STEP-LEVEL ON-POLICY DISTILLATION: INTERPOLATING BETWEEN ON-POLICY DISTILLATION AND SUPERVISED FINE-TUNING FOR AGENT

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation (OPD) aligns a student model with a teacher's logit distribution on student-generated trajectories. This approach has achieved strong empirical gains and can often surpass conventional off-policy distillation with substantially less data. However, standard token-level OPD can provide only fragmented corrections along an erroneous student trajectory and cannot unfold a complete and correct repair path. Motivated by this limitation, we propose *Step-Level On-Policy Distillation* (SOPD), which combines the long-horizon correction of supervised fine-tuning (SFT) with the on-policy advantage of OPD to provide step-level supervision over complete student-generated trajectories. We show that, at different limits of step length, SOPD reduces to SFT or approximates OPD. Compared with SFT, the teacher responses in SOPD are conditioned on student trajectories and therefore align more closely with student-visited states; compared with OPD, SOPD provides longer-horizon corrections rather than fragmented token-level guidance. On ALFWorld, the Qwen2.5-7B student trained with SOPD improves Seen and Unseen success rates over SFT by 36.42 and 38.58 percentage points, respectively, and over Vanilla OPD by 15.34 and 15.17 points, while reducing action rounds. We hope this work offers a new perspective for future research on distillation methods.

1 INTRODUCTION

Recently, on-policy distillation (OPD) (Agarwal et al., 2024; Yang et al., 2025; Lu & Lab, 2025) has emerged as an effective post-training paradigm for improving the capabilities of large language models (LLMs). Unlike earlier off-policy distillation methods (Taori et al., 2023; Guha et al., 2025), which train the student on teacher-generated trajectories, OPD allows the student to learn from teacher supervision—namely, predicted logits—on student-generated tokens.

Standard OPD queries a teacher distribution at every token generated by the student. Although this provides relatively dense supervision, it also fragments the teacher's correction signal: in a single OPD rollout, the student receives only one-token guidance at each position rather than a continuous and correct segment-level correction (Jiang et al., 2026). Recent work has revisited the potential of DAgger-style methods for training LLM agents, seeking to preserve SFT's long-horizon supervision while generating more training trajectories from student states (Lauffer et al., 2025). DAgger, however, inherently inserts teacher responses into the student-generated trajectory, substantially changing the trajectory the student would otherwise have produced.

We combine the on-policy property of OPD with the long-horizon supervision of SFT and propose **Step-Level On-Policy Distillation (SOPD)**. SOPD provides step-level supervision over a complete student-generated trajectory. After the student completes the full trajectory, SOPD asks the teacher to generate one step from the student prefix at every student step. Our method reduces to SFT when the number of steps approaches one and approximates forward-KL OPD when the step length approaches one token.

We evaluate SOPD on agent tasks, defining a step as one actual environment-interaction turn. With a domain-adapted teacher on ALFWorld, SOPD improves Seen and Unseen success over SFT, OPD, TCOD, and OEC for both 3B and 7B students while reducing the number of interaction rounds. With

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.

a Qwen3-30B-A3B-Instruct teacher, SOPD also outperforms Vanilla OPD and both reported TCOD variants across WebShop, ALFWorld, and ScienceWorld for Qwen3-1.7B and Qwen3-4B students. We also conduct experiments on mathematical reasoning tasks (Appendix A).

Moreover, standard OPD requires access to teacher logits, whereas SOPD requires only teacher-generated responses and can therefore be applied more directly to black-box distillation. The total amount of teacher generation per sample is close to the length of one complete response, making the practical cost of black-box calls comparable to standard SFT. In agent tasks, each teacher call generates only one step, so distillation avoids the additional waiting time caused by teacher–environment interaction and tool execution.

Our contributions are:

- We combine the respective advantages of OPD and SFT and propose Step-Level On-Policy Distillation (SOPD). Our method is also better suited than standard OPD to black-box distillation.
- SOPD connects SFT and OPD, combining SFT's long-horizon supervision with OPD's on-policy property. Experiments show substantial improvements over both distillation methods.
- We report agent experiments across multiple benchmarks and model sizes, with broad baseline comparisons showing consistent improvements over the evaluated methods.

2 Related Work

Knowledge distillation. Knowledge distillation transfers a teacher distribution into a smaller student (Hinton et al., 2015). Sequence-level distillation trains on teacher-generated sequences and reduces the train–test mismatch created by token-level targets (Kim & Rush, 2016). MiniLLM instead minimizes reverse KL and samples from the student during optimization (Gu et al., 2024). These methods establish sequence generation and student sampling as distinct design choices. SOPD combines student-induced prefixes with teacher-generated local targets.

On-policy language-model distillation. Generalized Knowledge Distillation trains the student on its own generations and queries teacher logits at those states (Agarwal et al., 2024). Recent work has expanded OPD in several directions. On-policy self-distillation (OPSD) lets the same model serve as student and teacher under ordinary and privileged contexts, achieving stronger reasoning performance and higher training efficiency than off-policy distillation without requiring a separate teacher model (Zhao et al., 2026). Extrapolative OPD (ExOPD) uses reward extrapolation to relax direct imitation of the teacher, allowing a student that fuses multiple experts to surpass the teachers' performance boundary (Yang et al., 2026). SOD, StepOPSD, and SAF-OPD combine step-level distillation signals with GRPO or RLVR to alleviate sparse trajectory rewards and local credit-assignment difficulties (Zhong et al., 2026; Zhang et al., 2026; Ding et al., 2026). GAD realizes black-box on-policy distillation through a generator–discriminator game and outperforms sequence-level distillation while accessing only teacher-generated text (Ye et al., 2025). Together, these advances show that OPD has expanded from standard teacher–student logit alignment to self-distillation, surpassing the teacher, integration with RL, and black-box distillation.

