VALVE: CRITIC-GUIDED ON-POLICY DISTILLATION FOR AGENTIC REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation (OPD) provides dense supervision for training large language model agents, but enforcing teacher preferences can suppress valid alternatives in tasks with multiple solution paths. The challenge is to distinguish disagreements that warrant correction from those that reflect successful alternative behaviors. Our analysis shows that student entropy and teacher–student divergence provide limited separation between these cases, while the relative performance of fixed distillation strengths reverses over training. We propose VALVE, a critic-guided, return-aware method that adapts teacher supervision when combining OPD with reinforcement learning. VALVE uses action-level advantages from the proximal policy optimization (PPO) critic as return credit for task completion, with their signs indicating whether interaction actions are estimated to yield higher or lower return than expected from the current state. It combines this return credit with token-level teacher support, measured by teacher–student log-probability gaps, assigning larger OPD weights when the two signals agree in sign and smaller weights when they conflict. The weights adapt across tokens, interaction steps, and training stages, while retaining the standard PPO actor loss and requiring no additional rollouts or model evaluations beyond PPO and OPD. Across ALFWorld and WebShop with three Qwen students ranging from 1.7B to 7B parameters, VALVE achieves the best or tied-best task success among the evaluated methods. On ALFWorld, it outperforms the strongest competing baseline by 5.3–11.7 percentage points across students. With Qwen2.5-3B, VALVE also yields greater solution diversity than OPD and GRPO on ALFWorld. It reaches the GRPO teacher's success rate on both benchmarks using 66–68% fewer GPU-hours than GRPO.

1 INTRODUCTION

On-policy distillation (OPD) (Lin et al., 2020; Agarwal et al., 2024; Gu et al., 2024) and reinforcement learning with verifiable rewards (RLVR) (Shao et al., 2024; Lambert et al., 2025; Wen et al., 2026) provide complementary signals for post-training agentic large language models. OPD transfers teacher behavior through dense token-level supervision on student-generated trajectories. This supervision can improve learning efficiency, but continually pulling the student toward the teacher may constrain exploration beyond the teacher's preferred behaviors, reduce policy entropy and behavioral diversity, and conflict with reward-driven learning (Li et al., 2026c; Jin et al., 2026; Lan et al., 2026). RLVR instead optimizes the policy from environment rewards, but sparse and delayed rewards make credit assignment to intermediate decisions difficult in multi-step agentic tasks (Zhou et al., 2024; Wang et al., 2026a; Li et al., 2026b). Recent work therefore combines OPD with RLVR to retain dense teacher supervision while allowing environment-driven policy improvement (Yu et al., 2026; Wang et al., 2026a; Lan et al., 2026).

The key question is **how much teacher supervision each decision should receive.** Agentic tasks make this choice nontrivial because the same objective often admits multiple valid solution paths. Consider a WebShop (Yao et al., 2023) instruction to buy black running shoes in size 10 (Figure 1). Selecting size 9 instead of the teacher's size 10 violates an explicit task constraint and can lead to failure. By contrast, selecting the size before the color reverses the teacher's order but can still complete the task. Both choices deviate from the teacher, yet they have different implications for return: OPD can correct the former, while enforcing the latter offers no inherent return benefit and

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

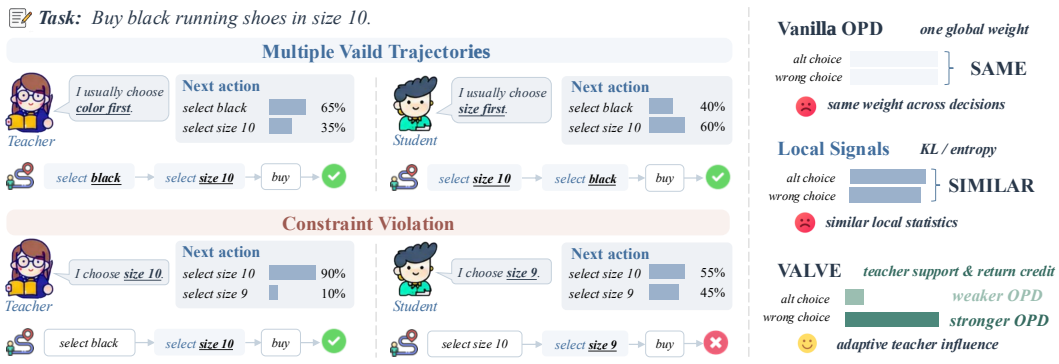


Figure 1: **Motivating example.** VALVE combines token-level teacher support with action-level return credit to adapt the OPD weight for each decision.

may reduce solution diversity. The distinction matters because teacher supervision should serve the task objective rather than merely enforce the teacher's preference.

However, existing adaptive methods cannot reliably make this distinction. They typically adjust distillation strength using local signals such as student uncertainty (Lu et al., 2026; Han et al., 2026; Xu et al., 2026; Tan et al., 2026) or the degree of teacher–student disagreement (Tan et al., 2026; Lu et al., 2026; Zhong et al., 2026; Han et al., 2026). Entropy-based methods measure student uncertainty but cannot distinguish uncertainty among valid alternatives from uncertainty caused by an erroneous choice. Disagreement-based methods measure deviation from the teacher but cannot distinguish a harmful error from a valid preference. Neither signal indicates whether stronger teacher supervision would improve task return, motivating task-grounded feedback.

Our preliminary analysis in Section 2 exposes this ambiguity. These experiments suggest that neither local uncertainty nor the degree of teacher–student disagreement alone can reliably distinguish these two types of teacher-student disagreement, and that the preferred global OPD–RL coefficient shifts as the policy evolves. These findings motivate adapting each token's distillation strength using both teacher guidance and task-grounded feedback.

PPO (Schulman et al., 2017) uses critic-derived advantages to connect policy decisions with downstream return. In language-model training, this credit is typically applied to token-level likelihood terms. In agentic tasks, however, the environment transitions only after receiving a complete multi-token interaction action. We therefore extend PPO credit assignment from token-level likelihood terms to complete interaction actions: the critic estimates one advantage at each interaction boundary and shares it across the tokens of that action. This action-level advantage links each interaction action to downstream task return, with its sign indicating whether the estimated return is above or below the critic's expectation at the current state.

Building on this insight, we propose VALVE, a return-aware method for on-policy distillation over student-generated trajectories. For each sampled token, VALVE combines token-level teacher support with the action-level return credit of the corresponding interaction step. It assigns larger OPD weights when the two signals are consistent and smaller weights when they are inconsistent. The resulting weights vary across tokens, interaction steps, and training stages, while the standard PPO actor loss is retained. VALVE reuses the critic estimates from PPO and the teacher distributions from OPD, requiring no additional model evaluations or rollouts.

