OPAD: ON-POLICY ACTION DISTILLATION FOR LARGE LANGUAGE MODEL AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation trains language model agents with teacher guidance at student-visited states, but token-level matching can entangle decision learning with imitation of the teacher's text. The environment responds only to executed actions, making them a natural unit of knowledge transfer. We introduce On-Policy Action Distillation (OPAD), which aligns teacher and student preferences over executable actions. We show that the student's success gap to a reference that keeps the student's reasoning but executes the teacher's actions is bounded by missing or mismatched actions at student-visited states. OPAD builds teacher targets from sampled responses, scores candidate actions after the student's reasoning, concentrates supervision at local peaks of action-span entropy, and weights it by trajectory outcomes and teacher support to balance success consolidation and failure correction. The action interface supports cross-tokenizer and black-box teachers without vocabulary alignment or teacher logits. On ALFWorld, WebShop, and ScienceWorld, OPAD outperforms token-level and black-box distillation baselines in nearly all comparisons across student sizes, model families, and teacher access settings, with several students surpassing their teachers.

1 INTRODUCTION

Distilling the decision-making capabilities of strong large language model (LLM) agents into smaller students offers a way to reduce the cost of interactive task execution. However, learning from teacher-generated trajectories alone leaves students vulnerable to compounding errors, as their own mistakes can lead to states poorly covered by those trajectories (Li et al., 2026a). To address this limitation of off-policy distillation, recent work has turned to on-policy distillation (OPD), which provides teacher guidance at states visited by the student's current policy (Wang et al., 2026; Li et al., 2026b). However, representative agent OPD methods match token-level distributions, potentially coupling decision transfer with imitation of the teacher's textual expression. As illustrated in Figure 1(a–b), different reasoning expressions can yield the same executable action, while changing an action argument can alter its environmental consequences even when the reasoning text is unchanged. Moreover, direct token-probability matching relies on tokenizer-specific representations and access to teacher probabilities, requiring additional mechanisms across tokenizers or with generation-only black-box teachers, as shown in Figure 1(c). We therefore shift the unit of distillation from tokens to complete executable actions, aligning teacher

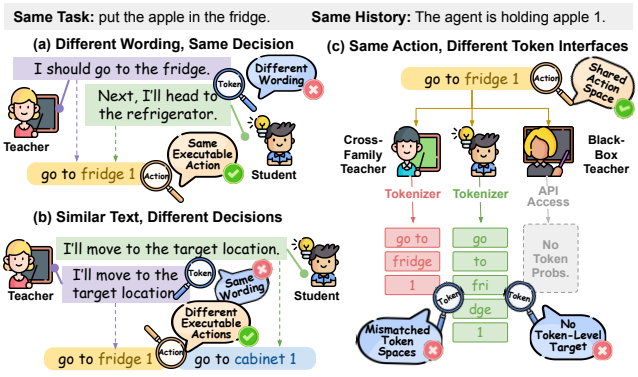

Figure 1: Limitations of token-level distillation and how a shared action interface addresses them. Magnifying glasses labeled Token and Action represent token-level and action-level views. (a–b) Textual differences need not reflect decision differences. (c) Shared actions provide a common interface across tokenizers and teacher access settings.

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.

and student preferences over decisions that drive environment transitions through a shared action interface, without vocabulary alignment or teacher logits. Section 3.1 theoretically shows that action mismatch at student-visited states bounds the student's success gap to a teacher-action reference.

Turning this action-level view into effective supervision requires addressing three challenges. First, distributions over complete actions cannot be read directly like next-token distributions, requiring teacher targets to be estimated and differentiable student distributions to be constructed. Second, supervising every step of a long interaction is costly, while the value of teacher guidance varies across decisions, making supervision selection essential. Finally, successful trajectories may contain inefficient actions and failed ones reasonable decisions, so trajectory outcomes alone cannot determine action-level supervision weights and must be combined with teacher assessments of individual decisions to coordinate success consolidation and failure correction.

To address these challenges, we introduce On-Policy Action Distillation (OPAD), a framework for LLM agents that uses complete executable actions as the unit of knowledge transfer. By aligning teacher and student action distributions at student-visited states, OPAD ties supervision directly to decisions that drive environment transitions while reducing reliance on imitation of the teacher's textual expression. For supervision selection, OPAD prioritizes local peaks in student action-span entropy to concentrate teacher guidance on uncertain decisions. To consolidate teacher-endorsed decisions in successful trajectories and focus correction on disagreements in failed ones, OPAD assigns larger selection budgets to failed trajectories and combines trajectory outcomes with teacher support for executed actions to weight selected decisions. OPAD also enables distillation across model families and from generation-only black-box teachers.

Our main contributions are as follows:

- We introduce OPAD, an on-policy distillation framework for LLM agents that transfers decision-making capabilities through complete executable actions, with support for cross-family and black-box distillation.
- We develop uncertainty-guided supervision selection and outcome-aware supervision allocation to coordinate success consolidation with failure correction.
- Extensive experiments on interactive agent benchmarks demonstrate that OPAD improves task performance and enables effective transfer across student sizes, model generations and families, and teacher access settings, validating complete executable actions as an effective unit for transferring agent decision-making capabilities.

2 Related Work

On-policy distillation for language model agents. Recent work adapts OPD to multi-turn agents. With external teachers, TCOD (Wang et al., 2026) expands the student's interaction horizon through a temporal curriculum, Guided-OPD (Li et al., 2026b) intersperses teacher-generated turns and gradually reduces intervention, and LSM (Fu et al., 2026) concentrates distribution matching on teacher-selected local token supports. Another line uses on-policy self-distillation (OPSD; Zhao et al., 2026) to refine credit assignment in Group Relative Policy Optimization (GRPO; Shao et al., 2024): StepOPSD (Zhang et al., 2026) shapes advantages with teacher–student token log-probability differences, and SERL (Li et al., 2026c) combines advantage reweighting with token-level distillation over action spans. These methods retain token-level distillation signals, whereas OPAD aligns teacher and student distributions over complete executable actions at student-visited states.

Cross-family and black-box distillation. Token-level OPD typically requires compatible tokenizers and teacher probabilities (Zhou et al., 2026). GOLD (Patiño et al., 2025) aligns sequences and vocabularies across tokenizers, SimCT (Sun et al., 2026) matches short continuations realizable by both tokenizers, and Cross-Tokenizer OPD (Niu et al., 2026) distributes the probabilities of identically decoded teacher chunks across student tokens; all still require teacher probabilities. Other methods learn from teacher outputs alone: DAgger (Ross et al., 2011; Li et al., 2026a) aggregates expert labels at learner-visited states, OmniOPD (Zhou et al., 2026) uses similarities between teacher samples and student text chunks, and GAD (Ye et al., 2025) and ROPD (Fang et al., 2026) derive rewards from teacher text through discriminator scores and contrastive rubrics, respectively. OPAD instead builds action-level distributional supervision, using executable actions as a shared interface without vocabulary alignment or teacher logits.

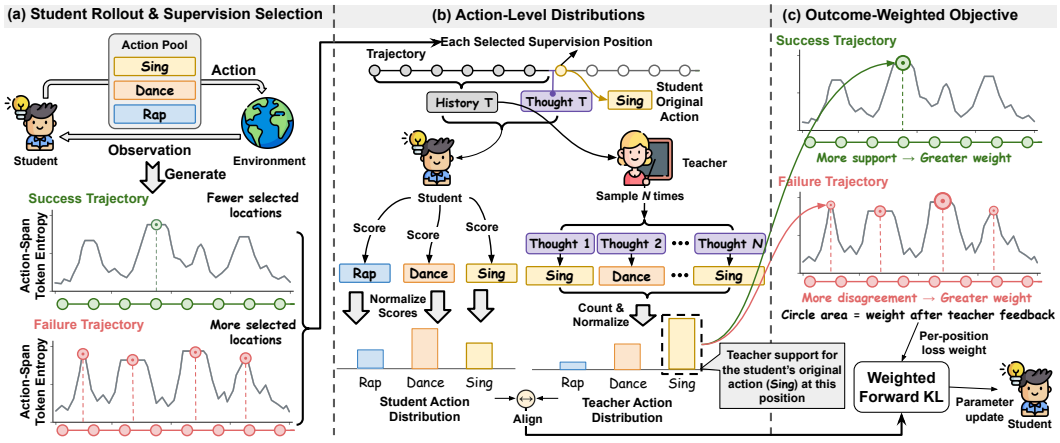

Figure 2: **Overview of OPAD. (a)** Student–environment interaction yields successful or failed trajectories; local peaks of mean action-span token entropy are selected for supervision, with smaller budgets for successful trajectories. **(b)** At each selected position, the teacher distribution is built from smoothed valid-action counts over N sampled responses, and the student scores the shared candidate actions given the same history and original thought. The highlighted teacher probability marks support for the student's executed action. **(c)** Weights grow with teacher support on successful trajectories and with disagreement on failed ones, promoting consolidation and correction; the student minimizes the weighted forward KL between the two action distributions to update its policy.

3 OPAD: On-Policy Action Distillation

We now present OPAD, an OPD framework that aligns teacher and student preferences over complete executable actions along student-generated trajectories. As illustrated in Figure 2, the framework prioritizes local peaks in action-span entropy to select supervision positions, constructs teacher and student action distributions at these positions, and updates the student through forward KL weighted by trajectory outcomes and teacher support for the student's originally executed actions.

3.1 Formulating On-Policy Distillation for LLM Agents

We first formalize OPD for LLM agents. Let π_ϕ and π_θ denote the teacher and student generation policies, respectively, and $\widetilde{\pi}_\theta$ the student's sampling distribution after decoding adjustments. For task x, let h_t denote the interaction history and o_t the environment observation at step t, with $h_1 = (x, o_1)$. The student samples a response $y_t \sim \widetilde{\pi}_\theta(\cdot \mid h_t)$ containing a reasoning prefix r_t followed by an executable action a_t. Execution yields o_{t+1} and updates $h_{t+1} = h_t \oplus (y_t, o_{t+1})$, where $\oplus$ denotes concatenation. This produces a trajectory $\tau = (x, o_1, y_1, o_2, \ldots, y_{T_\tau}, o_{T_\tau+1})$ of T_τ interaction steps, with outcome $R(\tau) \in \{0, 1\}$, where 1 indicates success.

At iteration k, OPD collects trajectories through environment interaction using $\widetilde{\pi}_{\theta_k}$ and applies teacher supervision at student-visited histories, yielding the objective

$$\mathcal{L}_{\mathrm{OPD}}^{(k)}(\theta) = \mathbb{E}_{\tau \sim \widetilde{\pi}_{\theta_k}} \left[\sum_{u \in \mathcal{U}(\tau)} \ell(\pi_{\phi,u}, \pi_{\theta,u}) \right], \tag{1}$$

where $\mathcal{U}(\tau)$ indexes supervision positions, $\pi_{\phi,u}$ and $\pi_{\theta,u}$ are the teacher target and student prediction distributions at u, respectively, and $\ell(\cdot, \cdot)$ denotes the distillation loss. Token-level supervision uses $u = (t, j)$, with j indexing tokens within response y_t, and matches next-token distributions. OPAD instead matches local distributions over complete executable actions at decision positions $u = t$.