At the same time, recent work has identified an inherent limitation of standard OPD. Trajectory-Refined Distillation (TRD) observes that supervision from a single rollout does not contain a complete counterfactual correction trajectory: even with a perfect teacher, token-wise OPD can provide only fragmented corrections along an erroneous student trajectory rather than unfold a complete, correct repair path (Jiang et al., 2026). Later targets remain conditioned on the original erroneous student prefixes instead of prefixes that incorporate earlier teacher corrections. TRD constructs a repair path by rewriting the complete student trajectory; SOPD instead preserves the complete student trajectory while allowing the teacher to generate one coherent local correction within each natural step.

Distillation and guidance for multi-turn agents. Action errors compound in interactive agents because every action changes the next observation. ALFWorld instantiates this dependence through textual actions in embodied household tasks (Shridhar et al., 2020). Existing approaches differ in who generates the training states and whether teacher outputs are executed in the environment.

On-policy Expert Correction (OEC) begins with a student rollout, switches control to an expert partway through the trajectory, and applies SFT to reward-filtered expert suffixes (Lauffer et al., 2025). Agent-RLVR lets the student attempt a task, augments failed attempts with plans, error feedback, or interaction information from an external model, and then applies RL with verifiable environment rewards; the guidance is used only during training (Da et al., 2025). These methods help the learner enter successful regions through expert suffixes or guidance-enhanced reward optimization, rather than producing an independent target at every fixed student state.

TCOD uses a temporal curriculum to control which interval of a multi-turn trajectory is assigned to the student: forward-to-backward expands the student's prefix of control, whereas backward-to-forward begins from a teacher trajectory prefix and moves student control toward the beginning (Wang et al., 2026). Guided-OPD chooses a teacher or student action at each environment turn according to a curriculum and anneals the probability of teacher intervention; selected teacher actions are executed, so training visits states induced by a teacher–student mixture (Li et al., 2026). ReOPD avoids repeated environment interaction during training by replaying prefixes from pre-collected teacher trajectories, letting the student act at selected steps, and requesting dense teacher supervision (Liao et al., 2026).

These methods largely depend on teacher intervention to adjust the generated trajectory, which weakens the on-policy nature of training. SOPD instead provides guidance over a complete student trajectory. At every student-generated step, the teacher produces one independent fragment conditioned on the corresponding student prefix, yielding supervision that remains more faithfully on-policy. Because each teacher call generates only one step, teacher generation does not wait for environment interaction or returned tool results; after the student trajectory is complete, the step-level teacher queries can also be issued in parallel.

3 Preliminaries: SFT and OPD

We first establish notation for the two training paradigms that delimit SOPD. Let $x \sim \mathcal{D}$ be a training prompt, π_θ the student to be optimized, and π_T a fixed teacher.

Supervised fine-tuning (SFT). SFT applies teacher forcing to a complete teacher or expert sequence $y^T = (y_1^T, \ldots, y_L^T)$:

$$\mathcal{L}_{\mathrm{SFT}} = \mathbb{E}_{(x,y^T)\sim\mathcal{D}_T}\left[-\frac{1}{L}\sum_{t=1}^{L}\log \pi_\theta(y_t^T \mid x, y_{<t}^T)\right]. \tag{1}$$

Supervision within a target sequence unfolds along the teacher's own preceding tokens, providing a coherent long-range target. Its contexts, however, come from teacher data rather than the current student, creating an off-policy state-distribution mismatch.

On-policy distillation (OPD). OPD first samples $y^S \sim \pi_\theta(\cdot \mid x)$ and then queries the teacher distribution at every student prefix $h_t^S = (x, y_{<t}^S)$. A common reverse-KL objective is

$$\mathcal{L}_{\mathrm{OPD}} = \mathbb{E}_{x,\, y^S\sim\pi_\theta}\left[\frac{1}{|y^S|}\sum_{t}\mathrm{KL}\big(\pi_\theta(\cdot \mid h_t^S)\,\|\,\pi_T(\cdot \mid h_t^S)\big)\right]. \tag{2}$$

This objective aligns training states with those encountered by the student at inference and supplies dense token-level feedback, but its standard implementation requires teacher logits. Moreover, after the student has entered an erroneous trajectory, the teacher distribution at the next position is still conditioned on an even longer erroneous student prefix. The resulting position-wise targets need not form a correction path that can be executed coherently (Jiang et al., 2026).

4 Step-Level On-Policy Distillation

SOPD preserves the student state occupancy of OPD while retaining coherent multi-token supervision within each local target. It first partitions a complete student trajectory into natural steps and then asks the teacher to generate one target from the beginning of every step. The student trajectory defines the states between steps; a teacher target unfolds autoregressively only within its own step. Figure 1 contrasts OEC, token-wise OPD, and SOPD and illustrates the packed attention mask.