We evaluate VALVE on ALFWorld and WebShop using Qwen2.5-3B, Qwen2.5-7B, and Qwen3-1.7B students. On ALFWorld, VALVE achieves success rates of 91.4%, 95.3%, and 92.2%, exceeding GRPO by 18.0, 14.8, and 42.2 percentage points. On WebShop, it improves success over GRPO by 17.2, 21.1, and 9.4 percentage points across the same students, respectively. VALVE further reaches the GRPO teacher's success rate on both benchmarks using 66–68% fewer GPU-hours than GRPO.

In summary, our contributions are as follows:

- We reveal a central ambiguity in teacher guidance: similar local teacher–student disagreements can correspond either to correctable errors or to valid alternatives, and the appropriate reliance on the teacher changes as the student evolves.

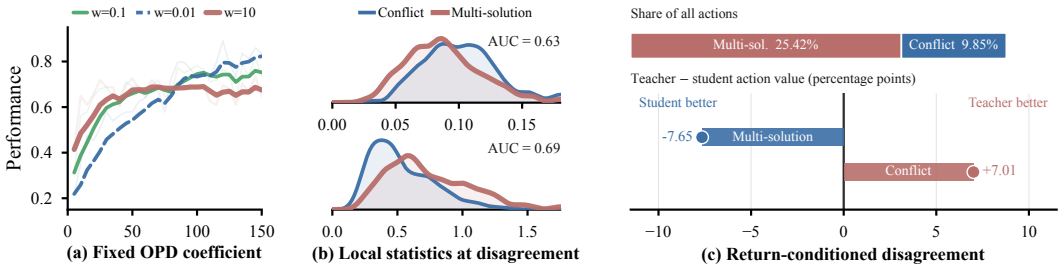


Figure 2: **(a)** Performance under fixed OPD coefficients w. **(b)** Student action entropy (top) and teacher–student divergence $D_{\mathrm{KL}}(S\|T)$ (bottom) at disagreement steps. **(c)** Trajectory-equal group shares (top) and teacher–student action-value gaps (bottom).

- We propose VALVE, a return-aware OPD framework that conditions token-level teacher supervision on action-level return credit while retaining the standard PPO actor loss and requiring no additional model evaluations or rollouts.
- Across two agentic benchmarks and three student models, VALVE matches or exceeds all competing methods in task success, preserves solution diversity, and reaches teacher-level success using 66–68% fewer GPU-hours than GRPO.

2 Preliminary Analysis

We ask when teacher supervision is beneficial and how its strength should vary across student decisions. At teacher–student disagreement steps, we combine algorithmic backward search over feasible task solutions with ALFWorld's built-in task-success verifier to distinguish semantically valid alternatives from successful student trajectories as *multi-solution* disagreements and task-inconsistent actions from failed trajectories as *conflict* disagreements. Using Qwen2.5-3B-Instruct on ALFWorld, Figure 2 examines whether the preferred fixed OPD coefficient changes during training and whether local policy statistics separate the two groups. Further details are provided in Appendix A.2.

The preferred OPD strength changes during training. Figure 2(a) compares fixed OPD coefficients $w \in \{0.01, 0.1, 10\}$ under the same RL objective. Across the six checkpoints at training steps 5–30, the mean performance is 0.622 for $w = 10$, compared with 0.431 for $w = 0.01$. At steps 125–150, this ordering reverses: the corresponding means are 0.673 and 0.811. This reversal shows that the relative performance of fixed OPD coefficients changes over the course of training.

Local statistics provide limited separation. In the balanced comparison in Figure 2(b), the multi-solution and conflict groups exhibit substantially overlapping distributions of both student action entropy and teacher–student KL divergence. The corresponding AUCs are 0.63 and 0.69, respectively, indicating weak separation between the two groups. These local statistics therefore provide limited information for determining whether stronger teacher supervision is appropriate at a given disagreement step.

The relative value of teacher actions depends on the disagreement group. Figure 2(c) shows that, under trajectory-equal aggregation, multi-solution and conflict disagreements account for 25.42% and 9.85% of all action steps, respectively; agreement steps constitute the remaining 64.73%. We estimate each teacher-student disagreement action's expected return using 128 student continuations after its execution from the same pre-action state (Appendix A.3). The teacher-minus-student action-value gap also changes sign across the two groups: it is -7.65 percentage points for multi-solution disagreements and $+7.01$ points for conflict disagreements. Thus, the student action has a higher estimated value in the former group, whereas the teacher action has a higher estimated value in the latter. Together with the limited separation in Figure 2(b), this contrast suggests that local disagreement alone does not determine the return implications of following the teacher.

These findings motivate adapting OPD strength to both policy evolution and task-grounded return information at each decision.

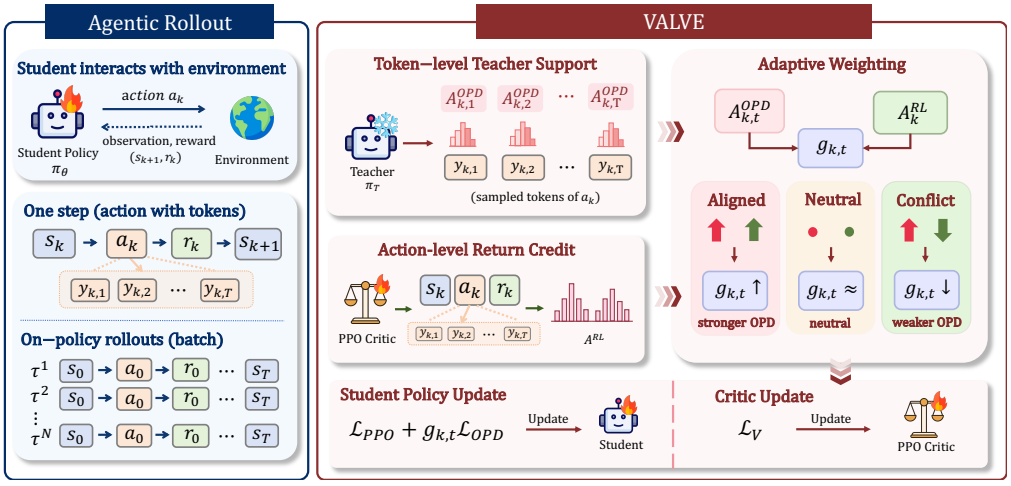


Figure 3: **Overview of VALVE.** The PPO critic assigns action-level return credit, while the frozen teacher supplies token-level support. Their signed product determines the adaptive OPD weight $g_{k,t}$; the student and critic are then updated with their respective objectives.

3 VALVE

Section 2 shows that the relative performance of fixed OPD coefficients changes over training, while local policy statistics provide limited separation between disagreement groups with opposite teacher–student action-value gaps. We introduce VALVE, a return-aware OPD framework that adapts teacher supervision using action-level return credit (Figure 3). VALVE combines the PPO critic's action-level advantage with token-level teacher support to determine an adaptive OPD weight for each sampled token (Sections 3.2 and 3.3), then minimizes the weighted OPD loss together with the clipped PPO actor loss using the same on-policy rollouts (Section 3.4).