Why distill actions? The environment responds only to the actions parsed from responses, so text affects success only through the actions it produces. Theorem 1 makes this precise by comparing the student with a *shadow student* that, at every history, generates the student's own reasoning prefix but then takes an action drawn from the teacher's action distribution $\mu(\cdot \mid h_t)$, the distribution of valid actions in the teacher's responses at h_t, independently of that reasoning. Let $J(\pi) = \mathbb{E}_{\tau \sim \pi}[R(\tau)]$

denote the success rate, and let $\kappa_\theta(\cdot \mid h_t, r_t)$ be the distribution of the action that the student executes after its reasoning prefix r_t.

Theorem 1 (Action-level performance bound; informal)**.** *Under the conditions in Appendix D, for any student, the success rates of the student and of its shadow student differ by at most*

$$\sum_t \mathbb{E}_{h_t}\Big[\big(1-\omega_\theta(h_t)\big) + \mathbb{E}_{r_t}\,\mathrm{TV}\big(\kappa_\theta(\cdot \mid h_t, r_t),\ \mu(\cdot \mid h_t)\big)\Big],$$

where h_t ranges over the histories that the student visits, $1-\omega_\theta(h_t)$ is the probability that its response does not open an action span, r_t is its sampled reasoning prefix given that it does, and TV *denotes total variation distance; the second term is zero when $\omega_\theta(h_t) = 0$.*

Theorem 1 supports on-policy action distillation. The success gap to the shadow student is controlled by how closely the student's action choices match the teacher's at the states the student visits, which are where on-policy distillation supervises. The bound never compares reasoning: the student may reason and phrase differently from the teacher without enlarging the bound as long as the chosen actions agree, whereas token-level objectives defined over full responses also require imitating the teacher's reasoning text. Appendix D gives the formal statement, conditions, and proof.

3.2 Constructing Action-Level Distributions

Action-level distillation requires constructing preferences over complete executable actions from token-generating policies. At a selected history h_t, we build a candidate set $\mathcal{C}_t$, starting from interface-provided actions and the student's valid action a_t from the current trajectory. We construct the teacher target q_t and student prediction distribution $p_{\theta,t}$ from the generation policies π_ϕ and π_θ, respectively, both over the final candidate set $\mathcal{C}_t$.

Student action distributions. We retain the student's generated prefix r_t, including its reasoning and the opening action delimiter, and score each candidate action as a continuation. Let $c(a)$ denote the action continuation including its closing delimiter, with its token sequence and length L_a determined by the student's tokenizer in the scoring context. To reduce the influence of action length, we average token log-likelihoods and normalize over the candidate set $\mathcal{C}_t$:

$$s_{\theta,t}(a) = \frac{1}{L_a}\sum_{j=1}^{L_a} \log \pi_\theta(c_j(a) \mid h_t, r_t, c_{<j}(a))\,, \qquad p_{\theta,t}(a) = \frac{\exp(s_{\theta,t}(a)/T_s)}{\sum_{a'\in\mathcal{C}_t}\exp(s_{\theta,t}(a')/T_s)}, \quad (2)$$

where $T_s > 0$ controls distribution sharpness. This local scoring distribution enables direct backpropagation of the distillation loss to the student parameters.

Teacher action targets. If the teacher used the same candidate-scoring procedure as the student, its target would reflect relative continuation scores rather than action-selection frequencies under its own generation process. To provide action-level guidance at student-visited states, we let the teacher reason and select an action in each of N independently sampled next-step responses at h_t, then aggregate its choices across these reasoning paths (Wang et al., 2022b). Parsed actions supplement $\mathcal{C}_t$ after available interface checks, canonicalization, and deduplication. Let $n_t(a)$ denote the number of occurrences of action a among the teacher responses, with $N_t = \sum_{a\in\mathcal{C}_t} n_t(a)$ denoting the total number of valid samples. The empirical teacher target is

$$q_t(a) = \frac{n_t(a)+\epsilon}{N_t + \epsilon|\mathcal{C}_t|}, \qquad a \in \mathcal{C}_t, \quad (3)$$

where $\epsilon > 0$ smooths the distribution by assigning positive probability to unsampled candidates. Positions without valid teacher samples are excluded from distillation.

3.3 Entropy-Guided Supervision Selection

Applying the action-level targets above at every step makes distillation cost grow with trajectory length. Recent studies link high-entropy tokens to branching reasoning paths and show that concentrating training on these positions can preserve or improve reasoning performance (Wang et al., 2025; Ni et al., 2026), suggesting that learning value varies across positions. For agents, action-span tokens jointly specify executable actions whose execution shapes subsequent interactions. We therefore use action-span token uncertainty to select supervision positions $\mathcal{U}(\tau)$.

The student first collects a complete trajectory τ using the sampling policy $\widetilde{\pi}_{\theta_k}$. For steps with a valid action and a nonempty action span, let $\mathcal{J}_t$ index the action-span tokens in response y_t. Using token probabilities already computed during generation, we define

$$e_t = \frac{1}{|\mathcal{J}_t| \log |\mathcal{V}_\theta|} \sum_{j \in \mathcal{J}_t} H(\widetilde{\pi}_{\theta_k}(\cdot \mid h_t, y_{t,<j})), \tag{4}$$

where $\mathcal{V}_\theta$ denotes the student's vocabulary and $H(\pi) = -\sum_v \pi(v) \log \pi(v)$ is Shannon entropy (Shannon, 1948). Averaging over the span reduces the influence of action length.

Ranking by entropy alone can concentrate supervision within a single persistently uncertain segment. We therefore prioritize local peaks among eligible steps along the trajectory, representing flat peaks by their midpoints. To retain broad peaks while reducing the priority of minor fluctuations, we use peak prominence (Virtanen et al., 2020):

$$\mathrm{Prom}_\tau(t) = e_t - \max\{\underline{e}_t^-, \underline{e}_t^+\}, \tag{5}$$

where $\underline{e}_t^-$ and $\underline{e}_t^+$ are the minimum entropy values found on each side up to a strictly higher value or the sequence boundary; boundary peaks use only the available side. Peaks satisfying $\mathrm{Prom}_\tau(t) \geq \delta_{\mathrm{peak}}$ form the priority set $\mathcal{P}_\tau$, where $\delta_{\mathrm{peak}} > 0$ is the minimum prominence threshold.

We rank priority peaks by entropy and initialize $\mathcal{U}(\tau) \leftarrow \mathrm{Top}_K(\mathcal{P}_\tau)$, where $\mathrm{Top}_K(\cdot)$ selects up to K positions in descending order and K is the per-trajectory selection limit. If fewer than K positions are selected, we add the highest-entropy remaining eligible positions up to this limit, allowing multiple decisions within the same high-entropy segment to receive supervision. We query the teacher only at h_t with $t \in \mathcal{U}(\tau)$ for the action-distillation objective in Eq. (7) below.

3.4 Outcome-Weighted Distillation Objective

Prior analysis of OPD on reasoning tasks suggests that teacher guidance offers greater corrective value on incorrect trajectories than on correct ones (Armandpour et al., 2026). Motivated by this finding, we use trajectory outcomes to distinguish between consolidating successful behavior and correcting policy weaknesses, adapting both the selection limit K and supervision weights accordingly. We set $K = K_{\mathrm{succ}}$ when $R(\tau) = 1$ and $K = K_{\mathrm{fail}}$ when $R(\tau) = 0$, with $K_{\mathrm{fail}} > K_{\mathrm{succ}}$.

For each selected position $t \in \mathcal{U}(\tau)$ with $N_t > 0$, let $v_t = q_t(a_t)$ denote teacher support for the executed action. On successful trajectories with $R(\tau) = 1$, task completion provides evidence of effective behavior, while teacher endorsement further supports consolidating the decision. We therefore set $w_t = v_t$, assigning greater supervision weight to teacher-supported decisions.

On failed trajectories with $R(\tau) = 0$, teacher disagreement provides direction for correcting policy weaknesses. We measure disagreement as $d_t = 1 - v_t$ and define

$$w_t = \left(\max_{t'} d_{t'}\right) \frac{m_\tau d_t}{\sum_{t'} d_{t'}}, \qquad \sum_{t'} d_{t'} > 0, \tag{6}$$

where t' ranges over positions satisfying $t' \in \mathcal{U}(\tau)$ and $N_{t'} > 0$ within the same trajectory, and m_τ is their number. This weighting concentrates corrective supervision on stronger teacher disagreements while retaining their absolute magnitude, preventing uniformly weak disagreement from being normalized into strong supervision. If $\sum_{t'} d_{t'} = 0$, we set $w_t = 0$, as the teacher provides no disagreement signal to guide correction at these positions.

We weight the local distillation losses in Eq. (1) by w_t. Since multiple actions may be reasonable at each selected state and the teacher's full action distribution is difficult to characterize precisely, we adopt forward Kullback–Leibler (KL) divergence (Kullback & Leibler, 1951) to encourage coverage of teacher-supported alternatives. The resulting action-distillation objective is

$$\mathcal{L}_{\mathrm{act}}^{(k)}(\theta) = \mathbb{E}_{\tau \sim \widetilde{\pi}_{\theta_k}} \left[\sum_{t \in \mathcal{U}(\tau) \wedge N_t > 0} w_t D_{\mathrm{KL}}(q_t \,\|\, p_{\theta,t}) \right], \tag{7}$$

where $D_{\mathrm{KL}}(q_t \| p_{\theta,t}) = \sum_{a \in \mathcal{C}_t} q_t(a) \log \frac{q_t(a)}{p_{\theta,t}(a)}$. Our loss ablations further support this choice, with comparisons with alternative objectives and detailed discussion provided in Section 4.5.

4 EXPERIMENTS

To examine how shifting the unit of on-policy distillation from tokens to complete executable actions affects agent learning, we evaluate OPAD through comparative experiments, ablation studies, and training analyses. Our study addresses the following research questions:

(Q1) Does action-level distillation improve agent task performance compared with token-level OPD, and how does its effectiveness vary with student size?

(Q2) How effectively does the shared action interface support capability transfer across model generations and families, as well as from generation-only black-box teachers?

(Q3) How do the distillation objective, action-distribution construction, supervision selection, and outcome-aware supervision affect task performance?

4.1 EXPERIMENTAL SETUP

Tasks and metrics. We evaluate on three interactive environments: ALFWorld (Shridhar et al., 2020) for household tasks, ScienceWorld (Wang et al., 2022a) for scientific experiments, and WebShop (Yao et al., 2022) for goal-directed shopping. On ALFWorld, we report success rate (SR) on Seen, Unseen, and the Hard stress-test split of TCOD (Wang et al., 2026). We report Task Score and strict success rate (Strict SR) on WebShop, Final Score and Strict SR on ScienceWorld, and mean interaction rounds (Rounds) over all evaluation episodes.