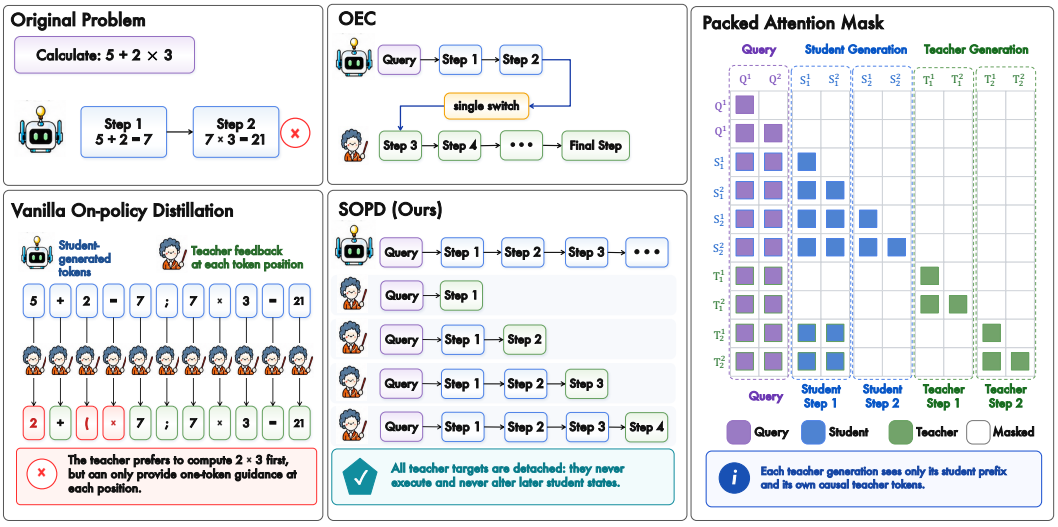

Figure 1: **Overview of Step-Level On-Policy Distillation (SOPD).** The left and center panels compare On-Policy Distillation (OPD), On-Policy Expert Correction (OEC), and SOPD. The right panel illustrates the packed attention mask used for efficient training. We pack the input prompt, the complete student trajectory, and its multiple step-level teacher targets into a single sequence, enabling these step-level supervision instances to be processed in parallel within a single forward pass to ensure training efficiency. Each teacher target can attend only to the input prompt, its corresponding student prefix, and its own preceding causal teacher tokens, while future student steps and other teacher targets remain masked.

4.1 Student Rollouts Fix the Visited States

Let x denote a prompt or initial task state. A rollout student π_{θ_r} generates a complete trajectory

$$\tau_S = (s_1, s_2, \ldots, s_K), \qquad s_k \sim \pi_{\theta_r}(\cdot \mid h_k), \tag{3}$$

where s_k is one natural step and h_k is the student-visited prefix before that step. For static reasoning, $h_k = (x, s_{<k})$. In an interactive environment, h_k also contains the observations returned after the environment executes the student actions in $s_{<k}$.

The partition follows the structure of each domain. For interactive agents, one assistant response and its parsed action form a step. The environment executes that student action and returns the observation that defines the next prefix. For static reasoning, natural reasoning boundaries define the steps; Appendix B specifies the partition used in the supplementary experiments.

SOPD records every h_k before teacher generation. It therefore preserves the state distribution induced by the rollout student:

$$h_k \sim d_k^{\pi_{\theta_r}}. \tag{4}$$

Because the rollout has already fixed the complete trajectory, teacher outputs cannot alter $d_k^{\pi_{\theta_r}}$. Thus, every step begins at a state actually visited by the student rather than at one produced by executing a teacher target.

4.2 Teacher Targets Do Not Alter the Trajectory

Let $q_T(\cdot \mid h_k)$ denote the teacher distribution over the next natural step from student prefix h_k. For every recorded prefix, the teacher generates one target step:

$$\tilde{s}_k \sim q_T(\cdot \mid h_k). \tag{5}$$

Each query contains the original prompt and the preceding *student* steps, but no earlier teacher target. This aligns every target with a state the student actually visited and prevents teacher-written text from

Algorithm 1 Step-Level On-Policy Distillation (SOPD)

Input: Student π_θ, frozen teacher q_T, task source $\mathcal{D}$ or environment $\mathcal{E}$, updates N, learning rate α, step partitioner P
Output: Trained student policy π_θ
for $n = 1, 2, \ldots, N$ **do**
  Sample a task batch B from $\mathcal{D}$ or $\mathcal{E}$; fix $\theta_r \leftarrow \theta$ and $\mathcal{S}_B \leftarrow \emptyset$
  for each task x in rollout batch B **do**
    Generate a complete student trajectory $\tau_S \sim \pi_{\theta_r}(\cdot \mid x)$
    In an environment, execute only student actions and record all observations
    $(s_1, \ldots, s_K) \leftarrow P(\tau_S)$; fix each student-visited prefix h_k
    for $k = 1, 2, \ldots, K$ **in parallel do**
      Generate one teacher step $\tilde{s}_k \sim q_T(\cdot \mid h_k)$
      Add $(h_k, \tilde{s}_k)$ to $\mathcal{S}_B$; never execute or insert $\tilde{s}_k$ into τ_S
    end for
  end for
  Pack shared prefixes and targets; mask loss to teacher-target tokens only
  Compute $\mathcal{L}_{\mathrm{SOPD}}(B)$ using the pooled step-mean in Equation 6
  Update $\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_{\mathrm{SOPD}}(B)$
end for
return π_θ

defining the context of a later target. The teacher supplies generated text only; SOPD requires neither teacher logits nor gradients and therefore supports black-box teachers.

For static reasoning, the teacher stops at the same family of natural boundaries used by the partitioner. For agents, all recorded turn states are submitted after the episode. Because no target appears in another target's context, these teacher queries are conditionally independent, can be generated in parallel, and require neither tool execution nor new environment observations.