3.1 Learning Formulation

We consider multi-turn agentic tasks $x = (q, \mathcal{E}, \mathcal{R}) \sim \mathcal{D}$, where q is an instruction, $\mathcal{E}$ is an interactive environment, and $\mathcal{R}$ is a verifiable reward function. At each interaction step k, the student policy π_θ observes the current state s_k and generates a multi-token interaction action a_k.

$$a_k = (y_{k,1}, \ldots, y_{k,L_k}), \tag{1}$$

where $y_{k,t}$ is generated from token context $h_{k,t}$. Executing a_k produces reward r_k and moves the environment to s_{k+1}. We write a rollout as

$$\tau = (s_1, a_1, r_1, \ldots, s_K, a_K, r_K, s_{K+1}), \tag{2}$$

with terminal outcome $R(\tau) = \mathcal{R}(\tau)$. For terminal-only tasks, $r_k = 0$ for $k < K$ and $r_K = R(\tau)$.

We train the student policy π_θ and an interaction-step critic V_ϕ, while keeping the teacher policy π_T fixed. Following prior work, we perform distillation on student-generated trajectories (Agarwal et al., 2024) using the reverse-KL loss (Gu et al., 2024):

$$\ell_{k,t}^{\mathrm{OPD}}(\theta) = \mathbb{E}_{y \sim \pi_\theta(\cdot \mid h_{k,t})} \left[\log \pi_\theta(y \mid h_{k,t}) - \log \pi_T(y \mid h_{k,t}) \right]. \tag{3}$$

Here, the expectation is over the full student distribution at token context $h_{k,t}$. VALVE retains this loss and adapts only its coefficient.

3.2 Action-Level Return Credit

The PPO critic converts environment feedback into action-level advantage estimates at interaction boundaries. We compute the temporal-difference residual

$$\delta_k = r_k + \gamma V_\phi(s_{k+1}) - V_\phi(s_k), \tag{4}$$

and the generalized advantage estimate

$$A_k^{\mathrm{RL}} = \sum_{l=0}^{K-k} (\gamma\lambda_{\mathrm{GAE}})^l \delta_{k+l}, \tag{5}$$

with $V_\phi(s_{K+1}) = 0$ at the terminal state. Here, A_k^{RL} is a critic-derived estimate of the sampled interaction action's advantage under the current student policy. Its sign indicates whether the action is estimated to yield a higher or lower return than expected from the current state.

Although a_k is generated token by token, the environment evaluates it as a single interaction decision. We therefore broadcast A_k^{RL} to all tokens in a_k, providing action-level return credit for each adaptive OPD weight while retaining context-specific token-level policy losses.

3.3 Return-Aware Token Weighting

Token-level teacher support. To compute the adaptive OPD weight, we use the teacher–student log-probability gap for each sampled rollout token:

$$A_{k,t}^{\mathrm{OPD}} = \log \pi_T(y_{k,t} \mid h_{k,t}) - \log \pi_{\theta_{\mathrm{old}}}(y_{k,t} \mid h_{k,t}), \tag{6}$$

where $\pi_{\theta_{\mathrm{old}}}$ generated the rollout. A positive value indicates stronger relative teacher support for the sampled token, whereas a negative value indicates weaker support. This token-level teacher support is combined with return credit below to determine the adaptive OPD weight applied to the full reverse-KL loss at the same token position.

Teacher–return consistency. The two signals describe the sampled behavior at different granularities: A_k^{RL} is action-level, whereas $A_{k,t}^{\mathrm{OPD}}$ is token-level. Because their magnitudes can differ, we clip $|A_{k,t}^{\mathrm{OPD}}|$ at the magnitude of the unnormalized advantage $|A_k^{\mathrm{RL}}|$ while preserving its sign and retain the notation $A_{k,t}^{\mathrm{OPD}}$ for the rescaled signal; the exact operation is given in Appendix A.4. We broadcast A_k^{RL} to each token in a_k and define the token-level teacher–return consistency score

$$M_{k,t} = A_k^{\mathrm{RL}} A_{k,t}^{\mathrm{OPD}}. \tag{7}$$

When $M_{k,t} > 0$, the teacher's relative support for the sampled token is consistent with the estimated return of the containing action; when $M_{k,t} < 0$, it is inconsistent. The magnitude of $M_{k,t}$ combines the strengths of the two signals.

Adaptive OPD weight. We map each token-level consistency score to a bounded OPD weight:

$$g_{k,t} = \sigma(\beta\, \mathrm{sg}(M_{k,t}))\,, \tag{8}$$

where $\mathrm{sg}(\cdot)$ denotes stop-gradient, $\sigma(\cdot)$ is the sigmoid function, and the gate sharpness $\beta > 0$ controls the response to the consistency score, with $g_{k,t} \in (0, 1)$. A positive score increases the OPD contribution, whereas a negative score attenuates it. The adaptive OPD weight is computed separately for each token and applied to the corresponding token-level OPD term.

3.4 Joint Training Objective

The student is trained by minimizing the clipped PPO actor loss together with an adaptively weighted reverse-KL loss:

$$\mathcal{L}_{\mathrm{VALVE}} = \mathcal{L}_{\mathrm{PPO}} + \mathbb{E}_{\tau\sim\pi_{\theta_{\mathrm{old}}}} \left[\frac{1}{K} \sum_{k=1}^{K} \frac{1}{L_k} \sum_{t=1}^{L_k} g_{k,t} \ell_{k,t}^{\mathrm{OPD}} \right], \tag{9}$$

where $\mathcal{L}_{\mathrm{PPO}}$ denotes the clipped PPO actor loss, including the entropy bonus specified in Appendix A.10. The adaptive OPD weight $g_{k,t}$ scales the full reverse-KL loss at token position (k, t). The gate uses the unnormalized action-level advantage, whereas the clipped PPO actor loss uses its masked-whitened version.

For each on-policy batch, we collect student rollouts, compute A_k^{RL} from the critic, and query the frozen teacher to evaluate $\ell_{k,t}^{\mathrm{OPD}}$ and extract $A_{k,t}^{\mathrm{OPD}}$ for every sampled token. We then form $M_{k,t}$ and $g_{k,t}$. After the policy and critic are updated, the next rollout produces new states, actions, action-level return credit, and token-level teacher support, so the adaptive OPD weights vary across training stages, interaction steps, and token positions. The complete per-batch procedure is given in Algorithm 1.

Table 1: **Main results on ALFWorld and WebShop** (%). Higher is better. Bold and underlined values mark the best and second-best non-reference results within each model block.