Baselines. For Q1, we compare with two groups: methods without an external teacher, namely offline supervised fine-tuning (SFT), GRPO (Shao et al., 2024), and its on-policy self-distillation variants StepOPSD (Zhang et al., 2026) and SERL (Li et al., 2026c); and token-level OPD methods with an external teacher, namely Vanilla OPD adapted to agent interaction, TCOD (Wang et al., 2026), LSM (Fu et al., 2026), and Guided-OPD (Li et al., 2026b). For Q2, we compare with On-policy SeqKD, which trains on teacher responses at student-visited states as in sequence-level distillation (Kim & Rush, 2016), DAgger (Ross et al., 2011; Li et al., 2026a), OmniOPD (Zhou et al., 2026), GAD (Ye et al., 2025), and ROPD (Fang et al., 2026), adding GOLD (Patiño et al., 2025), SimCT (Sun et al., 2026), and Cross-Tokenizer OPD (CT-OPD; Niu et al., 2026) for cross-generation and cross-family students. Teacher and zero-shot student results serve as references, and Appendix B gives further details.

4.2 EFFECTIVENESS OF ACTION-LEVEL DISTILLATION

To answer Q1, we distill from a Qwen2.5-7B-Instruct (Qwen et al., 2025) teacher trained on ALFWorld with GiGPO (Feng et al., 2025) into a Qwen2.5-3B-Instruct student. We then vary the student size to 1.5B and 7B while keeping the teacher fixed.

Table 1: ALFWorld SR and interaction rounds of OPAD and baselines. Values are means ± standard deviations. Bold/underline mark the best/second-best trained-student results. Green marks results better than the teacher for methods using an external teacher.

Method	Seen		Unseen		Hard	
	SR (%) ↑	Rounds ↓	SR (%) ↑	Rounds ↓	SR (%) ↑	Rounds ↓
Teacher: Qwen2.5-7B-Instruct						
GiGPO	$87.86_{\pm 2.76}$	$9.97_{\pm 0.67}$	$85.82_{\pm 2.72}$	$12.60_{\pm 0.69}$	$16.69_{\pm 3.31}$	$27.11_{\pm 0.47}$
Student: Qwen2.5-3B-Instruct						
Zero-Shot	$7.00_{\pm 0.32}$	$28.50_{\pm 0.28}$	$6.12_{\pm 2.82}$	$28.42_{\pm 0.47}$	$2.15_{\pm 1.81}$	$29.26_{\pm 0.35}$
SFT	$35.29_{\pm 4.04}$	$21.95_{\pm 0.98}$	$31.64_{\pm 3.63}$	$23.58_{\pm 0.92}$	$5.79_{\pm 2.07}$	$29.00_{\pm 0.29}$
GRPO	$\underline{84.43}_{\pm 1.92}$	$13.10_{\pm 0.55}$	$77.16_{\pm 1.12}$	$14.96_{\pm 0.09}$	$14.55_{\pm 2.11}$	$28.31_{\pm 0.29}$
StepOPSD	$76.57_{\pm 3.58}$	$12.55_{\pm 0.87}$	$71.49_{\pm 3.52}$	$15.51_{\pm 0.89}$	$16.36_{\pm 3.37}$	$26.98_{\pm 0.48}$
SERL	$79.71_{\pm 3.39}$	$11.81_{\pm 0.82}$	$74.63_{\pm 3.39}$	$14.86_{\pm 0.86}$	$\underline{18.68}_{\pm 2.52}$	$\underline{26.61}_{\pm 0.54}$
Vanilla OPD	$51.71_{\pm 2.18}$	$19.80_{\pm 0.36}$	$46.57_{\pm 4.56}$	$20.74_{\pm 0.38}$	$11.24_{\pm 1.72}$	$28.52_{\pm 0.07}$
TCOD	$80.00_{\pm 3.17}$	$12.28_{\pm 0.77}$	$78.36_{\pm 3.19}$	$14.05_{\pm 0.81}$	$12.73_{\pm 3.06}$	$27.61_{\pm 0.33}$
LSM	$81.43_{\pm 3.29}$	$11.44_{\pm 0.80}$	$77.01_{\pm 3.28}$	$14.38_{\pm 0.83}$	$14.21_{\pm 3.19}$	$27.36_{\pm 0.45}$
Guided-OPD	$83.86_{\pm 3.11}$	$\underline{10.89}_{\pm 0.76}$	$\underline{80.30}_{\pm 3.10}$	$\underline{13.73}_{\pm 0.79}$	$17.19_{\pm 2.42}$	$26.86_{\pm 0.29}$
OPAD	$\mathbf{89.29}_{\pm 0.35}$	$\mathbf{9.92}_{\pm 0.05}$	$\mathbf{86.57}_{\pm 0.76}$	$\mathbf{12.02}_{\pm 0.18}$	$\mathbf{20.83}_{\pm 1.69}$	$\mathbf{26.43}_{\pm 0.18}$

Main results. Table 1 shows that OPAD achieves the highest SR among all student methods on every split. Compared with Guided-OPD, the strongest token-level OPD baseline, it improves SR by 5.43, 6.27, and 3.64 points on three splits, respectively. These gains support action-level distillation, which aligns preferences over complete actions and their parameters to supervise decisions that determine environment transitions without additionally matching their textual expression. The advan-

tages on Unseen and Hard further support its effectiveness in unfamiliar and challenging interactions. Gains over GRPO and the OPSD-enhanced GRPO methods StepOPSD and SERL also highlight the value of external teacher guidance, which supplies explicit action preferences beyond task rewards and self-distilled feedback. Higher SR together with the lowest interaction rounds indicates better task completion with lower average interaction cost. The 3B student exceeds the teacher's SR on every split, showing that action-level guidance need not cap student performance at the teacher's level. This flexibility stems from combining local teacher guidance with outcome-aware weighting to consolidate teacher-supported successful decisions and correct disagreements in failed trajectories, allowing policy improvement without reproducing the teacher's full behavior. On ALFWorld Seen, OPAD first reaches 70% SR after 15.23 GPU-hours, compared with 41.34–53.67 GPU-hours for the six measured baselines; Appendix C.1 gives the full comparison and accounting scope.

Effect of student size. Holding the teacher fixed, we vary student size to test whether the benefits of action-level supervision persist under different capacity constraints. Figure 3 omits the substantially weaker Zero-Shot, SFT, and Vanilla OPD for clarity. OPAD leads across all evaluated sizes and splits. Its advantage holds even for the smallest student and widens on Hard for the 7B student, which matches the teacher in size. These results show that action-level distillation accommodates different teacher–student capability gaps and benefits students with varying capabilities, transferring domain expertise to smaller agents while further improving stronger students.

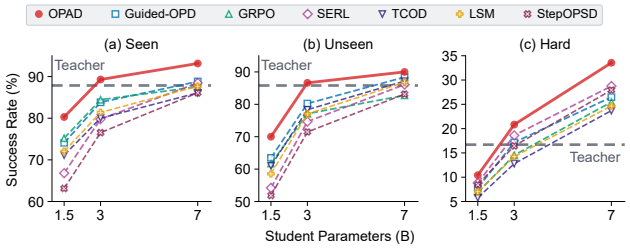

Figure 3: SR of OPAD and baseline methods across student sizes on ALFWorld, with the teacher held fixed.

Beyond the expert teacher above, we also evaluate two general-purpose teachers. The open-weight xLAM-2-32b-fc-r (Prabhakar et al., 2025) is based on Qwen2.5-32B-Instruct and fine-tuned for agentic tasks, with results in Appendix C.2. The black-box GPT-5.6 Terra is evaluated in the black-box distillation experiments in Section 4.4.

4.3 Cross-Generation and Cross-Family Distillation

To answer Q2, we test transfer to students of broadly comparable size from newer generations and different families while retaining the same teacher. We use Qwen3-4B (Yang et al., 2025) and Qwen3.5-4B (Team, 2026) for cross-generation transfer, and Llama-3.2-3B-Instruct (Grattafiori et al., 2024) and Phi-4-mini-instruct (Abouelenin et al., 2025) for cross-family transfer.

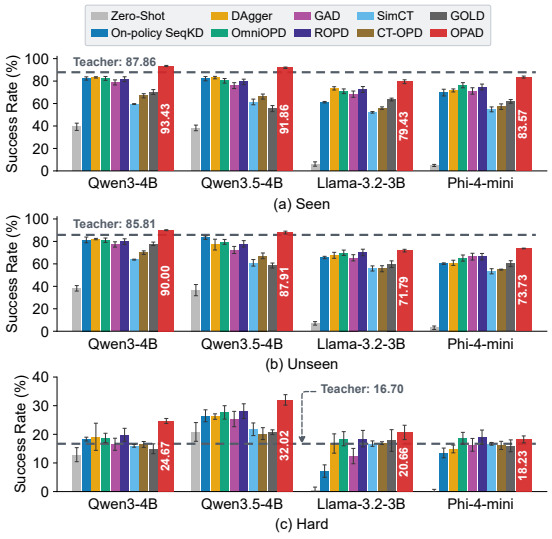

Figure 4: Cross-generation and cross-family distillation on ALFWorld. Bars show SR, error bars show std, and dashed lines indicate the teacher's SR.

Figure 4 shows that OPAD achieves the highest SR in nearly all comparisons across students and splits. Both Qwen students distilled with OPAD exceed the teacher on every split, demonstrating that action-level supervision can transfer an earlier-generation expert's domain knowledge to newer students. The gains on Llama and Phi further establish its effectiveness beyond the teacher's model family. The response-imitation baselines On-policy SeqKD and DAgger are generally more competitive for the later Qwen generations than for Llama and Phi, suggesting that transfer within a

Table 2: Task performance and teacher API cost of OPAD-API and baselines with a black-box teacher. Costs are per optimizer update, averaged across the three environments.

Method	ALFWorld SR (%) ↑			WebShop ↑		ScienceWorld ↑		API Cost ↓ ($/update)
	Seen	Unseen	Hard	Task Score	Strict SR (%)	Final Score	Strict SR (%)	
Teacher: GPT-5.6 Terra								
Zero-Shot	$87.57_{\pm 3.53}$	$89.55_{\pm 2.64}$	$50.41_{\pm 4.55}$	$62.76_{\pm 2.61}$	$34.00_{\pm 3.74}$	$69.38_{\pm 1.15}$	$60.78_{\pm 1.35}$	—
Student: Qwen2.5-3B-Instruct								
Zero-Shot	$7.00_{\pm 0.32}$	$6.16_{\pm 2.82}$	$2.15_{\pm 1.81}$	$11.89_{\pm 2.53}$	$2.00_{\pm 1.41}$	$9.94_{\pm 0.50}$	$1.97_{\pm 0.43}$	—
On-policy SeqKD	$52.43_{\pm 4.45}$	$46.57_{\pm 1.87}$	$9.26_{\pm 2.29}$	$46.48_{\pm 3.18}$	$19.20_{\pm 3.42}$	$43.23_{\pm 2.17}$	$27.26_{\pm 2.46}$	**0.087**
DAgger	$65.00_{\pm 1.34}$	$63.43_{\pm 2.51}$	$25.62_{\pm 3.09}$	$\underline{53.60}_{\pm 2.87}$	$\underline{26.60}_{\pm 3.13}$	$50.95_{\pm 1.94}$	$36.67_{\pm 2.21}$	$\underline{0.088}$
OmniOPD	$70.14_{\pm 3.27}$	$69.70_{\pm 3.24}$	$\underline{28.60}_{\pm 3.21}$	$52.08_{\pm 3.06}$	$24.00_{\pm 3.39}$	$53.28_{\pm 2.09}$	$40.90_{\pm 2.32}$	0.486
GAD	$64.29_{\pm 3.62}$	$61.34_{\pm 3.79}$	$21.49_{\pm 3.18}$	$51.26_{\pm 3.12}$	$24.20_{\pm 3.19}$	$49.17_{\pm 2.28}$	$34.56_{\pm 2.67}$	0.107
ROPD	$\underline{75.71}_{\pm 3.24}$	$\underline{74.18}_{\pm 3.51}$	$27.11_{\pm 3.16}$	$53.42_{\pm 2.94}$	$\underline{26.60}_{\pm 3.21}$	$\underline{54.31}_{\pm 2.16}$	$\underline{43.32}_{\pm 2.54}$	0.349
OPAD-API	**83.29**$_{\pm 3.76}$	**82.69**$_{\pm 4.32}$	**34.21**$_{\pm 4.24}$	**58.18**$_{\pm 4.72}$	**30.80**$_{\pm 3.96}$	**57.49**$_{\pm 3.83}$	**46.67**$_{\pm 4.05}$	0.117

family is relatively easier than transfer across families. OPAD aligns relative preferences over complete actions, whose operations and arguments have the same meaning across models even when their tokenization and reasoning differ. This common decision space allows students to learn domain knowledge from teachers of different generations or families using their own tokenizers and reasoning prefixes. Its consistent gains over these imitation baselines, as well as SimCT, CT-OPD, and GOLD, demonstrate the practical value of transferring knowledge at this decision level.