4.3 Step Balancing Prevents Long Targets from Dominating

SOPD trains the student to generate each teacher target from its corresponding student prefix. Let $\mathcal{S}_B$ contain every teacher step target collected from rollout batch B. We average token cross-entropy within each target and then average over the pooled steps:

$$\mathcal{L}_{\mathrm{SOPD}}(B) = \frac{1}{|\mathcal{S}_B|} \sum_{(h_k, \tilde{s}_k) \in \mathcal{S}_B} \frac{1}{|\tilde{s}_k|} \sum_{j=1}^{|\tilde{s}_k|} -\log \pi_\theta \left(\tilde{s}_{k,j} \mid h_k, \tilde{s}_{k,<j} \right). \tag{6}$$

The inner average prevents a long teacher step from dominating shorter decisions, and the outer average assigns equal weight to every collected step. A trajectory with more retained steps consequently contributes more step terms. Under stochastic teacher decoding, one target is a Monte Carlo sample of the length-normalized risk

$$\mathbb{E}_{\tilde{s} \sim q_T(\cdot \mid h)} \left[-\frac{1}{|\tilde{s}|} \log \pi_\theta(\tilde{s} \mid h) \right]. \tag{7}$$

Under greedy decoding, $q_T = \delta_{g_T(h_k)}$, and the step term equals $|g_T(h_k)|^{-1} \mathrm{KL}(\delta_{g_T(h_k)} \| \pi_\theta)$. This length normalization distinguishes Equation 6 from unnormalized sequence-level forward KL.

The packed implementation shares repeated prefixes without changing Equation 6. It stores the prompt, the student steps needed by every prefix, and all teacher targets in one packed example. Its attention mask lets target $\tilde{s}_k$ attend to x, $s_{<k}$, and its own causal target prefix, while blocking $s_{\geq k}$ and every other teacher target. Loss labels cover teacher tokens only.

Algorithm 1 summarizes the synchronous procedure. In the asynchronous implementation of Section 4.6, version-tagged rollout policies supply the same student-prefix targets, with rollout-to-learner lag bounded by $\Delta_{\max} = 2$.

4.4 SOPD Connects Two Granularity Limits

The step granularity locates SOPD between sequence SFT and token-level OPD.

One-step limit: sequence SFT. If an entire response is the sole step, then $K = 1$, $h_1 = x$, and the teacher generates a complete target $\tilde{y} \sim q_T(\cdot \mid x)$. Equation 6 becomes

$$\mathcal{L}_{K=1} = -\frac{1}{|\tilde{y}|} \sum_j \log \pi_\theta(\tilde{y}_j \mid x, \tilde{y}_{<j}), \tag{8}$$

which is the sequence teacher-forcing loss in Equation 1. Caching the teacher target yields ordinary teacher-data SFT; online generation changes how the data are acquired, not the optimization objective.

One-token limit: a sampled forward-KL OPD objective. If every step contains exactly one token, the step at position t begins at student prefix h_t^S, and the teacher samples $\tilde{y}_t \sim q_T(\cdot \mid h_t^S)$. Taking expectation over this sample gives

$$\begin{aligned} \mathbb{E}_{\tilde{y}_t \sim q_T}[-\log \pi_\theta(\tilde{y}_t \mid h_t^S)] &= H\big(q_T(\cdot \mid h_t^S), \pi_\theta(\cdot \mid h_t^S)\big) \\ &= H\big(q_T(\cdot \mid h_t^S)\big) + \mathrm{KL}\big(q_T(\cdot \mid h_t^S) \| \pi_\theta(\cdot \mid h_t^S)\big)\,. \end{aligned} \tag{9}$$

The teacher entropy is constant with respect to θ, so minimizing expected cross-entropy is equivalent to minimizing forward KL on student-visited prefixes. A single sampled teacher token is a Monte Carlo estimate of that objective. A greedy teacher instead yields hard imitation of the teacher's argmax action, not an unbiased estimate of the full-distribution forward KL.

A natural step lies between these limits. Its start remains student-selected, retaining OPD's student-state coverage; its multiple target tokens unfold along the teacher's own prefix, retaining the locally coherent supervision of SFT. The teacher can therefore express an executable correction within a step without changing the state at which the next student step was observed.

4.5 Relation to Other Distillation Procedures

Off-policy sequence distillation trains on teacher-generated trajectories (Kim & Rush, 2016). Standard OPD queries teacher probabilities along student outputs and minimizes a token-level divergence (Agarwal et al., 2024). SOPD retains student-induced states but replaces logit access with local teacher generation.

ExOPD uses reward extrapolation to enable a student to exceed a single teacher (Yang et al., 2026); SOPD uses no reward in its objective. TCOD assigns control intervals in a multi-turn trajectory through a temporal curriculum (Wang et al., 2026); Guided-OPD allows teacher actions to enter the environment according to a curriculum (Li et al., 2026); and ReOPD replays offline teacher prefixes instead of interacting with the environment throughout training (Liao et al., 2026). SOPD instead keeps the natural temporal order, queries every student-visited step, and never executes a teacher target. Step granularity together with student-state anchoring defines the method.

4.6 Asynchronous Training Details for Stability

While the core SOPD framework is conceptually straightforward, several practical design choices significantly impact training stability and efficiency in real-world deployments. All experiments were conducted on $8\times$ AMD MI308 (192GB) GPUs. We describe our key implementation strategies, following the asynchronous organization of TCOD (Wang et al., 2026).

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 $4\times$ MI308 GPUs for actors and $2\times$ MI308 GPUs for learners. Moreover, we use the remaining $2\times$ MI308 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 L, we store each prefix sequence $\tau_{1:t} = (o_0, a_0, \ldots, o_t)$ as an independent experience entry in the replay buffer for $t \in \{1, \ldots, L\}$. Here o_t denotes an environment

observation and a_t a student action; the entry for turn t retains its pre-action student history and independently generated teacher-step target. 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.