		ALFWorld							WebShop	
Type	**Method**	Pick	Look	Clean	Heat	Cool	Pick2	Avg	Score	Acc
Qwen2.5-3B (Teacher: Qwen2.5-3B-GRPO)										
Reference	Base (Qwen et al., 2025)	44.4	11.1	6.2	15.4	28.6	12.5	21.9	6.7	0.8
RL	PPO (Schulman et al., 2017)	78.8	66.7	66.7	72.2	66.7	43.5	66.4	48.3	42.2
RL	GRPO (Shao et al., 2024)	93.5	83.3	63.2	85.0	61.5	50.0	73.4	69.0	53.9
Global	Fixed OPD (Agarwal et al., 2024)	93.5	75.0	84.2	65.0	73.1	45.0	74.2	72.0	57.8
Global	Annealed OPD (Tan et al., 2026)	93.5	**91.7**	73.7	75.0	73.1	55.0	77.3	77.4	60.9
Local	KL-weighted (Zhong et al., 2026)	90.3	66.7	73.7	80.0	69.2	60.0	75.0	74.2	60.2
Local	Entropy-weighted (Jin et al., 2026)	96.8	75.0	68.4	65.0	69.2	50.0	72.7	72.1	57.0
Adaptive	ATOD (Tan et al., 2026)	93.5	58.3	68.4	70.0	**84.6**	55.0	75.0	74.0	59.4
Adaptive	KDRL (Xu et al., 2025)	90.3	**91.7**	68.4	50.0	61.5	65.0	71.1	77.0	59.4
Adaptive	RLSD (Yang et al., 2026)	87.9	75.0	90.9	75.0	73.1	68.4	79.7	84.4	66.4
Adaptive	SDAR (Lu et al., 2026)	97.1	62.5	**100.0**	61.9	75.0	84.2	84.4	85.0	68.0
Adaptive	AgentOPSD (Wang et al., 2026a)	**100.0**	68.8	86.4	66.7	**84.6**	73.7	84.4	**90.4**	69.5
Adaptive	VALVE (Ours)	97.1	86.7	95.8	**93.3**	83.3	**86.7**	**91.4**	87.6	**71.1**
Qwen2.5-7B (Teacher: Qwen2.5-7B-GRPO)										
Reference	Base (Qwen et al., 2025)	36.1	22.2	3.1	0.0	0.0	0.0	12.5	5.9	1.6
RL	PPO (Schulman et al., 2017)	92.3	64.0	92.5	89.5	80.3	68.8	80.4	81.4	68.7
RL	GRPO (Shao et al., 2024)	**100.0**	75.0	84.2	60.0	73.1	80.0	80.5	77.2	61.7
Global	Fixed OPD (Agarwal et al., 2024)	**100.0**	66.7	89.5	50.0	69.2	75.0	77.3	88.2	76.6
Global	Annealed OPD (Tan et al., 2026)	97.1	63.6	87.5	54.5	55.0	68.4	77.3	85.8	70.3
Local	KL-weighted (Zhong et al., 2026)	**100.0**	72.7	93.8	72.7	65.0	94.7	87.5	84.2	70.3
Local	Entropy-weighted (Jin et al., 2026)	96.8	66.7	94.7	70.0	73.1	75.0	81.3	82.1	65.6
Adaptive	ATOD (Tan et al., 2026)	**100.0**	66.7	84.2	75.0	73.1	80.0	82.0	83.1	75.8
Adaptive	KDRL (Xu et al., 2025)	91.4	86.7	91.7	60.0	75.0	73.3	82.0	77.8	67.2
Adaptive	RLSD (Yang et al., 2026)	**100.0**	87.5	92.3	58.8	80.0	65.2	82.0	87.4	77.3
Adaptive	SDAR (Lu et al., 2026)	94.7	75.0	**100.0**	86.7	68.2	78.9	85.9	89.4	**82.8**
Adaptive	AgentOPSD (Wang et al., 2026a)	91.2	87.5	**100.0**	90.5	75.0	84.2	89.1	**90.2**	79.7
Adaptive	SERL (Li et al., 2026b)	92.3	**100.0**	88.9	**100.0**	76.2	82.6	90.0	89.5	80.1
Adaptive	VALVE (Ours)	**100.0**	93.3	91.7	**100.0**	**87.5**	**100.0**	**95.3**	89.8	**82.8**
Qwen3-1.7B (Teacher: Qwen3-4B-GRPO)										
Reference	Base (Yang et al., 2025)	25.0	22.2	3.1	0.0	21.4	4.2	12.5	46.5	4.7
Reference	Teacher (Yang et al., 2025)	100.0	75.0	73.7	60.0	84.6	80.0	81.3	80.4	71.1
RL	PPO (Schulman et al., 2017)	48.5	40.0	23.8	38.9	44.4	17.4	35.9	49.0	41.4
RL	GRPO (Shao et al., 2024)	71.0	41.7	52.6	30.0	53.8	35.0	50.0	73.9	54.7
Global	Fixed OPD (Agarwal et al., 2024)	**100.0**	75.0	63.2	60.0	73.1	70.0	75.8	59.4	50.0
Global	Annealed OPD (Tan et al., 2026)	96.8	66.7	84.2	60.0	**80.8**	55.0	76.6	48.5	42.1
Local	KL-weighted (Zhong et al., 2026)	93.5	**83.3**	63.2	70.0	76.9	75.0	78.1	60.6	50.8
Local	Entropy-weighted (Jin et al., 2026)	96.8	66.7	68.4	65.0	73.1	70.0	75.8	72.2	57.0
Adaptive	ATOD (Tan et al., 2026)	**100.0**	75.0	84.2	65.0	**80.8**	65.0	80.5	46.6	42.1
Adaptive	KDRL (Xu et al., 2025)	87.1	16.7	26.3	55.0	46.2	40.0	50.8	69.3	53.9
Adaptive	RLSD (Yang et al., 2026)	50.0	37.5	61.5	19.0	50.0	21.1	42.2	74.0	50.8
Adaptive	SDAR (Lu et al., 2026)	73.5	25.0	76.9	33.3	40.0	36.8	53.9	**76.8**	58.6
Adaptive	VALVE (Ours)	97.1	81.8	**100.0**	**81.8**	80.0	**94.7**	**92.2**	72.6	**64.1**

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Benchmarks. We evaluate VALVE on ALFWorld and WebShop, two multi-turn agent benchmarks. ALFWorld (Shridhar et al., 2021) comprises text-based household tasks with sparse terminal rewards. We report task success across six categories: Pick, Look, Clean, Heat, Cool, and Pick2. WebShop (Yao et al., 2023) evaluates product search and purchase under natural-language constraints. We report both task score and success rate.

Baselines. We compare against three groups of baselines. **(1) RL only.** We include PPO (Schulman et al., 2017) and GRPO (Shao et al., 2024). **(2) OPD weighting controls.** We include Fixed OPD (Agarwal et al., 2024) and Annealed OPD (Tan et al., 2026) as global weighting baselines, together with KL-weighted OPD (Zhong et al., 2026) and entropy-weighted OPD (Jin et al., 2026) as local-statistic baselines. **(3) Adaptive RL–distillation.** We compare against ATOD (Tan et al., 2026), KDRL (Xu et al., 2025), RLSD (Yang et al., 2026), SDAR (Lu et al., 2026), AgentOPSD (Wang et al., 2026a), and SERL (Li et al., 2026b). For each model block, we additionally report the initial student and, when the student and teacher differ in scale, the teacher.