4.4 Distillation from Black-Box Teachers

To further answer Q2, we evaluate OPAD with the general-purpose black-box teacher GPT-5.6 Terra and the Qwen2.5-3B-Instruct student, using only teacher-generated outputs. To reduce API queries, we introduce OPAD-API, a variant that elicits the teacher target q_t over $\mathcal{C}_t$ through a single call at each selected history h_t, replacing the sampling-frequency target in Section 3.2. To effectively transfer the strong teacher's decision knowledge, we combine forward and reverse KL to retain multiple reasonable actions while encouraging the student to concentrate on teacher-preferred choices. Accordingly, we replace the forward-KL factor multiplying w_t in Eq. (7) with

$$\mathcal{D}_\rho(q_t \| p_{\theta,t}) = D_{\mathrm{KL}}(q_t \| p_{\theta,t}) + \rho\, D_{\mathrm{KL}}(p_{\theta,t} \| \bar{q}_t), \tag{8}$$

where $\rho \geq 0$ weights the reverse-KL term and is set to 1 for symmetric KL, and $\bar{q}_t = (1-\eta_{\mathrm{rev}})\, q_t + \eta_{\mathrm{rev}}/|\mathcal{C}_t|$ slightly smooths the reverse target.

Table 2 shows that OPAD-API outperforms all baselines. Its gains over OmniOPD's text-chunk matching and the response rewards of GAD and ROPD suggest that comparing executable actions can guide interactive decisions more effectively without teacher logits. With GPT-5.6 Terra, the strongest evaluated teacher on ALFWorld Hard, OPAD-API achieves the highest student SR there and improves task scores and Strict SR on WebShop and ScienceWorld, indicating that its action interface can translate stronger teacher capabilities into gains on difficult tasks and enable effective cross-environment guidance for task progress and completion. OmniOPD and ROPD incur $4.15\times$ and $2.98\times$ its teacher API cost per update despite weaker performance; although On-policy SeqKD, DAgger, and GAD cost less, OPAD-API's stronger results offer a favorable trade-off between transfer effectiveness and API spending.

4.5 Distillation Objectives

To examine how the distillation objective shapes action-level transfer, we change only the distillation loss in the main ALFWorld setting. We compare the objectives defined in Appendix A.4: forward KL, reverse KL, top-1 preference supervision (Preference), and the bidirectional KL in Eq. (8) with fixed $\rho \in \{0.3, 0.5, 1\}$ (SymKL for $\rho = 1$) or with an adaptive ρ that scales the reverse term by the fraction of teacher samples yielding valid actions observed more than once.

Effect of the distillation objective. Forward KL outperforms Preference on all three splits in Figure 5, showing the value of supervising relative preferences among multiple executable actions. Reverse KL discourages probability on actions with weak teacher support (Gu et al., 2024), while bidirectional KL balances this concentration with coverage of alternative choices (Sessa et al., 2025). Compared with forward KL, reverse KL improves Hard SR but reduces SR on Seen and Unseen.

SymKL achieves the highest Hard SR and shows substantial performance potential, but also exhibits the largest fluctuations across all three splits. Adaptive ρ leads on Unseen but performs worse on Hard, while other fixed coefficients yield no consistent gains. Forward KL achieves the highest Seen SR and second-highest Unseen SR, with the smallest evaluation fluctuations on both, providing a favorable balance between task performance and consistency. Auxiliary training diagnostics appear in Appendix C.4.

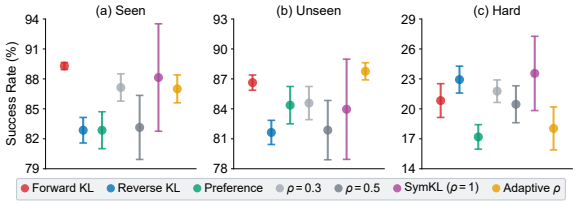

Figure 5: ALFWorld SR under different distillation objectives. Error bars show std.

4.6 Component Ablations

To identify which design choices contribute to OPAD's gains, we conduct ablations under the main ALFWorld configuration. Each variant changes one choice of the full method while keeping the others fixed: for action distributions, Token-Level Supervision applies token-level distillation at the same positions, Teacher Candidate Scoring builds teacher targets by scoring candidates instead of sampling responses, and Top-1 Preference trains on the teacher's most likely action as a hard label; for supervision selection, Global Entropy Top-K ignores local peaks, Random Selection picks positions at random, and All-State Supervision supervises every step under a matched budget; for outcome-aware supervision, Success Only and Failure Only keep only successful or only failed trajectories, Equal K sets $K_{\text{succ}} = K_{\text{fail}}$, and Unit Weights sets every w_t to one. Figure 6 shows Seen results; Appendix C.3 details the variants and reports Unseen and Hard results, which lead to broadly the same conclusions.

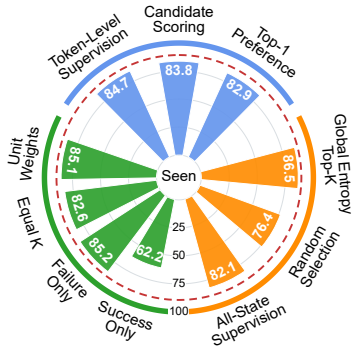

Figure 6: Component ablations on ALFWorld Seen. Bars show SR; the dashed circle marks full OPAD.

On Seen, Token-Level Supervision lowers SR by 4.62 percentage points relative to the full method, while Teacher Candidate Scoring and Top-1 Preference also perform worse. These comparisons distinguish the gains from matching complete actions, constructing teacher targets through sampling, and retaining preferences over multiple actions. Global Entropy Top-K and Random Selection both trail local-peak selection. All-State Supervision also lowers SR; under the same budget of consumed supervision positions, OPAD covers approximately $6.05\times$ as many trajectories. Local peaks thus allocate limited supervision more effectively across trajectories. Success Only produces the largest drop, whereas Failure Only comes closer to the full method but falls short. Equal K and Unit Weights also reduce SR: failure correction is particularly important, while retaining successful trajectories and using outcome-dependent budgets and position weights further improve performance.

5 Conclusion

We presented OPAD, an on-policy distillation framework that transfers agent decision-making knowledge through shared executable actions. Combining distributional action supervision with uncertainty-guided selection and outcome-aware weighting enables effective transfer across student sizes, model generations and families, and teacher access settings. Experiments on ALFWorld, WebShop, and ScienceWorld demonstrate improvements over the evaluated baselines, including cases where students surpass their teacher. The ablations show that learning relative preferences among multiple actions, selecting informative decision points, and balancing success consolidation with failure correction all contribute to effective distillation. Under sampled teacher targets, forward KL offers a favorable balance between retaining alternative decisions and achieving consistent task performance. Together with the action-level performance bound, these findings support executable actions as a practical interface for transferring domain expertise while accommodating the student's own reasoning and tokenization.

AI USE STATEMENT

Generative AI tools were used to assist with manuscript drafting, language editing, identifying relevant literature, and drafting the mathematical proof. The authors reviewed and revised all AI-assisted text for technical accuracy and consistency with the reported methods and results. The authors take full responsibility for the final content of this paper.

REPRODUCIBILITY STATEMENT

Section 3 describes the method, and Appendix A provides the training algorithm and objective details. Appendix B describes the data splits, model and training configurations, evaluation protocol, and prompts. An anonymous implementation of OPAD is available at `https://anonymous.4open.science/status/OPAD-4EF1`.

REFERENCES

Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, et al. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras. *arXiv preprint arXiv:2503.01743*, 2025.

Mohammadreza Armandpour, Fatih Ilhan, David Harrison, Ajay Jaiswal, Duc NM Hoang, Fartash Faghri, Yizhe Zhang, Minsik Cho, and Mehrdad Farajtabar. Unmasking on-policy distillation: Where it helps, where it hurts, and why. *arXiv preprint arXiv:2605.10889*, 2026.

Junfeng Fang, Zhepei Hong, Mao Zheng, Mingyang Song, Gengsheng Li, Houcheng Jiang, Dan Zhang, Haiyun Guo, Xiang Wang, and Tat-Seng Chua. Rubric-based on-policy distillation. *arXiv preprint arXiv:2605.07396*, 2026.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 46375–46408. Curran Associates, Inc., 2025. doi: 10.52202/085713-1544. URL `https://proceedings.neurips.cc/paper_files/paper/2025/file/420c9f777c0b4f78d515e53cf74d58b2-Paper-Conference.pdf`.

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

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

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

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the nineteenth international conference on machine learning*, pp. 267–274, 2002.

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.

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

Changhao Li, Rushi Qiang, Jiawei Huang, Chenxiao Gao, Chao Zhang, Niao He, and Bo Dai. Revisiting dagger in the era of llm-agents. *arXiv preprint arXiv:2605.12913*, 2026a.

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

Xiaozhe Li, Tianyi Lyu, Yang Li, Yichuan Ma, Peiji Li, Linyang Li, Qipeng Guo, Dahua Lin, and Kai Chen. What and when to distill: Selective hindsight distillation for multi-turn agents. *arXiv preprint arXiv:2605.19447*, 2026c.

Zanlin Ni, Shenzhi Wang, Yang Yue, Tianyu Yu, Weilin Zhao, Yeguo Hua, Tianyi Chen, Jun Song, Cheng Yu, Bo Zheng, et al. The flexibility trap: Rethinking the value of arbitrary order in diffusion language models. *arXiv preprint arXiv:2601.15165*, 2026.

Yifan Niu, Han Xiao, Dongyi Liu, Zelong Wang, Dihong Gong, Yasheng Wang, and Jia Li. Breaking the tokenizer barrier: On-policy distillation across model families. *arXiv preprint arXiv:2606.09456*, 2026.