5 EXPERIMENTS

We evaluate SOPD on interactive agent tasks, defining each natural step as one environment turn. Following TCOD (Wang et al., 2026), we first study ALFWorld with a domain-adapted teacher, then compare two student sizes across WebShop, ALFWorld, and ScienceWorld with a larger general-purpose teacher. We also conduct experiments on mathematical reasoning tasks; their setup, results, and step statistics are reported in Appendix A.

5.1 ALFWORLD AGENT INTERACTION

Setup. Following TCOD (Wang et al., 2026), we use Qwen2.5-3B and Qwen2.5-7B students with a Qwen2.5-7B teacher trained by GRPO on ALFWorld. We evaluate on the Valid Seen and Valid Unseen splits, containing 140 and 134 tasks, respectively. We report success rate (SR), the percentage of successfully completed tasks, and Rounds, the average number of environment interactions per task. Baselines include the zero-shot student, SFT, Vanilla OPD, TCOD-B2F/F2B, and On-policy Expert Correction (OEC) (Lauffer et al., 2025). OEC trains on expert continuations after switching control from the student, whereas SOPD generates teacher targets at the fixed student-visited states.

Performance across model sizes. Table 1 shows that SOPD consistently improves success rate and reduces action rounds across both student sizes. With Qwen2.5-3B, SOPD reaches 84.29% on Valid Seen and 82.09% on Valid Unseen, improving over Vanilla OPD by 18.57 and 21.64 points while reducing rounds by 3.53 and 4.33. With Qwen2.5-7B, SOPD reaches 90.71% and 87.31%, with gains of 15.34 and 15.17 points and 3.69 and 2.24 fewer rounds. SOPD achieves the best success rate and the fewest action rounds in every student-size and split combination, indicating more effective and efficient interaction policies.

Comparison with expert correction. SOPD also outperforms OEC on both splits. The success-rate gains are 2.86 and 2.24 points for the 3B student, and 5.00 and 7.46 points for the 7B student, with fewer action rounds in all four comparisons. Moreover, the 7B SOPD student exceeds the domain-trained teacher on both Valid Seen and Valid Unseen. These results support the effectiveness of generating step-level corrections at student-visited states.

5.2 CROSS-BENCHMARK EVALUATION WITH QWEN3 STUDENTS

Setup. Following TCOD (Wang et al., 2026), we evaluate Qwen3-1.7B and Qwen3-4B students with a Qwen3-30B-A3B-Instruct teacher across WebShop, ALFWorld, and ScienceWorld, covering online shopping, household interaction, and scientific tasks. We report success rate (%) on the validation sets and the unweighted average across the three benchmarks.

Different benchmarks and model sizes. Table 2 shows that SOPD outperforms Vanilla OPD and both TCOD variants across all three benchmarks and both student sizes. For Qwen3-1.7B, Vanilla OPD achieves only 0.17% average success, whereas SOPD reaches 27.14%, improving over TCOD-F2B by 8.30 points. For Qwen3-4B, SOPD reaches 35.44%, exceeding Vanilla OPD and TCOD-F2B by 7.80 and 5.90 points, respectively. The largest gains occur on WebShop. The 4B student also surpasses the teacher's 30.28% average, with higher success on WebShop and ALFWorld and a slightly lower score on ScienceWorld (18.02% versus 18.42%).

Table 1: **ALFWorld performance with Qwen2.5-3B and Qwen2.5-7B students.** SR is success rate (%) and Rounds is the average number of action rounds per task. The teacher and all baselines except OEC are reported by TCOD (Wang et al., 2026); OEC and SOPD are our results. Shaded rows highlight SOPD; colored arrows report its absolute changes from Vanilla OPD within each student-size block. Bold marks the best student result in each column.

	Valid Seen		Valid Unseen	
Method	SR ↑	Rounds ↓	SR ↑	Rounds ↓
Qwen2.5-7B-RL (Teacher)	85.71	10.61	76.87	13.06
Qwen2.5-3B Student				
Zero-Shot	7.86	28.73	2.24	29.63
SFT	32.14	22.85	25.37	24.16
Vanilla OPD	65.72	14.73	60.45	16.21
TCOD (B2F)	77.86	12.57	70.90	14.56
TCOD (F2B)	81.43	11.76	79.19	12.47
OEC	81.43	11.97	79.85	13.16
SOPD	**84.29** ↑18.57	**11.20** ↓3.53	**82.09** ↑21.64	**11.88** ↓4.33
Qwen2.5-7B Student				
Zero-Shot	9.29	28.34	8.96	28.46
SFT	54.29	18.92	48.73	20.11
Vanilla OPD	75.37	13.18	72.14	13.37
TCOD (B2F)	86.43	11.06	77.61	13.16
TCOD (F2B)	82.14	13.22	76.12	13.22
OEC	85.71	10.80	79.85	12.62
SOPD	**90.71** ↑15.34	**9.49** ↓3.69	**87.31** ↑15.17	**11.13** ↓2.24

Table 2: **Cross-benchmark results with a Qwen3-30B teacher.** Values are success rates (%); Avg is the unweighted mean of the three benchmark scores. The teacher is Qwen3-30B-A3B-Instruct. All rows except SOPD are reported by TCOD (Wang et al., 2026). Shaded rows highlight SOPD; colored arrows report its absolute gains over Vanilla OPD. Bold marks the best student result within each model-size block.