Implementation. We use Qwen2.5-3B/7B-Instruct (Qwen et al., 2025) and Qwen3-1.7B (Yang et al., 2025) as students. Their frozen teachers are step-150 GRPO checkpoints: same-scale Qwen2.5 models for the Qwen2.5 students and Qwen3-4B for Qwen3-1.7B. Unless otherwise specified, we

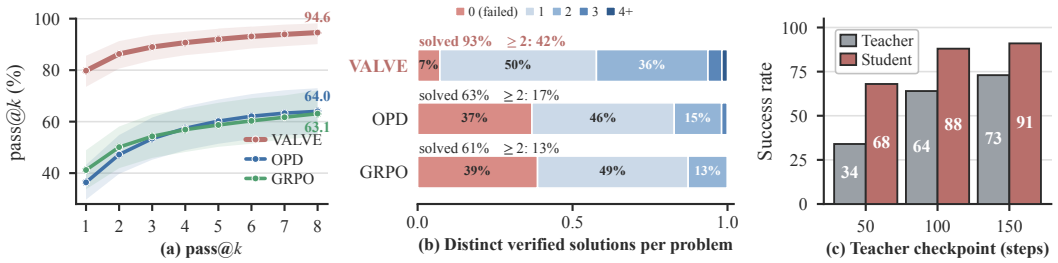


Figure 4: **(a)** pass@k on 111 multi-solution ALFWorld problems under a matched sampling budget. **(b)** Distribution of distinct verified solutions per problem. **(c)** Success rates of GRPO teachers after 50/100/150 steps and their VALVE students.

use actor/critic learning rates of $1 \times 10^{-6}/1 \times 10^{-5}$, $\gamma = 0.95$, and $\beta = 10$, and report student checkpoints at step 150. Further details are provided in Section A.10.

4.2 Main Results

Overall performance. Across both benchmarks and all three students, VALVE achieves the best or tied-best success rates among the compared training methods (Table 1). On ALFWorld, its Avg scores reach 91.4%, 95.3%, and 92.2% with Qwen2.5-3B, Qwen2.5-7B, and Qwen3-1.7B, exceeding GRPO by 18.0, 14.8, and 42.2 points, respectively. On WebShop, its Acc reaches 71.1%, 82.8%, and 64.1%, improving over GRPO by 17.2, 21.1, and 9.4 points.

Performance beyond the teacher. Both Qwen2.5 students outperform their same-scale GRPO teachers on ALFWorld Avg and WebShop Acc. The smaller Qwen3-1.7B student also surpasses its Qwen3-4B teacher on ALFWorld Avg (92.2% vs. 81.3%). We further examine robustness to teacher maturity across training stages in Section 4.3.

4.3 Analysis

We analyze VALVE in terms of solution diversity, training dynamics, robustness to teacher maturity, and compute efficiency, using Qwen2.5-3B on ALFWorld unless otherwise specified.

Task success and solution diversity. On 111 multi-solution ALFWorld problems, VALVE achieves higher pass@k than OPD and GRPO at every evaluated k under the same eight-rollout budget (Figure 4(a)). At $k = 8$, it reaches a pass@8 of 94.6%, compared with 64.0% for OPD and 63.1% for GRPO. Under the environment-based criterion defined in Appendix A.1, VALVE finds at least two distinct verified solutions for 42% of the problems, compared with 17% for OPD and 13% for GRPO (Figure 4(b)). Together, these results indicate that VALVE improves task success while preserving a broader set of valid solution paths.

Training dynamics. Figure 5 illustrates how VALVE combines rapid early learning with sustained improvement by adapting distillation throughout training. In Figure 5(a), VALVE catches up with OPD by step 30 and continues to improve after OPD largely plateaus, ultimately reaching a higher validation success rate. The OPD weight distributions in Figure 5(b) reveal the corresponding adaptation: initially polarized weights near 0 and 1 shift toward a broad distribution centered around 0.5, while retaining token-level variation. This redistribution coincides with the entropy dynamics in Figure 5(c): whereas entropy-weighted and KL-weighted OPD exhibit a sharp early decline in policy entropy, VALVE maintains higher entropy early in training, followed by a more gradual decline. Together, these dynamics suggest that return-aware weighting balances rapid teacher-guided learning with continued exploration, allowing VALVE to sustain improvement beyond OPD's plateau.

Robustness to teacher maturity. We vary teacher maturity using frozen GRPO checkpoints at steps 50, 100, and 150, while holding all other settings fixed. With the weakest 50-step teacher, VALVE doubles the teacher's success rate from 34% to 68%. Moving from the 100-step to the 150-step teacher raises teacher success from 64% to 73%, but VALVE improves only from 88% to

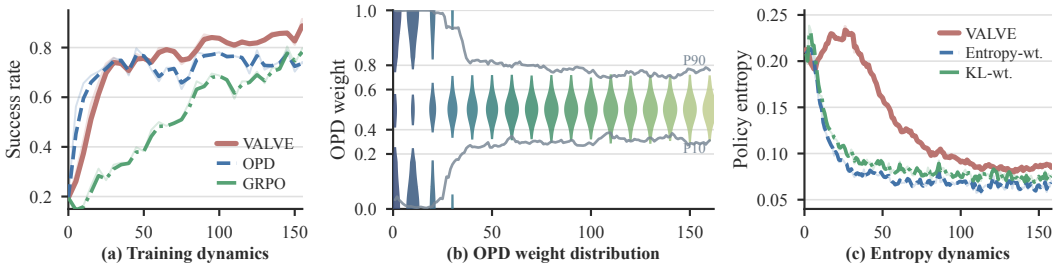


Figure 5: **Training dynamics. (a)** Validation success under VALVE, OPD, and GRPO. **(b)** Distribution of VALVE OPD weights at each checkpoint, with 10th and 90th percentiles. **(c)** Policy entropy under VALVE, entropy-weighted OPD, and KL-weighted OPD.

Table 2: **GPU-hours to reach fixed performance targets.** Targets are 25/50/100% of the GRPO teacher's success rate. **best** and second-best.

	WebShop			ALFWorld		
Method	25%	50%	100%	25%	50%	100%
OPD	**2.36**	**2.36**	**4.84**	**0.49**	**2.11**	**10.07**
VALVE (Ours)	5.61	6.73	7.54	0.60	4.57	14.60
GRPO	6.37	9.45	22.05	0.73	15.00	45.31

Table 3: **Ablation on ALFWorld with Qwen2.5-3B.** Δ is the success-rate drop from the full VALVE model in percentage points.

Variant	Success (%)	Δ
VALVE (full)	**91.4**	–
- w/o action-level credit	28.1	$\downarrow$ 63.3
- w/o adaptive weighting	74.2	$\downarrow$ 17.2
- w/o return credit in g	81.2	$\downarrow$ 10.2
- w/o teacher support in g	83.5	$\downarrow$ 7.9

91% (Figure 4(c)). Together, these results show that VALVE can benefit substantially from a weak teacher and reach near-peak performance without relying on the most mature teacher.