Carlos Miguel Patiño, Kashif Rasul, Quentin Gallouédec, Ben Burtenshaw, Sergio Paniego, Vaibhav Srivastav, Thibaud Frere, Ed Beeching, Lewis Tunstall, Leandro von Werra, and Thomas Wolf. Unlocking on-policy distillation for any model family. `https://huggingface.co/spaces/HuggingFaceH4/on-policy-distillation`, 2025.

Akshara Prabhakar, Zuxin Liu, Ming Zhu, Jianguo Zhang, Tulika Manoj Awalgaonkar, Shiyu Wang, Zhiwei Liu, Haolin Chen, Thai Hoang, Juan Carlos Niebles, Shelby Heinecke, Weiran Yao, Huan Wang, Silvio Savarese, and Caiming Xiong. Apigen-mt: Agentic pipeline for multi-turn data generation via simulated agent-human interplay. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference. Curran Associates, Inc., 2025. doi: 10.52202/085713-2184. URL `https://proceedings.neurips.cc/paper_files/paper/2025/file/5e3661f7fe4c8ac5652d62eb3d3c96ea-Paper-Datasets_and_Benchmarks_Track.pdf`.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 Technical Report, January 2025. URL `http://arxiv.org/abs/2412.15115`. arXiv:2412.15115 [cs].

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

Pier Giuseppe Sessa, Robert Dadashi, Léonard Hussenot-Desenonges, Johan Ferret, Nino Vieillard, Alexandre Ramé, Bobak Shahriari, Sarah Perrin, Abram Friesen, Geoffrey Cideron, et al. Bond: Aligning llms with best-of-n distillation. In *International Conference on Learning Representations*, volume 2025, pp. 59040–59060, 2025.

Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

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.

Jie Sun, Mao Zheng, Mingyang Song, Qiyong Zhong, Yilin Cheng, Bichuan Feng, Pengfei Liu, Junfeng Fang, and Xiang Wang. Simct: Recovering lost supervision for cross-tokenizer on-policy distillation. *arXiv preprint arXiv:2605.07711*, 2026.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.

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

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

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 115452–115486. Curran Associates, Inc., 2025. doi: 10.52202/085713-3850. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/a797c2d2e0c1fdabf4d1ab8cd0b465c6-Paper-Conference.pdf.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022b.

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

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

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.

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.

Yuhang Zhou, Lizhu Zhang, Yifan Wu, Mingyi Wang, Bo Peng, Jiayi Liu, Xiangjun Fan, and Zhuokai Zhao. Omniopd: Logit-free on-policy distillation via speculative verification. *arXiv preprint arXiv:2606.01476*, 2026.

A Algorithmic Supplement

This section gives the complete training procedure, protocol adaptation, black-box teacher variant, and objective definitions used in the loss ablations. A supervision position t is an interaction step; its recorded history h_t, student reasoning r_t, and executed action a_t belong to that step.

A.1 Training Procedure

Algorithm 1 combines protocol initialization with the sampled-teacher training procedure. A trajectory is completed before its outcome determines the selection budget and position weights. Teacher responses are generated only at selected histories; they are not executed in the environment. The operation Pop_B removes B positions from the pending pool. The algorithm describes the open-weight teacher setting; Appendix A.3 gives the API-teacher variant.

Algorithm 1 OPAD training

Require: Base student $\pi_{\theta_{\text{base}}}$, frozen teacher π_ϕ, student-derived protocol examples $\mathcal{D}_{\text{self}}$, environment tasks, update count J, minibatch size B, position budgets $K_{\text{succ}}, K_{\text{fail}}$, teacher sample budget N, and distribution/support hyperparameters

Ensure: Trained student π_{θ_J}

// Phase 1: Initialize the interaction protocol

$\theta_0 \leftarrow \mathrm{ProtocolInit}(\theta_{\text{base}}, \mathcal{D}_{\text{self}})$ (Appendix A.2)

$\mathcal{Q} \leftarrow \varnothing$ ▷ Pool of valid, weighted supervision positions

// Phase 2: Collect student decisions and teacher targets

for $k = 0$ to $J - 1$ **do**

  while $|\mathcal{Q}| < B$ **do**

    Roll out π_{θ_k} in the environment to obtain a complete trajectory τ and outcome $R(\tau)$

    Record (h_t, r_t, a_t, e_t) at each eligible step using Eq. (4)

    $K_\tau \leftarrow K_{\text{succ}}$ if $R(\tau) = 1$, else K_{fail}

    $\mathcal{U}(\tau) \leftarrow \mathrm{PeakThenBackfill}(\{e_t\}, K_\tau)$ (Section 3.3)

    $\mathcal{V}_\tau \leftarrow \varnothing$

    for each $t \in \mathcal{U}(\tau)$ **do**

      Sample N teacher responses independently from h_t; parse and validate their actions

      if $N_t > 0$ **then**

        Build the shared candidate set $\mathcal{C}_t$ and target q_t by Eq. (3)

        Add $(\tau, t, h_t, r_t, a_t, \mathcal{C}_t, q_t)$ to $\mathcal{V}_\tau$

      end if

    end for

    if $\mathcal{V}_\tau \neq \varnothing$ **then**

      Compute each w_t in $\mathcal{V}_\tau$ from $R(\tau)$ and $q_t(a_t)$ (Section 3.4)

      Append the weighted positions in $\mathcal{V}_\tau$ to $\mathcal{Q}$

    end if

  end while

// Phase 3: Match action distributions and update the student

  $\mathcal{B}_k \leftarrow \mathrm{Pop}_B(\mathcal{Q})$

  for each $(\tau, t) \in \mathcal{B}_k$ **do**

    Score every $a \in \mathcal{C}_t$ under (h_t, r_t) to obtain $p_{\theta_k, t}$ by Eq. (2)

    Compute gated action-start loss $\ell_{\text{sup},t}$ by Eq. (10)

  end for

  Compute $\widehat{\mathcal{L}}_k$ by Eq. (11) on $\mathcal{B}_k$

  $\theta_{k+1} \leftarrow \mathrm{OptimizerStep}(\theta_k, \nabla_\theta \widehat{\mathcal{L}}_k)$

end for

return π_{θ_J}

A.2 Protocol Adaptation

Action-level matching compares preferences within a candidate set. Successful environment interaction also requires the student to emit responses in the expected protocol and assign sufficient

generation probability to nonempty action starts. We use protocol self-distillation initialization and a gated action-start support objective to address these requirements. The training procedure is given in Algorithm 1.

Protocol Self-Distillation Initialization. We form initialization examples from actions generated or selected by the frozen student itself and serialize the responses in the required thought–action format. The supervised tokens cover the response tags and action span; the thought content is masked out of the token-level loss. The initialization therefore teaches the student to express its own actions through the interaction interface, without substituting task-optimal action labels or teacher reasoning traces.

Gated Action-Start Support. The candidate softmax is invariant to a common additive shift in all candidate scores. Consequently, matching $p_{\theta,t}$ to q_t alone does not ensure that free generation assigns sufficient probability to entering a nonempty candidate action. We apply an auxiliary loss at the boundary where nonempty action continuations begin.

To account for tokenizer merges around whitespace and delimiters, we tokenize complete candidate responses and an empty-action response in the same reasoning context. Their common token prefix determines a support boundary, denoted b_t. Let $\mathcal{S}_t$ contain the distinct next-token identifiers of valid nonempty candidates at this boundary. Candidates sharing a first token contribute that token only once. The support mass is

$$M_{\theta,t} = \sum_{v \in \mathcal{S}_t} \pi_\theta(v \mid h_t, b_t). \tag{9}$$

Here, the sum uses the model's full-vocabulary probabilities, not probabilities renormalized over the candidate actions. With threshold $\eta_{\text{sup}} \in (0, 1]$, define

$$g_t = \mathbf{1}\{\text{sg}(M_{\theta,t}) < \eta_{\text{sup}}\}, \qquad \ell_{\text{sup},t} = g_t\big(-\log M_{\theta,t}\big), \tag{10}$$

where sg denotes stop-gradient. The gate is evaluated without differentiation, while the active negative log-mass remains differentiable. The log-mass can be evaluated with log-sum-exp over the relevant token log-probabilities.

Complete Optimization Objective. On the same minibatch of valid supervised positions used for action distillation, the complete training objective is

$$\widehat{\mathcal{L}}(\theta; \mathcal{B}) = \frac{1}{|\mathcal{B}|} \sum_{(\tau,t) \in \mathcal{B}} \left[w_t D_{\text{KL}}(q_t \,\|\, p_{\theta,t}) + \lambda_{\text{sup}} \ell_{\text{sup},t} \right]. \tag{11}$$

The outcome weight multiplies only the action-distillation term. The support term has its own coefficient λ_{sup} and activation gate, so a position with zero outcome weight can still receive a support update when its valid-start mass is low. Setting $\lambda_{\text{sup}} = 0$ leaves only the weighted action-distillation term.

A.3 Black-Box Teacher Adaptation

At each selected student history h_t, OPAD-API asks an API teacher to assign probabilities to the indexed complete actions in $\mathcal{C}_t$. The request omits the student's current reasoning r_t and does not identify its executed action as the correct choice. Unlike the sampled-teacher target, this distribution comes from one joint assessment of the candidates. The API returns probabilities and a brief reason; the reason is not used for training.

Let $z_t(a)$ be the returned nonnegative value for action a. The implementation normalizes and smooths the values to obtain

$$q_t^{\text{API}}(a) = \frac{z_t(a) / \sum_{b \in \mathcal{C}_t} z_t(b) + \epsilon_{\text{API}}}{1 + \epsilon_{\text{API}} |\mathcal{C}_t|}. \tag{12}$$

The reference configuration uses $\epsilon_{\text{API}} = 10^{-3}$; positions with missing or nonpositive returned mass have no valid target. We use $q_t^{\text{API}}(a_t)$ as teacher support for the executed student action and retain the outcome weights from Section 3.4. The API experiment uses forward plus reverse KL, with reverse-target smoothing as defined in Appendix A.4.

Teacher prompts. The API teacher receives a benchmark-specific prompt and returns a structured object with `probabilities` keyed by candidate index and a brief `reason`. The reason is not a supervision target.

Prompt Template: ALFWorld API Teacher

[SYSTEM / ROLE INSTRUCTION]
You are an expert ALFWorld next-action policy teacher. Given the current interaction state and the exhaustive admissible action list, return a calibrated probability distribution over every candidate. Probability should reflect how likely each action is to be the best next environment action for completing the task. Do not force a one-hot answer when multiple actions are genuinely plausible.

[USER INPUT]
Current interaction state:
`[role-labelled history]`
Exhaustive admissible actions:
`[numbered candidate actions]`
Assign a probability to every numbered action. The probabilities must be non-negative and sum to 1. Use the numeric keys exactly as supplied.

Prompt Template: WebShop API Teacher

[SYSTEM / ROLE INSTRUCTION]
You are an expert WebShop next-action policy teacher. Given the shopper's request, current page, interaction history, and available actions, return a probability distribution over every numbered candidate. Judge actions by how well they advance finding and purchasing a product that matches the requested attributes. Consider whether the next useful step is searching, inspecting a product, selecting an option, or buying. Do not force a one-hot answer when multiple actions are plausible.