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	21.12	23.87	11.34	18.78
TCOD-F2B	21.78	23.65	11.08	18.84
SOPD	**38.17** ↑38.03	**30.00** ↑29.68	**13.25** ↑13.20	**27.14** ↑26.97
Qwen3-4B Student				
Vanilla OPD	30.12	36.85	15.95	27.64
TCOD-B2F	32.15	38.62	17.46	29.41
TCOD-F2B	31.81	38.95	17.85	29.54
SOPD	**48.43** ↑18.31	**39.86** ↑3.01	**18.02** ↑2.07	**35.44** ↑7.80

5.3 ALFWorld Training Dynamics

Figure 2 shows the 3B SOPD training trajectory corresponding to Table 1. Seen and Unseen success improve overall during training as their interaction lengths decrease. This trend agrees with the

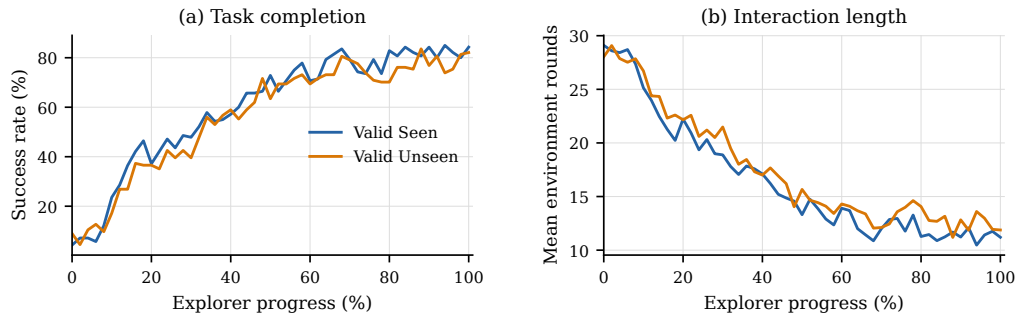

Figure 2: **ALFWorld evaluation of the 3B SOPD student over normalized Explorer progress.** Seen and Unseen success improve over training while interaction length decreases overall. The curves summarize Seen and Unseen evaluations at the saved checkpoints.

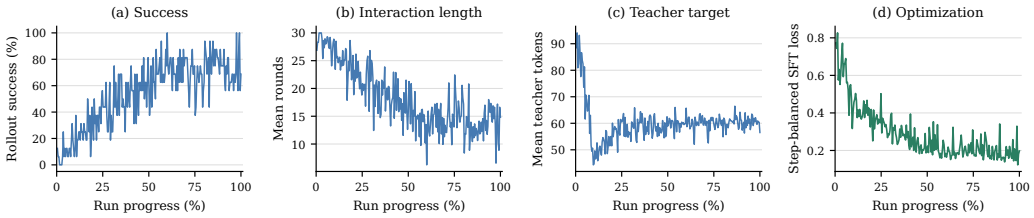

Figure 3: **ALFWorld training diagnostics for the 3B SOPD student.** (a) Explorer-batch task success; (b) mean environment rounds; (c) mean tokens per generated teacher turn target; and (d) step-balanced SFT loss in the Trainer. The horizontal axis normalizes each signal to the endpoint of its corresponding component.

main table and shows that SOPD improves task completion while learning more efficient interaction policies.

Following TCOD's multi-panel analysis (Wang et al., 2026), Figure 3 reports SOPD-specific online signals. The generation-only objective computes no KL quantity. Explorer curves report the current 16-task rollout batches, whereas the optimization panel follows the Trainer. These online signals complement the full-split evaluations in Figure 2.

6 Conclusion

SOPD preserves complete student trajectories and learns from independently generated, step-balanced teacher targets at student-visited states. Its granularity connects sequence SFT with sampled forward-KL OPD. On ALFWorld, SOPD improves success and reduces action rounds over Vanilla OPD, TCOD, and OEC with both 3B and 7B students. It also outperforms Vanilla OPD and TCOD across WebShop, ALFWorld, and ScienceWorld for two Qwen3 student sizes. The Qwen3-4B student reaches 35.44% average success, exceeding the teacher's 30.28%. These results support local teacher supervision without altering the student's trajectory.

Reproducibility Statement

Algorithm 1 specifies the distillation procedure, and Section 4.6 describes the hardware allocation, replay buffer, and staleness filter. Section 5 identifies the teacher–student pairs, benchmarks, metrics, and sources of the agent results. Appendix A contains the supplementary mathematical reasoning experiments, and Appendix B gives the detailed mathematical and Qwen2.5 ALFWorld training and evaluation configurations. In the Qwen3 cross-benchmark comparison, the teacher and baseline rows are reproduced from TCOD, with the selected curriculum settings stated in Appendix B.3. The source package includes figure-generation scripts and aggregated records for the reported results.

AI USE STATEMENT

Generative AI tools were used to assist language editing, translation, figure-layout iteration, and LaTeX preparation. The authors determined the research claims, methods, experiments, and conclusions, and manually verified the final manuscript and reported results. The authors take responsibility for the final content, including all AI-assisted text and artifacts.

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. URL https://openreview.net/forum?id=3zKtaqxLhW.

AI-MO. AIME 2024, 2024.

Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. MathArena: Evaluating LLMs on uncontaminated math competitions, 2025. URL https://matharena.ai/.

Jeff Da, Clinton Wang, Xiang Deng, Yuntao Ma, Nikhil Barhate, and Sean Hendryx. Agent-RLVR: Training software engineering agents via guidance and environment rewards. *arXiv preprint arXiv:2506.11425*, 2025. URL https://arxiv.org/abs/2506.11425.