Compute–performance trade-off. Table 2 reports the GPU-hours required by Fixed OPD, VALVE, and GRPO to reach 25%, 50%, and 100% of the GRPO teacher's success rate on WebShop and ALFWorld; cost-accounting details are provided in Appendix A.9. Fixed OPD reaches the teacher-level target with the least compute but improves little thereafter, resulting in lower final performance than VALVE (Table 1). At the same target, VALVE requires 7.54 and 14.60 GPU-hours, compared with 22.05 and 45.31 for GRPO, reducing compute by 66–68%. Thus, VALVE retains OPD's rapid progress toward teacher-level performance while continuing to improve after Fixed OPD plateaus.

4.4 Ablation Study

As shown in Table 3, replacing action-level return credit with trajectory-level reward causes the largest drop, reducing success from 91.4% to 28.1% ($\downarrow$ 63.3 points). This drop shows that coarse trajectory feedback is insufficient for deciding which interaction steps should receive stronger teacher supervision. Fixing the OPD weight to $g_{k,t} = 1$ reduces success to 74.2%, while removing return credit or teacher support from the weight yields 81.2% and 83.5%, respectively. Thus, adaptive weighting requires both action-level return credit and token-level teacher support to condition teacher supervision on task return. Implementation details are provided in Appendix A.8.

4.5 Case Study

We next illustrate how VALVE preserves multiple successful routes for the same WebShop goal. Figure 6 compares a teacher-preferred route, an alternative route, and a self-correcting route under VALVE, GRPO, and OPD. VALVE assigns high probability to the decisive actions on all three routes, whereas GRPO and OPD concentrate probability on a narrow path and assign near-zero mass to at least one alternative decision. This example suggests that return-aware weighting keeps multiple valid behaviors executable while preserving reliable task completion.

Teacher-preferred route — success

Step / action	VALVE	GRPO	OPD
search[women's tops …]	**0.97**	0.99	0.99
click[b097rk2b2q] *item*	**0.80**	0.68	0.63
click[b3-black] *colour*	**0.91**	0.92	0.91
click[small] *size*	**0.96**	0.97	0.96
click[buy now]	**0.95**	**0.10**	**<.01**

Alternative route — success

Step / action	VALVE	GRPO	OPD
search[…] *reformulated*	**0.38**	0.03	0.03
click[b094q7b3ss] *alt item*	**0.79**	0.64	0.70
click[a01#black] *alt colour*	**0.87**	**0.09**	**0.15**
click[small] *size*	**0.95**	0.97	0.96
click[buy now]	**0.94**	0.37	0.59

Self-correcting route — success

Step / action	VALVE	GRPO	OPD
search[women's tops …]	**0.97**	0.99	0.99
click[b09s3bn15c] ×wrong	**<.01**	<.01	<.01
click[back to search] ↺	**0.05**	0.08	0.07
click[b097rk2b2q] ✓fixed	**0.69**	0.54	0.59
click[b3-black], click[small]	**0.91**	0.92	0.90
click[buy now]	**0.95**	**0.21**	**0.01**

Figure 6: **Action probabilities across three successful WebShop routes.** *Task: Buy a loose-fit women's top in black, size small, for under $40.* Cells show per-step action probabilities under VALVE, GRPO, and OPD. Green and red denote high and low probability.

5 Related Work

On-policy distillation with reinforcement learning. Knowledge distillation transfers teacher distributions to student models and has been extended to autoregressive generation and decision policies (Hinton et al., 2015; Kim & Rush, 2016; Rusu et al., 2016). On-policy language-model distillation trains on student-generated sequences and studies objectives suited to autoregressive generation (Lin et al., 2020; Gu et al., 2024; Ko et al., 2024). Recent methods integrate OPD with reinforcement learning to combine dense teacher supervision with task-reward optimization (Yu et al., 2026; Wang et al., 2026a; Lan et al., 2026). These methods differ primarily in how teacher influence is controlled during policy learning.

Adaptive teacher guidance. Existing methods regulate teacher influence through training-stage or turn-level schedules (Li et al., 2026a; Yu et al., 2026), local uncertainty and teacher–student disagreement (Lu et al., 2026; Han et al., 2026; Xu et al., 2026; Tan et al., 2026; Zhong et al., 2026), or reward-, group-, influence-, and hindsight-based feedback (Akhondzadeh et al., 2026; Xu et al., 2025; Lian et al., 2026; Wang et al., 2026a; Lan et al., 2026; Li et al., 2026b). As summarized in Table 4, VALVE combines action-level return credit from the PPO critic with token-level teacher support, enabling adaptation across tokens, interaction steps, and training stages.

Table 4: **Comparison of OPD–RL weighting schemes.** Columns indicate adaptation across training stages and interaction steps and the use of action-level return credit.

Method	Training stages	Interaction steps	Action-level return credit
Fixed OPD	×	×	×
Annealed OPD	✓	×	×
ATOD	✓	✓	×
SDAR	✓	✓	×
VALVE (Ours)	✓	✓	✓

6 Conclusion

We studied how to regulate teacher supervision in multi-turn agentic tasks, where teacher–student disagreement may indicate either a correctable error or a successful alternative. We introduced VALVE, a critic-guided, return-aware OPD method that combines action-level return credit from the PPO critic with token-level teacher support, assigning larger distillation weights when the two signals agree in sign and smaller weights when they conflict. Across ALFWorld and WebShop with three Qwen students, VALVE achieves the best or tied-best task success among the evaluated methods and outperforms the strongest competing baseline on ALFWorld by 5.3–11.7 percentage points across students. With Qwen2.5-3B, it yields greater solution diversity than OPD and GRPO on ALFWorld and reaches the GRPO teacher's success rate on both benchmarks using 66–68% fewer GPU-hours than GRPO. Together, these results support using action-level return credit to regulate teacher influence while preserving successful alternatives.

AI USE STATEMENT

Generative AI tools were used to draft and revise portions of the manuscript, improve readability, edit LaTeX and table formatting, locate and verify bibliographic metadata, and check compliance with the ICLR 2027 guidelines. They were not used to generate experimental data or execute experiments. The authors reviewed all AI-assisted text and artifacts, verified the technical claims, results, and citations, and take full responsibility for the final content of this work.

ETHICS STATEMENT

We have read and adhere to the ICLR Code of Ethics. This work uses publicly available benchmark environments, ALFWorld (Shridhar et al., 2021) and WebShop (Yao et al., 2023), and pretrained language models; it involves no human participants, user studies, collection of personal data, or interaction with real users or commercial services. All agent actions are executed in benchmark environments, and the WebShop experiments place no real orders. More capable language agents could be misused if deployed for autonomous web interaction; such deployment should include task-specific safeguards and human oversight. We report the compute requirements and evaluation limitations in Sections A.9 and A.10.