[USER INPUT]
Current interaction state (request, page, and history):
`[role-labelled history]`
Available WebShop actions:
`[numbered candidate actions]`
Assign a non-negative probability to every numbered action so the values sum to 1. Use the numeric keys exactly as supplied.

Prompt Template: ScienceWorld API Teacher

[SYSTEM / ROLE INSTRUCTION]
You are an expert ScienceWorld next-action policy teacher. Given the science task, current observations, interaction history, and available actions, return a probability distribution over every numbered candidate. Judge actions by whether they advance the experiment from its present state, including needed observations, object manipulations, and measurements. Use only the objects and action arguments represented in the candidates. Do not force a one-hot answer when multiple actions are plausible.

[USER INPUT]
Current interaction state (task, observations, and history):
`[role-labelled history]`
Available ScienceWorld actions:
`[numbered complete candidate actions]`
Assign a non-negative probability to every numbered action so the values sum to 1. Use the numeric keys exactly as supplied.

A.4 Ablation Objective Definitions

All objective variants in Section 4.5 use the same action-distribution construction, outcome weights, and gated support auxiliary term. Only the local action loss changes. Forward KL uses $D_{\mathrm{KL}}(q_t \| p_{\theta,t})$. For the reverse direction, the implementation additionally mixes the count-smoothed

teacher target with a uniform distribution:

$$\bar{q}_t(a) = (1 - \eta_{\text{rev}})q_t(a) + \frac{\eta_{\text{rev}}}{|\mathcal{C}_t|}, \qquad \eta_{\text{rev}} = 10^{-3}. \tag{13}$$

Reverse KL minimizes $D_{\text{KL}}(p_{\theta,t} \| \bar{q}_t)$. The bidirectional variants minimize

$$\ell_t^{\text{bi}} = D_{\text{KL}}(q_t \| p_{\theta,t}) + \rho_t D_{\text{KL}}(p_{\theta,t} \| \bar{q}_t). \tag{14}$$

Fixed coefficients are 0.3, 0.5, and 1; the last is labelled SymKL, with the numerical reverse-target smoothing above.

Top-1 Preference. Let $a_t^\star \in \arg\max_{a \in \mathcal{C}_t} q_t(a)$. Preference minimizes $-\log p_{\theta,t}(a_t^\star)$, keeping the same candidate normalization and outcome weighting. It replaces the soft distributional target with one hard action label; it is not a pairwise preference objective. Ties are resolved by the implementation's deterministic candidate ordering.

Adaptive reverse weight. Let S_t count the distinct teacher actions that occur exactly once among N_t valid responses. Let B_t be the maximum of the configured sample budget, the recorded number of valid plus invalid responses, and one. With $N_t > 0$, the adaptive coefficient is

$$\rho_t = \frac{N_t}{B_t} \max\left(0, 1 - \frac{S_t}{N_t}\right) = \frac{N_t - S_t}{B_t}. \tag{15}$$

The configured sample budget is 16. Since each singleton action accounts for one response, $N_t - S_t$ is the number of valid responses whose actions are observed more than once. Thus the coefficient decreases with invalid outputs and unrepeated choices. It measures recurrence within the sampled batch, not whether an action is correct. Positions without a valid teacher target are excluded before loss computation. The teacher target, selected action label, adaptive coefficient, and outcome weight are detached from student differentiation.

B Experimental Details

B.1 Environments and Data Splits

The main comparison, student-scaling experiments, cross-generation and cross-family comparisons, and component and loss ablations use ALFWorld. The general-teacher experiments with xLAM and GPT-5.6 Terra additionally use WebShop and ScienceWorld.

ALFWorld. The official Seen and Unseen evaluation lists contain 140 and 134 tasks, respectively. We also reuse TCOD's fixed 121-task Hard set (Wang et al., 2026). This set was constructed from the ALFWorld training partition by retaining tasks on which TCOD's original teacher failed in all ten sampled attempts. We keep this task list fixed when changing the teacher or student. Hard therefore measures performance on a train-derived difficulty stress test, whereas Unseen assesses held-out generalization.

WebShop and ScienceWorld. Following TCOD's data construction (Wang et al., 2026), WebShop uses session IDs 0–4095 for training and the next 100 IDs (4096–4195) for evaluation. ScienceWorld separates its 30 task types into 17 training types and 13 held-out evaluation types; for each type, it includes the first half of the available variations. This yields 2,294 training instances and 1,308 evaluation instances. WebShop tests product search and selection against a user request, whereas ScienceWorld measures progress on scientific tasks. Table 3 gives the evaluation sizes and interaction limits. Its task counts are evaluation episodes for each trained run, not the number of distinct trajectories collected online.

B.2 Evaluation Protocol and Metrics

At evaluation, the model generates a response, the parser extracts the action, and the environment executes that action directly. The protocol uses model-native chat templates, temperature 0.4, and top-p $= 1.0$. ALFWorld prompts list currently admissible actions; WebShop and ScienceWorld expose their corresponding interaction interfaces. Candidate scoring used during distillation is not an evaluation-time action reranker, and the evaluation does not repair invalid commands. Episodes

Table 3: Evaluation lists and interaction limits.

Environment	Evaluation set	Tasks	Maximum rounds
ALFWorld	Seen	140	30
ALFWorld	Unseen	134	30
ALFWorld	Hard	121	30
WebShop	Held-out sessions	100	15
ScienceWorld	Held-out task types	1,308	30

Table 4: Training configuration for open-weight OPAD training.

Parameter	Value
LoRA rank / scale / dropout	64 / 128 / 0
Optimizer updates / selected states per update	500 / 8
Failure / success selection budgets	$K_{\text{fail}} = 4$ / $K_{\text{succ}} = 1$
Student rollout temperature / top-p	0.4 / 1.0
Training prompt / student generation limit	2,048 / 512 tokens
Teacher samples per selected history	16 next-response samples
Teacher temperature / top-p	0.5 / 0.95
Teacher generation limit	256 tokens per sample
Teacher count smoothing / minimum valid samples	0.001 / 1
Student candidate temperature / length normalization	1.0 / enabled
Candidate scoring microbatch	8 candidates
Default action objective	Outcome-weighted forward KL
Action-start support coefficient / threshold	0.25 / 0.5

are subject to environment termination and the applicable round limit. The reference ALFWorld evaluator allows 16,384 prompt tokens and 512 generated tokens per response.

For M evaluation episodes, let z_i indicate ALFWorld task success, r_i be the terminal WebShop reward, s_i the terminal ScienceWorld score, and R_i the number of interaction rounds. We report ALFWorld success rate as $\mathrm{SR}_{\mathrm{ALF}} = \frac{100}{M}\sum_{i=1}^{M} z_i$; WebShop task score and strict success rate as $\mathrm{TaskScore}_{\mathrm{WebShop}} = \frac{100}{M}\sum_{i=1}^{M} r_i$ and $\mathrm{StrictSR}_{\mathrm{WebShop}} = \frac{100}{M}\sum_{i=1}^{M} \mathbf{1}[r_i = 1]$; ScienceWorld final score and strict success rate as $\mathrm{FinalScore}_{\mathrm{ScienceWorld}} = \frac{1}{M}\sum_{i=1}^{M} \max(s_i, 0)$ and $\mathrm{StrictSR}_{\mathrm{ScienceWorld}} = \frac{100}{M}\sum_{i=1}^{M} \mathbf{1}[s_i \geq 100]$; and mean interaction rounds as $\mathrm{Rounds} = \frac{1}{M}\sum_{i=1}^{M} R_i$. Rounds includes both successful and failed episodes. WebShop Task Score and ScienceWorld Final Score retain their environment-specific definitions and are not pooled into a cross-environment score.

Each trained student method uses five independent training seeds. The reported mean and standard deviation summarize these runs.

B.3 Models, Training, and Hyperparameters

The default teacher is Qwen2.5-7B-Instruct trained on ALFWorld with GiGPO, and the default student is Qwen2.5-3B-Instruct. Scaling changes the student to Qwen2.5-1.5B-Instruct or Qwen2.5-7B-Instruct. Cross-generation transfer uses Qwen3-4B and Qwen3.5-4B; cross-family transfer uses Llama-3.2-3B-Instruct and Phi-4-mini-instruct. These students have broadly comparable parameter counts, but differ in architecture, pretraining, tokenization, and initial capability. General-teacher experiments use xLAM-2-32b-fc-r or GPT-5.6 Terra with Qwen2.5-3B-Instruct.

The default OPAD run and the loss comparisons use 500 optimizer updates. An update uses eight valid selected supervision positions, which may come from multiple trajectories. Training uses four NVIDIA H200 GPUs: one hosts the teacher and three host student workers. The reference implementation fine-tunes LoRA adapters (Hu et al., 2021) on the attention and feed-forward projections while keeping the base-model weights frozen. Table 4 records the training configuration. It distinguishes rollout decoding from differentiable candidate scoring and from evaluation decoding.

Table 5: Baselines for the main ALFWorld and student-scaling comparisons.

Method	Supervision and configuration
SFT	Fixed offline task-supervised training; distinct from protocol initialization.
GRPO	Reinforcement learning from environment rewards.
StepOPSD	GRPO augmented with on-policy self-distillation.
SERL	GRPO augmented with self-distillation-based supervision.
Vanilla OPD	Token-level OPD instantiated for agent interaction on student-generated data.
TCOD	Forward-to-backward curriculum (F2B), the variant specified for the reported comparison.
LSM	Local Support Matching, included as a token-level OPD comparison.
Guided-OPD	Guided token-level OPD with the same external teacher in each teacher comparison.

B.4 Baseline Configurations and Comparison Scope

Tables 5 and 6 specify the comparison groups and the adaptation of each method. SFT, GRPO, StepOPSD, and SERL do not query the fixed external distillation teacher in these comparisons.

Table 6: Additional transfer baselines. "Both" denotes cross-generation/cross-family and black-box comparisons; "Cross-model" denotes the former only.

Method	Scope	Supervision or comparison role
On-policy SeqKD	Both	Cross-entropy on teacher responses queried at student-visited states.
DAgger	Both	Dataset aggregation with teacher action labels using an LLM-agent formulation.
OmniOPD	Both	Teacher-output transfer through text-chunk matching.
GAD	Both	Distillation using teacher feedback through a generated-output interface.
ROPD	Both	Distillation using teacher feedback through a generated-output interface.
GOLD	Cross-model	Additional cross-model distillation baseline.
SimCT	Cross-model	Comparison designed to accommodate differing tokenizers.
CT-OPD	Cross-model	Cross-Tokenizer OPD adapted to agent interaction.

On-policy SeqKD transfers teacher target text at student-visited states; its name denotes the on-policy instantiation of sequence-level distillation. It is distinct from both fixed-dataset SFT and DAgger's aggregation procedure. Text-based targets can be retokenized by the student without requiring a shared vocabulary. In contrast, the token-level OPD group directly tests a different supervision unit from OPAD under the default teacher–student pairing.

TCOD's F2B variant expands the student's interaction horizon from the initial state, whereas its backward-to-forward variant uses precollected successful teacher prefixes (Wang et al., 2026). F2B fits our comparison on student-collected histories without requiring such prefixes.

B.5 Prompt Templates

Prompt Template: ALFWorld Agent

[SYSTEM / ROLE INSTRUCTION]
You are an expert agent operating in the ALFRED Embodied Environment.

