameters for TCOD across all environments.

Hyperparameter	Value
Algorithm	
Algorithm type	On-Policy Distillation
Advantage function	Multi-turn OPD
KL coefficient	1.0
Learning rate	1×10^{-6}
Gradient clipping	1.0
Repeat times	1
Sample strategy	Staleness control (max staleness: 2)
Training	
Total training steps	250
Batch size	16
Train batch size	64
Save interval	250
Evaluation interval	5 steps
Model Configuration	
Max prompt tokens	10,240
Max response tokens	512
Inference (Rollout)	
Temperature (training)	1.0
Temperature (evaluation)	0.4
Logprobs	Enabled (all tokens)
Seed	42
Environment-Specific	
ALFWorld max steps	30
ScienceWorld max steps	30
WebShop max steps	15
TCOD Curriculum	
Workflow name	B2F, F2B
Starting step	1
Checkpoint steps	2, 4, 6
Distributed Training	
Number of nodes	1
GPUs per node	8
Tensor parallel size	2
Sequence parallel size	2 (Ulysses)
Max tokens per GPU	16,384
GPU memory utilization	0.7
Data type	BFloat16

overfitting or instability, but rather reflects a controlled curriculum transition. Overall, these results indicate that TCOD-B2F introduces temporary difficulty as the horizon expands, yet maintains stable generalization while ultimately improving performance, validating the effectiveness of progressive horizon expansion.

E Environment Prompts

This section provides the detailed prompts used for each environment during training and evaluation. All prompts follow a consistent structure: task description, observation-action history, current observation, admissible actions, and thinking/action format requirements.

Table 5: Evaluation hyperparameters for TCOD across all environments.

Hyperparameter	Value
Generation	
Maximum tokens	4,096
Temperature	0.4 (evaluation)
Top-p	1.0
Top-k	-1
MinP	0.0
Environment	
Max environment steps	30 (ALFWorld, ScienceWorld) / 15 (WebShop)
History length	2 steps
Parallelization	
Number of workers	8 (evaluation)
Process timeout	3,600 seconds
Synchronization style	Dynamic by explorer
Data	
ALFWorld test sets	test_unseen.jsonl, test.jsonl, train_hard.jsonl
ScienceWorld test sets	Test split from training data
WebShop test sets	Test split from training data

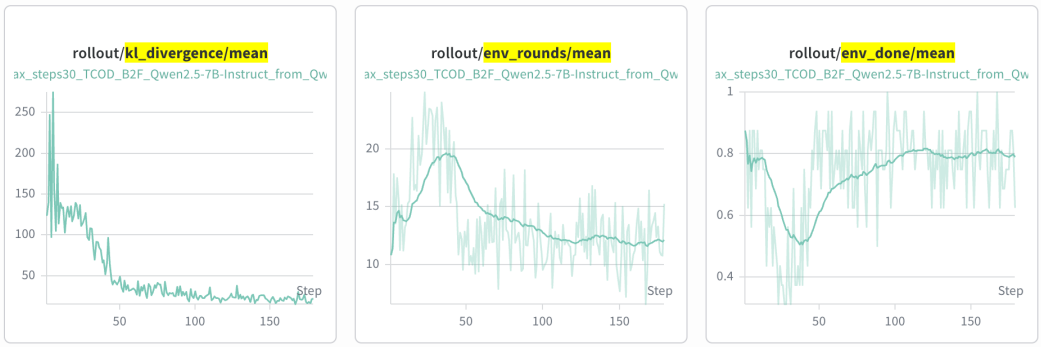


Figure 9: Training dynamics of TCOD-B2F ($\eta = 2$), including KL divergence, student action horizon, and success rate, for a Qwen2.5-7B student distilled from a GRPO-trained Qwen2.5-7B teacher on ALFWorld.

E.1 ALFWorld Prompts

ALFWorld is an embodied AI task requiring agents to navigate household environments and complete object manipulation tasks. The prompt structure emphasizes step-by-step reasoning within `<thought>` tags followed by executable actions in `<action>` tags.

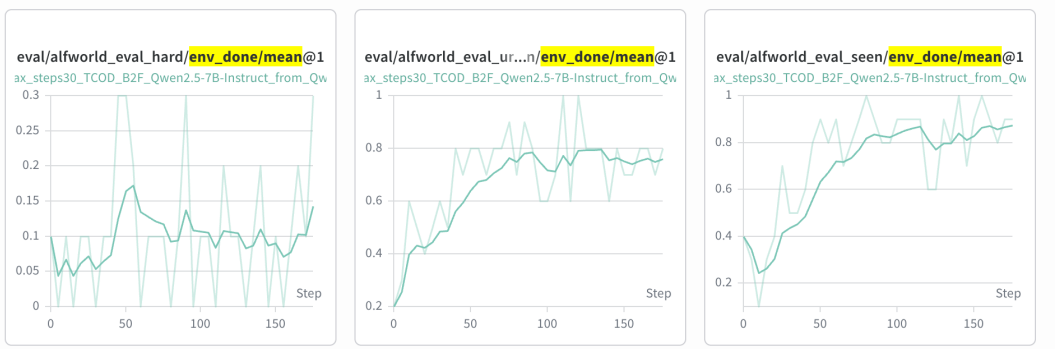


Figure 10: Success rates of TCOD-B2F ($\eta = 2$), including train hard (left), valid unseen(middle), and valid seen(right), for a Qwen2.5-7B student distilled from a GRPO-trained Qwen2.5-7B teacher on ALFWorld.

ALFWorld Task Prompt Template

```
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
current step and present it within <action> </action> tags.
```

E.2 ScienceWorld Prompts

ScienceWorld focuses on scientific reasoning tasks in a text-based laboratory environment. The prompt structure guides agents through multi-step experiments requiring domain knowledge and procedural reasoning.

ScienceWorld Task Prompt Template

```
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.
```

E.3 WebShop Prompts

WebShop presents e-commerce shopping tasks requiring agents to navigate product listings, apply filters, and make purchasing decisions based on natural language instructions. The prompt emphasizes matching user preferences to available product attributes.

WebShop Task Prompt Template

```
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
current step and present it within <action> </action> tags.
```

Action Format for WebShop. WebShop uses a specific action format with two primary action types:

- `search[<query>]`: Search for products using a text query (only available when search bar is present)
- `click[<button_name>]`: Click on interactive elements (e.g., product links, filter buttons, pagination)

The available actions are dynamically presented based on the current page state, including clickable elements and search bar availability.