Yifan Ding, Xincheng Wei, Yoshua Y. Li, Ziheng Li, Yuquan Lu, Siyu Zhang, Dongsheng Ma, Rongxiang Weng, Xunliang Cai, and Yun Chen. SAF-OPD: Stable advantage fusion for on-policy distillation. *arXiv preprint arXiv:2607.29209*, 2026. URL https://arxiv.org/abs/2607.29209.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=5h0qf7IBZZ.

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, et al. OpenThoughts: Data recipes for reasoning models. *arXiv preprint arXiv:2506.04178*, 2025. URL https://arxiv.org/abs/2506.04178.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. DeepMath-103K: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *arXiv preprint arXiv:2504.11456*, 2025. URL https://arxiv.org/abs/2504.11456.

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

Li Jiang, Haoran Xu, Yichuan Ding, and Amy Zhang. Trajectory-refined distillation. *arXiv preprint arXiv:2606.08432*, 2026. URL https://arxiv.org/abs/2606.08432.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327, 2016.

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. URL https://arxiv.org/abs/2512.14895.

Gengsheng Li, Mao Zheng, Mingyang Song, Ruiqi Liu, Tianyu Yang, Jie Sun, Qiyong Zhong, Haiyun Guo, Junfeng Fang, Dan Zhang, and Jinqiao Wang. On-policy distillation with curriculum turn-level guidance for multi-turn agents. *arXiv preprint arXiv:2606.15912*, 2026. URL https://arxiv.org/abs/2606.15912.

Baohao Liao, Hanze Dong, Christof Monz, Xinxing Xu, Li Dong, and Furu Wei. Multi-turn on-policy distillation with prefix replay. *arXiv preprint arXiv:2607.04763*, 2026. URL https://arxiv.org/abs/2607.04763.

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

OpenCompass. AIME 2025, 2025.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Yang Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

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

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpaca: A strong, replicable instruction-following model. Stanford Center for Research on Foundation Models, 2023. URL https://crfm.stanford.edu/2023/03/13/alpaca.html.

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. URL https://arxiv.org/abs/2604.24005.

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. URL https://arxiv.org/abs/2505.09388.

Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. Learning beyond teacher: Generalized on-policy distillation with reward extrapolation. *arXiv preprint arXiv:2602.12125*, 2026. URL https://arxiv.org/abs/2602.12125.

Tianzhu Ye, Li Dong, Zewen Chi, Xun Wu, Shaohan Huang, and Furu Wei. Black-box on-policy distillation of large language models. *arXiv preprint arXiv:2511.10643*, 2025.

Yanfei Zhang, Xu Lin, and Chenglin Wu. StepOPSD: Step-aware online preference distillation for agent reinforcement learning. *arXiv preprint arXiv:2605.27140*, 2026. URL https://arxiv.org/abs/2605.27140.

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.

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. URL https://arxiv.org/abs/2605.07725.

A Mathematical Reasoning Experiments

Setup. We primarily use the Qwen3-4B non-thinking model (Yang et al., 2025). The student is initialized from Qwen3-4B non-thinking, and the domain teacher is obtained by applying reinforcement learning to the same base model on domain-specific data. Following G-OPD (Yang et al., 2026), we filter DeepMath (He et al., 2025) to retain 57,000 examples with difficulty level at least 6 and use this set as the mathematical RL data. We then apply Group Relative Policy Optimization (GRPO) (Shao et al., 2024) to obtain Qwen3-4B-Non-Thinking-RL-Math. The teacher receives reward 1.0 when the final answer is correct and 0.0 otherwise. Appendix B reports the complete teacher-training configuration.

For distillation, SOPD partitions responses at natural reasoning boundaries, whereas standard OPD optimizes sampled-token reverse KL. Both methods use the same filtered DeepMath prompt pool and sample one student response for each prompt. Detailed SOPD and OPD hyperparameters, including the step-partition rule, are given in Appendix B.

We evaluate mathematical reasoning on four competition-level benchmarks: AIME24 (AI-MO, 2024), AIME25 (OpenCompass, 2025), HMMT25-Feb, and HMMT25-Nov (Balunović et al., 2025). For every evaluation, we set temperature and top-p to 1.0 and the maximum generation length to 32,768 tokens. We sample 32 solutions for each problem and report mean accuracy on each benchmark. Answer correctness is determined by the rule-based `Math-Verify`[1] verifier.

Table 3: **Comparison of OPD and SOPD using the same-size teacher–student pair.** Teacher denotes Qwen3-4B-Non-Thinking-RL-Math, and Student denotes the initial Qwen3-4B non-thinking model. The first three rows are reference results reported by G-OPD (Yang et al., 2026). Values are mean accuracy (%); Avg_4 averages the four benchmarks, and bold marks the best student method in each column.

Method	AIME24	AIME25	HMMT25 (Feb.)	HMMT25 (Nov.)	Avg_4
Teacher	58.0	54.6	32.5	38.9	46.0
+ continued RL (100 updates)	60.9	55.6	32.8	38.4	46.9
Student	21.5	21.9	10.0	8.0	15.4
OPD	61.9	57.0	32.5	39.6	47.7
SOPD	**71.8**	**67.4**	**38.9**	**52.7**	**57.7**

The SOPD run leads on all four benchmarks. SOPD raises mean accuracy over OPD by 9.9 points on AIME24, 10.4 points on AIME25, 6.4 points on HMMT25-Feb, and 13.1 points on HMMT25-Nov. Its four-benchmark average reaches 57.7, compared with 47.7 for OPD. Each cell aggregates 32 decoding samples for each of 30 problems, providing a comprehensive evaluation of the final checkpoint.