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

Prompt Template: WebShop Agent

[SYSTEM / ROLE INSTRUCTION]
You are an agent in WebShop. Choose one action that advances the user's shopping request.

[USER INPUT]
Shopping request: {task_description}
Steps taken: {step_count}. Recent observations and actions ({history_length} steps): {action_history}
Current step: {current_step}. Current page observation: {current_observation}
Actions currently available: [{available_actions}].

Choose one listed action. A search query uses search[<query>] when a search bar is available; a displayed clickable item uses click[<button name>]. Place your reasoning inside <think> </think> and the complete executable command inside <action> </action>.

Prompt Template: ScienceWorld Agent

[SYSTEM / ROLE INSTRUCTION]
You are an agent in the ScienceWorld text environment. Solve the task through environment actions and output exactly one action at each step.

[USER INPUT]
ScienceWorld task: {task_description}
Steps taken: {step_count}. Recent observations and actions ({history_length} steps): {action_history}
Current step: {current_step}. Current observation: {current_observation}
Available action-command templates: [{action_templates}]
Available interactable objects: [{objects}]

Combine a command template with appropriate object names to form one valid action. Place your reasoning inside <think> </think> and the complete action inside <action> </action>.

C Additional Results and Ablation Details

C.1 GPU-Hours to Target Performance

We measure the GPU-hours required to reach at least 70% SR on ALFWorld Seen for the first time. Methods that do not reach 70% are excluded. Each method trains a Qwen2.5-3B-Instruct student using four NVIDIA H200 GPUs. GPU-hours include all four GPUs during online training. Evaluation, the shared protocol cold-start, and teacher pretraining are excluded. Among the six baselines shown, LSM reaches the target with the fewest GPU-hours (41.34); OPAD uses 15.23 GPU-hours, 63.2% less.

Table 7: GPU-hours to first reach 70% SR on ALFWorld Seen.

Method	GPU-hours ↓
GRPO	42.75
StepOPSD	50.10
SERL	53.67
TCOD (F2B)	48.22
LSM	41.34
Guided-OPD	52.82
OPAD	**15.23**

C.2 Distillation from a General-Purpose Open-Weight Teacher

We replace the default teacher with xLAM-2-32b-fc-r (Prabhakar et al., 2025) and train a separate Qwen2.5-3B-Instruct student for ALFWorld, WebShop, and ScienceWorld. Table 8 compares OPAD with the four external-teacher OPD baselines.

Across all five task-performance metrics, OPAD achieves the highest mean among the trained students. Relative to Guided-OPD, it improves ALFWorld SR by 1.28 percentage points, WebShop

Table 8: Distillation from xLAM-2-32b-fc-r to Qwen2.5-3B-Instruct across three environments. Subscripts show ± standard deviations. SR and Strict SR are percentages. Each environment has a separately trained student. Bold and underlined means mark the best and second-best trained-student results.

Method	ALFWorld Seen		WebShop			ScienceWorld		
	SR ↑	Rounds ↓	Task Score↑	Strict SR↑	Rounds ↓	Final Score↑	Strict SR↑	Rounds ↓
Teacher: xLAM-2-32b-fc-r								
Zero-Shot	$33.57_{\pm 3.42}$	$24.20_{\pm 0.71}$	$42.82_{\pm 3.07}$	$14.00_{\pm 2.63}$	$6.52_{\pm 0.43}$	$41.35_{\pm 1.76}$	$28.90_{\pm 1.91}$	$16.10_{\pm 0.36}$
Student: Qwen2.5-3B-Instruct								
Zero-Shot	$7.00_{\pm 0.32}$	$28.50_{\pm 0.28}$	$11.89_{\pm 2.53}$	$2.00_{\pm 1.41}$	$12.67_{\pm 0.46}$	$9.94_{\pm 0.50}$	$1.97_{\pm 0.43}$	$21.34_{\pm 0.35}$
Vanilla OPD	$25.00_{\pm 3.81}$	$25.68_{\pm 0.91}$	$34.76_{\pm 3.18}$	$8.60_{\pm 2.39}$	$8.37_{\pm 0.58}$	$26.84_{\pm 2.12}$	$13.69_{\pm 1.98}$	$22.47_{\pm 0.62}$
TCOD	$29.57_{\pm 3.29}$	$24.89_{\pm 0.78}$	$40.28_{\pm 2.86}$	$12.80_{\pm 2.51}$	$7.58_{\pm 0.49}$	$32.61_{\pm 1.94}$	$19.85_{\pm 1.87}$	$20.76_{\pm 0.55}$
LSM	$28.86_{\pm 3.64}$	$25.01_{\pm 0.86}$	$39.17_{\pm 3.03}$	$12.00_{\pm 2.62}$	$7.73_{\pm 0.53}$	$31.28_{\pm 2.06}$	$18.27_{\pm 2.04}$	$21.14_{\pm 0.58}$
Guided-OPD	$\underline{31.43}_{\pm 3.08}$	$\underline{24.57}_{\pm 0.72}$	$\underline{43.06}_{\pm 2.77}$	$\underline{14.80}_{\pm 2.47}$	$\mathbf{7.21}_{\pm 0.45}$	$\underline{37.49}_{\pm 1.87}$	$\underline{24.36}_{\pm 1.92}$	$\underline{19.43}_{\pm 0.51}$
OPAD	$\mathbf{32.71}_{\pm 3.36}$	$\mathbf{24.35}_{\pm 0.81}$	$\mathbf{45.72}_{\pm 2.89}$	$\mathbf{17.40}_{\pm 2.68}$	$\underline{7.43}_{\pm 0.47}$	$\mathbf{40.26}_{\pm 2.03}$	$\mathbf{27.19}_{\pm 2.14}$	$\mathbf{18.82}_{\pm 0.54}$

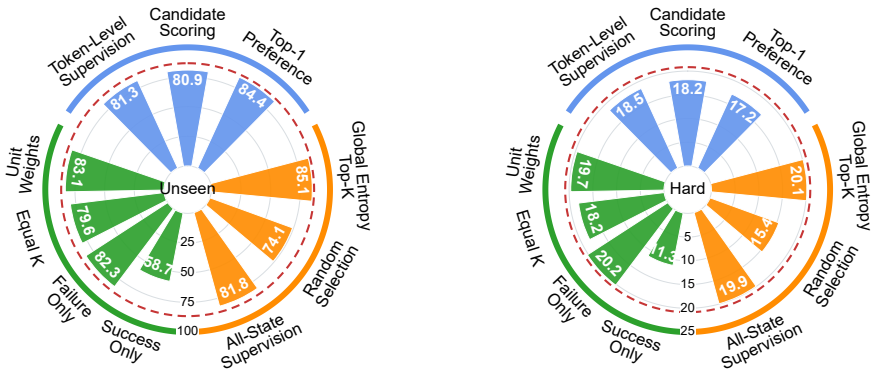

Figure 7: Component ablations on ALFWorld Unseen and Hard. Bars show SR; the outer arcs mark 100% for Unseen and 25% for Hard. Dashed circles mark full OPAD.

Task Score and Strict SR by 2.66 and 2.60 points, and ScienceWorld Final Score and Strict SR by 2.77 and 2.83 points. These consistent gains suggest that the benefit of action-level supervision extends beyond the task-specialized GiGPO teacher and across different interaction environments. On WebShop, OPAD also exceeds xLAM in Task Score and Strict SR; on ALFWorld and ScienceWorld, it remains below the teacher.

C.3 Component-Ablation Variants

We detail the variants used in the three component-ablation groups. Token-Level Supervision replaces action-level matching with token-level distillation at the same selected histories, while Teacher Candidate Scoring constructs teacher targets by scoring and normalizing candidate likelihoods. Top-1 Preference uses the hard-label objective defined in Appendix A.4; its result is reused from that loss ablation. For supervision selection, Global Entropy Top-K chooses the highest-entropy eligible positions without prioritizing local peaks, Random Selection samples under the same position budget, and All-State Supervision uses every eligible position under a matched budget of consumed supervision positions. For outcome-aware supervision, Success Only and Failure Only retain positions from trajectories with the respective outcome, Equal K sets $K_{\text{succ}} = K_{\text{fail}} = 4$, and Unit Weights sets every w_t to one.

Figure 7 shows that the full method has the highest mean SR on both Unseen (86.57%) and Hard (20.83%). Token-Level Supervision reduces SR by 5.24 and 2.35 percentage points, respectively; Teacher Candidate Scoring and Top-1 Preference also fall below the full method. Random Selection has the largest losses among the selection variants, at 12.44 and 5.40 points. Success Only produces

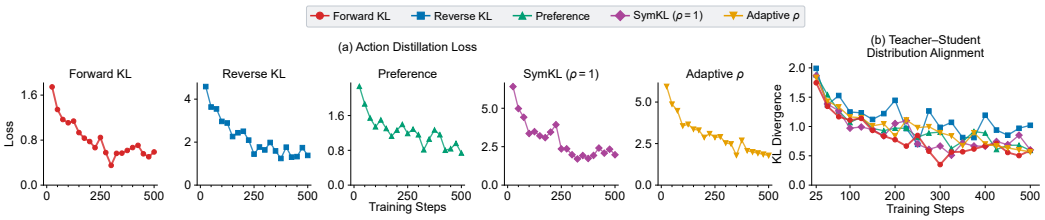

Figure 8: Auxiliary training diagnostics. Each measured point averages 200 consumed supervision positions from a non-overlapping 25-update window. (a) Method-specific action losses before outcome weighting and excluding the protocol-support term; panels use separate scales. (b) Unweighted forward KL $D_{\mathrm{KL}}(q_t \| p_{\theta,t})$ on each student's own visited positions.

Table 9: Outcome allocation in the recorded forward-KL diagnostic run, updates 1–500. Weight sums are rounded to two decimal places; shares are calculated before rounding.

Quantity	Failure	Total	Failure share (%)
Collected trajectories	498	2,591	19.22
Consumed supervision positions	1,916	4,000	47.90
Outcome-weight sum	1,697.98	3,339.71	50.84

the largest losses overall, at 27.83 and 9.57 points, whereas Failure Only remains just 4.23 and 0.62 points below the full method. The contrast shows the greater contribution of failed trajectories in this setting, while retaining successful trajectories still improves mean SR over Failure Only. Equal K, Unit Weights, Global Entropy Top-K, and All-State Supervision also fall below the full method on both splits. Overall, the Unseen and Hard results follow the same broad pattern as Seen.

C.4 Auxiliary Training Dynamics

Figure 8 shows auxiliary training curves from runs using a TCOD teacher checkpoint. The final-performance comparison in Section 4.5 uses the GiGPO teacher.

The method-specific action losses generally decline, but their magnitudes are not directly comparable because the objectives have different definitions and scales. Under the common forward-KL diagnostic, reverse-KL training leaves a larger teacher–student distributional mismatch near the end of the measured run.

C.5 Supervision Allocation by Outcome

Table 9 summarizes one 500-update forward-KL diagnostic run. Failed trajectories make up 19.22% of those collected but provide 47.90% of consumed supervision positions and 50.84% of the outcome-weight sum. This pattern is consistent with the larger selection budget for failed trajectories ($K_{\mathrm{fail}} = 4$ versus $K_{\mathrm{succ}} = 1$). The weight share slightly exceeds the position share, indicating that weighting further emphasizes failure correction while retaining substantial supervision from successful trajectories.