REPRODUCIBILITY STATEMENT

We provide the information needed to reproduce our experiments throughout the paper and appendix. Section 3 defines the learning objective and return-aware token weighting; Section A.4 details the signal rescaling, and Section A.5 gives the complete training procedure. Section A.6 specifies the benchmark splits, metrics, interaction budgets, seeds, student models, and teachers. Section A.9 describes compute-cost accounting, while Section A.10 documents the hardware, optimization hyperparameters, batch and context settings, and evaluation details, including the use of one fixed comparison seed without a seed sweep.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes, 2024. URL https://arxiv.org/abs/2306.13649.

Mohammad Sadegh Akhondzadeh, Vijay Lingam, Atula Tejaswi, Chanakya Ekbote, Sujay Sanghavi, and Aleksandar Bojchevski. Reward-gated on-policy distillation, 2026. URL https://arxiv.org/abs/2607.04037.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training, 2025. URL https://arxiv.org/abs/2505.10978.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 32694–32717, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/8ac015d409635f196f9e3e9dcfb9a94e-Paper-Conference.pdf.

Zhuowen Han, Jinwei Xiao, Zhengxi Lu, Renren Jin, Zhiyuan Yao, Yuxin Liu, Hongyan Hao, Yueqing Sun, Yu Yang, Qi GU, Xunliang Cai, and Deyi Xiong. Distill where you fail: Recovering learning signals of negative rl-groups from adaptive teacher guidance, 2026. URL https://arxiv.org/abs/2608.00782.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

Woogyeol Jin, Taywon Min, Yongjin Yang, Dennis Wei, Yi Zhou, Swanand Ravindra Kadhe, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models, 2026. URL https://arxiv.org/abs/2603.07079.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In Jian Su, Kevin Duh, and Xavier Carreras (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1139. URL https://aclanthology.org/D16-1139/.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DistiLLM: Towards streamlined distillation for large language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 24872–24895. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/ko24c.html.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tulu 3: Pushing frontiers in open language model post-training, 2025. URL https://arxiv.org/abs/2411.15124.

Qizhen Lan, Xi Xiao, Xiangchen Guan, Mengchen Fan, Moule Lin, Jung Im Choi, and Lijing Zhu. Trust is not enough: Influence calibration for on-policy self-distillation in agentic rl, 2026. URL https://arxiv.org/abs/2608.14945.

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, 2026a. URL https://arxiv.org/abs/2606.15912.

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, 2026b. URL https://arxiv.org/abs/2605.19447.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan ang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe, 2026c. URL https://arxiv.org/abs/2604.13016.

Niu Lian, Tongbo Chen, Zhehao Yu, Chengzhen Duan, Fazhan Liu, Hui Liu, Pei Fu, Jian Luan, Heng Qu, Shu-Tao Xia, and Jinpeng Wang. Ui-mopd: Multi-platform on-policy distillation for unified gui agents, 2026. URL https://arxiv.org/abs/2607.04425.

Alexander Lin, Jeremy Wohlwend, Howard Chen, and Tao Lei. Autoregressive knowledge distillation through imitation learning. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6121–6133, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.494. URL https://aclanthology.org/2020.emnlp-main.494/.

Zhengxi Lu, Zhiyuan Yao, Zhuowen Han, Zi-Han Wang, Jinyang Wu, Qi Gu, Xunliang Cai, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. Self-distilled agentic reinforcement learning, 2026. URL https://arxiv.org/abs/2605.15155.

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, 2025. URL https://arxiv.org/abs/2412.15115.

Andrei A. Rusu, Sergio Gomez Colmenarejo, Caglar Gulcehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy distillation, 2016. URL https://arxiv.org/abs/1511.06295.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 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, 2021. URL https://arxiv.org/abs/2010.03768.

Qitai Tan, Zefang Zong, Mo Li, Yipeng Shi, Yang Li, and Peng Chen. Atod: Annealed turn-aware on-policy distillation for multi-turn agentic tasks, 2026. URL https://arxiv.org/abs/2606.27814.

Zi-Han Wang, Zhengxi Lu, Zhiyuan Yao, Jinyang Wu, Jie Wu, Zhengzhou Cai, Yueqing Sun, Ziang Ye, Linji Hao, Qi Gu, Xunliang Cai, Yongliang Shen, and Yujiu Yang. Agentopsd: Recursive self-distillation for agentic reinforcement learning, 2026a. URL https://arxiv.org/abs/2608.05987.

Zixuan Wang, Yuchen Yan, Hongxing Li, Teng Pan, Dingming Li, Ruiqing Zhang, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. Milestone-guided policy learning for long-horizon language agents, 2026b. URL https://arxiv.org/abs/2605.06078.

Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhirong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, Ziming Miao, Jiang Bian, and Mao Yang. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base llms. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 49450–49483, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/517f9b9c227b9dd51dba4560f37165ed-Paper-Conference.pdf.

Hongling Xu, Qi Zhu, Heyuan Deng, Jinpeng Li, Lu Hou, Yasheng Wang, Lifeng Shang, Ruifeng Xu, and Fei Mi. Kdrl: Post-training reasoning llms via unified knowledge distillation and reinforcement learning, 2025. URL https://arxiv.org/abs/2506.02208.

Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, Zhipeng Wang, and Alborz Geramifard. Tip: Token importance in on-policy distillation, 2026. URL https://arxiv.org/abs/2604.14084.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. Self-distilled rlvr, 2026. URL https://arxiv.org/abs/2604.03128.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents, 2023. URL https://arxiv.org/abs/2207.01206.

Yan Yu, Zhengxi Lu, Yizhou Liu, Yichen Pan, Aozhe Wang, Qipeng Chen, Hua Yang, Wenqi Zhang, Weiming Lu, Qianglong Chen, and Yongliang Shen. Retireopd: Self-retiring on-policy distillation for agentic reinforcement learning, 2026. URL https://arxiv.org/abs/2609.20784.

Zijing Zhang, Ziyang Chen, Mingxiao Li, Zhaopeng Tu, and Xiaolong Li. Rlvmr: Reinforcement learning with verifiable meta-reasoning rewards for robust long-horizon agents, 2025. URL https://arxiv.org/abs/2507.22844.

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, 2026. URL https://arxiv.org/abs/2605.07725.

Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. ArCHer: Training language model agents via hierarchical multi-turn RL. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 62178–62209. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/zhou24t.html.

A Appendix

A.1 Definition and Identification of Distinct Solutions

Definition. For a fixed ALFWorld task and initial environment configuration, we consider two trajectories to be distinct solutions if both successfully complete the task and differ in either the order of operations or the objects used to satisfy the task requirements. A task is termed *multi-solution* if it admits at least two such solutions. Textual differences that preserve the same operation sequence and object choices do not constitute distinct solutions.

Environment-based identification. We identify distinct solutions using the ALFWorld environment semantics and task-completion conditions. For trajectories verified as successful by the environment, we compare the operation order and objects used to complete the task according to the definition above. This grounds solution identity in executed behavior rather than surface-form differences between action sequences.