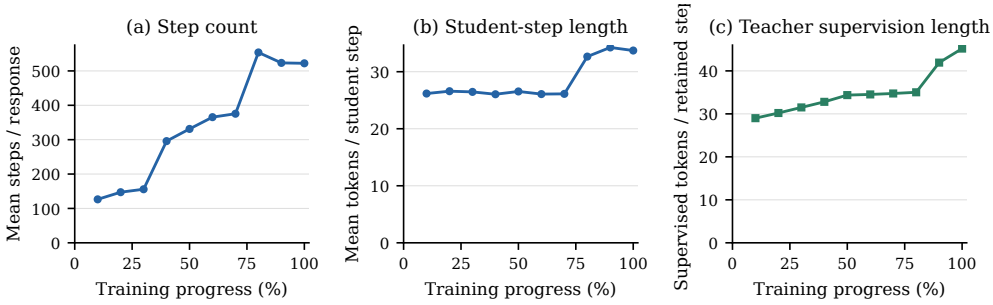

Figure 4: **SOPD step structure over mathematical training.** (a) Mean natural steps per AIME24 evaluation response; (b) mean student tokens per natural step; and (c) mean supervised teacher tokens per retained step during training. All boundaries use the semantic-step rule and 1,024-token safety cap.

[1] `https://github.com/huggingface/Math-Verify`

The learned response structure changes during training. Figure 4 tracks the quantities that define SOPD's supervision units. As training progresses, the number of natural reasoning steps generated by the student increases substantially, while the length of each step remains broadly stable. This indicates that the model extends its solution process mainly by organizing more structured reasoning steps. Mean teacher supervision grows from 29.0 tokens at update 1 to 35.0 at update 8, then increases to 41.9 and 45.2 at the final two updates. Together, the three statistics show that later checkpoints both organize more reasoning steps and receive richer teacher targets at each local state.

B Training and Evaluation Details

B.1 Mathematical Reasoning

The mathematical runs use Qwen3-4B non-thinking as the base student and Qwen3-4B-Non-Thinking-RL-Math-Step500 as the teacher. We construct the teacher following the mathematical GRPO configuration of G-OPD (Yang et al., 2026). Specifically, we retain 57,000 DeepMath examples with difficulty level at least 6, assign reward 1.0 when the final answer is correct and 0.0 otherwise, and train for 500 optimizer updates. Table 4 lists the complete teacher-training configuration used in our experiments.

Table 4: GRPO hyperparameters used to train the mathematical domain teacher, following G-OPD (Yang et al., 2026).

Hyperparameter	Value
Training batch size	128
Micro-batch size	128
Rollouts per prompt	8
Maximum prompt length	2,048
Maximum response length	16,384
Temperature	1.0
Top-p	1.0
Learning rate	1×10^{-6}
Optimizer updates	500
KL coefficient	0.0

SOPD and OPD then consume the same fixed order of prompts from the filtered DeepMath pool. Each optimizer update uses 1,024 prompts and one student rollout per prompt. Both methods run for ten updates, so each processes 10,240 prompt instances and 10,240 student rollouts. The maximum prompt and response lengths are 2,048 and 32,768 tokens, respectively; student rollouts use temperature 1.0 and top-p 1.0. SOPD uses learning rate 2×10^{-6} and generates each teacher step greedily at temperature 0. Standard OPD uses learning rate 10^{-5} and sampled-token reverse KL.

For SOPD, a blank-line paragraph boundary or an explicit reasoning heading such as "Step" or "Solution" terminates a natural step. If no such boundary appears, a 1,024-token safety cap closes the span. Every retained student prefix receives one independently generated teacher step, and the loss is averaged first over tokens within each target and then over retained steps.

The terminal evaluation sends 32 independent generations for each of 30 problems in AIME24, AIME25, HMMT25-Feb, and HMMT25-Nov. Each method therefore contributes 960 responses per benchmark. Evaluation uses temperature 1.0, top-p 1.0, and a 32,768-token response cap. Every response is a separate request with seed $42 + 32i + j$ for problem i and sample j. The evaluator extracts the final boxed answer and applies rule-based symbolic verification with `Math-Verify`. The result extractor checks all numerators and denominators against the durable summaries.

B.2 ALFWorld

Table 1 evaluates Qwen2.5-3B and Qwen2.5-7B students with the same domain-trained 7B teacher. The configuration below describes the 3B SOPD run used for the training-dynamics figures.

We use the Qwen2.5-3B-Instruct student and 'langfeng01/GiGPO-Qwen2.5-7B-Instruct-ALFWorld' teacher from the TCOD model pairing. The Explorer samples 16 tasks with one rollout each. The Trainer consumes 64 experiences per batch with learning rate 10^{-6}, temperature 1.0, staleness two, a 512-token training response cap, a 10,240-token prompt cap, and at most 30 environment turns. The released asynchronous launcher lets Explorer and Trainer counters advance independently.

Full evaluation covers Valid Seen-140, Valid Unseen-134, and Hard-121. It samples at temperature 0.4 with a 4,096-token response cap. We report the task-level success mean and mean environment rounds.

B.3 QWEN3 CROSS-BENCHMARK BASELINES

Table 2 reports the TCOD-B2F and TCOD-F2B configurations with the best reported average performance across WebShop, ALFWorld, and ScienceWorld for each student size (Wang et al., 2026). The selected B2F/F2B curriculum growth rates are $\eta = 4/6$ for the Qwen3-1.7B student and $\eta = 2/2$ for the Qwen3-4B student.