D Action-Level Performance Bound: Formal Statement and Proof

D.1 Setup

Process. A history $h_t = (x, o_1, y_1, \ldots, y_{t-1}, o_t)$ contains all past responses and observations, and the environment returns $o_{t+1} \sim P(\cdot \mid h_t, y_t)$. Episodes last at most $T_{\max}$ steps, and terminated episodes are treated as absorbing, with a fixed outcome and ineffective responses. Policies are defined on full histories, while each model receives its configured prompt, which may summarize

or truncate that history. Define $J(\pi) = \mathbb{E}_{\tau\sim\pi}[R(\tau)]$ for the binary terminal success outcome $R(\tau)$. For a policy π, the occupancy $d^t_\pi(h) = \Pr_\pi(h_t = h,\ \text{episode active at step } t)$ is a sub-probability measure, and $\mathbb{E}_{h_t\sim d^t_\pi}[f]$ denotes $\sum_h d^t_\pi(h) f(h)$.

Let $g(h, y)$ be the action executed after parsing response y at history h, with $\perp$ denoting an invalid action. At each history, let $\mu(\cdot \mid h)$ be the teacher's distribution over valid parsed actions in its next-step responses, conditioned on a valid action being produced. We assume this distribution is defined at the histories considered and is conditioned on h independently of the student's current reasoning prefix.

Values on full histories. For a policy ρ on full histories, define backward in t

$$V^\rho_{T_{\max}+1}(h) = \Pr(R = 1 \mid h), \qquad V^\rho_t(h) = \sum_y \rho(y \mid h)\, Q^\rho_t(h, y),$$
$$Q^\rho_t(h, y) = \mathbb{E}_{o\sim P(\cdot|h,y)}\big[V^\rho_{t+1}(h \oplus (y, o))\big]. \tag{16}$$

Then $J(\rho) = \mathbb{E}[V^\rho_1(h_1)]$, where the law of $h_1 = (x, o_1)$ does not depend on the policy, and $Q^\rho_t \in [0, 1]$. On absorbing histories, $Q^\rho_t(h, \cdot)$ does not depend on the response.

Responses. Every response that opens an action span splits uniquely at the first opening delimiter into a prefix r and a continuation s. Responses without an opening delimiter contribute at most $1 - \omega_\theta(h)$ to the bound. Write $\omega_\theta(h)$ for the probability, under $\widetilde{\pi}_\theta(\cdot \mid h)$, that the response opens an action span, and $\nu_\theta(\cdot \mid h)$ for the conditional distribution of its prefix through that opening delimiter. For a prefix r, $\kappa_\theta(\cdot \mid h, r)$ is the executed-action distribution of the student's continuation, including $\perp$. For each history h, every prefix r in the support of $\nu_\theta(\cdot \mid h)$ when $\omega_\theta(h) > 0$ (or the fixed prefix r_0 when it is zero), and action $a \in \operatorname{supp}\mu(\cdot \mid h)$, assume there is a witness continuation $c_{h,r}(a)$ that executes a, namely $g(h, r \oplus c_{h,r}(a)) = a$. This is the expressibility condition used below. Under the student's sampling distribution $\widetilde{\pi}_\theta(\cdot \mid h)$, including its decoding configuration, $\widetilde{\sigma}_\theta(\cdot \mid h, r)$ is the distribution of the continuation, $\kappa_\theta(a \mid h, r)$ is the probability that it executes a, and $\widetilde{\sigma}_\theta(\cdot \mid h, r, a)$ is its conditional distribution given that it executes a; these are defined for $\omega_\theta(h) > 0$ and r in the support of $\nu_\theta(\cdot \mid h)$. For such h and r, let $\bar{\sigma}_\theta(\cdot \mid h, r, a) = \widetilde{\sigma}_\theta(\cdot \mid h, r, a)$ if $\kappa_\theta(a \mid h, r) > 0$ and the point mass at $c_{h,r}(a)$ otherwise. The reference is

$$\rho_\theta(r \oplus s \mid h) = \nu_\theta(r \mid h)\, \lambda_\theta(s \mid h, r), \qquad \lambda_\theta(s \mid h, r) = \sum_a \mu(a \mid h)\, \bar{\sigma}_\theta(s \mid h, r, a).$$

If $\omega_\theta(h) = 0$, then $\nu_\theta(\cdot \mid h)$ is the point mass at a fixed valid prefix r_0 that ends with its first opening action delimiter, for example the delimiter alone, and $\lambda_\theta(\cdot \mid h, r_0) = \sum_a \mu(a \mid h)\, \delta_{c_{h,r_0}(a)}$; in Eq. (17), the second term is then taken to be zero.

D.2 Formal Statement

Theorem (Theorem 1). *(i) For any student,*

$$\big|J(\rho_\theta) - J(\widetilde{\pi}_\theta)\big| \le \sum_{t=1}^{T_{\max}} \mathbb{E}_{h_t\sim d^t_{\widetilde{\pi}_\theta}}\Big[1 - \omega_\theta(h_t) + \omega_\theta(h_t)\, \mathbb{E}_{r_t\sim\nu_\theta(\cdot|h_t)} \mathrm{TV}\big(\kappa_\theta(\cdot \mid h_t, r_t), \mu(\cdot \mid h_t)\big)\Big], \tag{17}$$

where the second term is zero when $\omega_\theta(h_t) = 0$.

(ii) Let $z_t = (x, o_1, a_1, o_2, \ldots, a_{t-1}, o_t)$, where $a_j = g(h_j, y_j)$. If there is an action policy $\bar{\mu}$ such that $\mu(a \mid h_t) = \bar{\mu}(a \mid z_t)$ at every history, and the conditional laws of the next observation, termination, and success depend on (h_t, y_t) only through $(z_t, g(h_t, y_t))$, then, under the same initial (x, o_1) law, $J(\rho_\theta) = J_{\mathrm{act}}(\bar{\mu})$, the success rate of directly executing $\bar{\mu}$ through the same action interface.

The informal statement in Theorem 1 follows from Eq. (17) because $0 \le \omega_\theta(h) \le 1$ and total variation is nonnegative. Its prefix-expectation term is defined as zero when $\omega_\theta(h_t) = 0$.

D.3 Proof of Part (i)

Proof. Step 1 (performance difference). Let (h_t) be generated by the student and write $\rho = \rho_\theta$. Since $J(\widetilde{\pi}_\theta) = \mathbb{E}[V^\rho_{T_{\max}+1}(h_{T_{\max}+1})]$, telescoping and $\mathbb{E}[V^\rho_{t+1}(h_{t+1}) \mid h_t, y_t] = Q^\rho_t(h_t, y_t)$ give

$$J(\widetilde{\pi}_\theta) - J(\rho) = \sum_{t=1}^{T_{\max}} \mathbb{E}_{h_t \sim d^t_{\widetilde{\pi}_\theta}} \Big[\sum_y \big(\widetilde{\pi}_\theta(y \mid h_t) - \rho(y \mid h_t)\big)\, Q^\rho_t(h_t, y) \Big], \tag{18}$$

the performance difference lemma (Kakade & Langford, 2002) on full histories. Terms from terminated episodes vanish because $Q^\rho_t(h, \cdot)$ is constant on absorbing histories, which is why the occupancy counts active episodes only.

Step 2 (total variation). For distributions P, Q and any f with values in $[0, 1]$, $\big|\sum_y (P(y) - Q(y)) f(y)\big| \le \mathrm{TV}(P, Q)$. Applying this to Eq. (18) gives $|J(\rho) - J(\widetilde{\pi}_\theta)| \le \sum_t \mathbb{E}_{h_t}\big[\mathrm{TV}\big(\widetilde{\pi}_\theta(\cdot \mid h_t), \rho(\cdot \mid h_t)\big)\big]$.

Step 3 (shared prefix). Fix $h = h_t$ and write $\omega = \omega_\theta(h)$. The student's response distribution is $(1 - \omega)P_0 + \omega S$, where P_0 is supported on responses that open no action span and $S(r \oplus s) = \nu_\theta(r \mid h)\, \widetilde{\sigma}_\theta(s \mid h, r)$. Writing $\rho(\cdot \mid h) = (1 - \omega)\rho(\cdot \mid h) + \omega\rho(\cdot \mid h)$, joint convexity of total variation gives

$$\begin{aligned} \mathrm{TV}\big(\widetilde{\pi}_\theta(\cdot \mid h), \rho(\cdot \mid h)\big) &\le (1 - \omega)\, \mathrm{TV}(P_0, \rho) + \omega\, \mathrm{TV}(S, \rho) \\ &\le 1 - \omega + \omega\, \mathbb{E}_{r \sim \nu_\theta(\cdot \mid h)} \mathrm{TV}\big(\widetilde{\sigma}_\theta(\cdot \mid h, r), \lambda_\theta(\cdot \mid h, r)\big), \end{aligned}$$

because $\mathrm{TV}(P_0, \rho) \le 1$, and S and ρ draw the prefix from the same $\nu_\theta(\cdot \mid h)$. When $\omega = 0$, the student's distribution is P_0, and the bound $\mathrm{TV}(P_0, \rho) \le 1$ matches Eq. (17) with the second term taken to be zero; Step 4 below concerns $\omega > 0$.

Step 4 (from continuations to actions). Group continuations by the action they execute; groups for different actions are disjoint. Within the group of an action a with $\kappa_\theta(a \mid h, r) > 0$, the student and the reference assign the text distribution $\widetilde{\sigma}_\theta(\cdot \mid h, r, a)$ with total masses $\kappa_\theta(a \mid h, r)$ and $\mu(a \mid h)$. If $\kappa_\theta(a \mid h, r) = 0$, the student places no mass on the group and the reference places $\mu(a \mid h)$ on $c_{h,r}(a)$. Invalid continuations receive mass $\kappa_\theta(\perp \mid h, r)$ from the student and none from the reference. Summing absolute differences over the groups gives

$$\mathrm{TV}\big(\widetilde{\sigma}_\theta(\cdot \mid h, r), \lambda_\theta(\cdot \mid h, r)\big) = \mathrm{TV}\big(\kappa_\theta(\cdot \mid h, r), \mu(\cdot \mid h)\big),$$

which with Steps 2 and 3 proves Eq. (17). □

D.4 Proof of Part (ii) and Connection to OPAD

Under the assumptions of part (ii), the executed action under ρ_θ conditional on z_t has distribution $\bar{\mu}(\cdot \mid z_t)$. The action-only environment property gives the same conditional laws for the next observation, termination, and success as under direct execution of $\bar{\mu}$. Starting from the common initial distribution, induction over interaction steps gives identical distributions of projected action–observation trajectories, proving $J(\rho_\theta) = J_{\mathrm{act}}(\bar{\mu})$.

Connection to OPAD. The theorem identifies executed-action distributions at student-visited histories as the performance-relevant object. OPAD forms the teacher target q_t from sampled valid actions and the student distribution $p_{\theta,t}$ by scoring shared candidates after the student's own reasoning prefix. Matching these distributions supplies action-level supervision at the histories appearing in the bound.