Counting distinct verified solutions. For each of the 111 evaluation problems, we group sampled successful trajectories by their operation order and object choices, counting each group once. We compute the proportion of problems with at least two distinct verified solutions over all 111 problems using a matched sampling budget across methods. These counts characterize the solution diversity observed within the evaluation budget (Figure 4(b)).

A.2 Definition and Semantic Filtering of Disagreement Actions

Identifying disagreement actions. We first collect student trajectories on ALFWorld. At each interaction step k, let h_k denote the task instruction and interaction history preceding the student action a_k^{S}. We condition the teacher on the same context and sample its next action, $a_k^{\mathrm{T}} \sim \pi_T(\cdot \mid h_k)$. We identify the step as a teacher–student disagreement when $a_k^{\mathrm{S}} \neq a_k^{\mathrm{T}}$. Thus, the two actions are compared at the same decision point in the original student trajectory.

Trajectory-based candidate pools. Let $z(\tau) \in \{0, 1\}$ indicate whether the original student trajectory τ successfully completes the task, as determined by the ALFWorld environment. For each disagreement at step k, we assign an initial candidate type:

$$\widetilde{c}(\tau, k) = \begin{cases} \text{multi-solution}, & z(\tau) = 1, \\ \text{conflict}, & z(\tau) = 0. \end{cases} \tag{10}$$

Disagreements from successful trajectories therefore form the multi-solution candidate pool, whereas those from failed trajectories form the conflict candidate pool. We then resolve the candidate type at the individual-action level through environment search and verification.

Automated environment search and verification. We obtain the final groups through automated environment search and verification. For every candidate disagreement, we restore the ALFWorld simulator to the pre-action state s_k and search for task-completing paths from each candidate action. The search starts from the environment goal predicates, recursively expands their prerequisite predicates and actions, and enumerates executable paths under the ALFWorld transition rules and the remaining episode budget. We then execute each candidate path in the simulator and accept it only if ALFWorld's built-in task-success verifier confirms that the resulting state satisfies the task goal. Formally, for an action a at s_k, we define

$$F(s_k, a) = \mathbb{I}\big[\exists\, \mathbf{a}_{k+1:H} \text{ s.t. } \mathrm{Verify}_{\mathrm{env}}\big(s_k, a, \mathbf{a}_{k+1:H}\big) = 1\big]\,, \tag{11}$$

where the continuation consists of executable environment actions and terminates within the episode horizon H. A candidate is retained as a *multi-solution disagreement* when both the student and teacher actions admit verifier-accepted completion paths and the two paths differ in their operation sequence or object choices. A candidate is retained as a *conflict disagreement* when the student action violates a required precondition or admits no verifier-accepted completion path, while the teacher action admits one. The retained candidates form the two analysis groups.

The procedure takes as input the task specification, reconstructed simulator state, environment transition rules, and binary success verifier. The resulting analysis contains 7,980 multi-solution and 7,564 conflict disagreement steps. For Figure 2(b), we uniformly sample 512 steps without replacement from each verified group and compute entropy, KL divergence, and AUC on the balanced sample. Figure 2(c) computes the group shares using all classified steps.

A.3 Estimating the Return Impact of Teacher Actions

Rollout protocol. For each sampled teacher–student disagreement at step k on ALFWorld, let s_k denote the pre-action state and let a_k^{S} and a_k^{T} denote the student and teacher actions, respectively. We construct two evaluation branches from the same environment state and interaction history immediately before the disagreement. In the student branch, we execute a_k^{S} and sample 128 continuations using the student policy. In the teacher branch, we restore the same pre-action state and history, execute a_k^{T}, and sample another 128 continuations using the same student policy. Each continuation starts from the corresponding post-action state and updated interaction history, and the student policy remains fixed throughout the evaluation.

Monte Carlo action-value estimates. Let $R_k^{(n)}(a)$ denote the final environment return of the n-th continuation after executing action a at s_k and following the student policy π_θ thereafter. We estimate the expected return of each candidate action as

$$\widehat{Q}^{\pi_\theta}(s_k, a) = \frac{1}{N}\sum_{n=1}^{N} R_k^{(n)}(a), \qquad N = 128, \tag{12}$$

where $a \in \{a_k^{\mathrm{S}}, a_k^{\mathrm{T}}\}$. Each disagreement step is therefore evaluated with 256 continuations, equally divided between the two actions.

Teacher-minus-student action-value gap. We estimate the return impact of replacing the student action with the teacher action as

$$\widehat{\Delta}_k = \widehat{Q}^{\pi_\theta}(s_k, a_k^{\mathrm{T}}) - \widehat{Q}^{\pi_\theta}(s_k, a_k^{\mathrm{S}}). \tag{13}$$

A positive gap favors the teacher action, whereas a negative gap favors the student action. Both branches use the same student continuation policy, so the comparison measures the change in expected return from replacing the current action while holding the continuation policy fixed.

A.4 Relative Rescaling of the OPD Signal

Since action-level return credit and token-level teacher support can differ in scale, we rescale the token-level teacher support relative to the corresponding action-level advantage before computing the consistency score:

$$A_{k,t}^{\mathrm{OPD}} \leftarrow \mathrm{sign}\big(A_{k,t}^{\mathrm{OPD}}\big)\min\big(|A_{k,t}^{\mathrm{OPD}}|, |A_k^{\mathrm{RL}}|\big)\,, \tag{14}$$

This operation leaves the OPD signal unchanged when its magnitude does not exceed $|A_k^{\mathrm{RL}}|$ and otherwise caps its magnitude at that value. In implementation, the consistency score is computed as $M_{k,t} = A_k^{\mathrm{RL}} A_{k,t}^{\mathrm{OPD}}$. This relative rescaling limits the influence of disproportionately large OPD signals while preserving the sign of the original consistency score.

A.5 Training Algorithm

The complete on-policy update is summarized in Algorithm 1. The action-level return credit is shared across the tokens of an interaction step, while the adaptive OPD weight is computed separately for each token-level OPD term.

Algorithm 1 Training procedure for VALVE.

Require: Student policy π_θ, critic V_ϕ, frozen teacher π_T, gate sharpness β

for each on-policy training batch **do**
  Sample student rollouts $\tau \sim \pi_{\theta_{\mathrm{old}}}$ and record (s_k, a_k, r_k).
  for each interaction step k in τ **do**
    Compute the unnormalized A_k^{RL} with the action-level TD residuals and GAE.
    for each token t in a_k **do**
      Query π_T and compute the raw token-level OPD signal $A_{k,t}^{\mathrm{OPD}} = \log \pi_T(y_{k,t} \mid h_{k,t}) - \log \pi_{\theta_{\mathrm{old}}}(y_{k,t} \mid h_{k,t})$.
      Rescale $A_{k,t}^{\mathrm{OPD}}$ using Eq. 14.
      Compute $M_{k,t} = A_k^{\mathrm{RL}} A_{k,t}^{\mathrm{OPD}}$ and $g_{k,t} = \sigma(\beta\, \mathrm{sg}(M_{k,t}))$.
      Compute the token-level OPD loss $\ell_{k,t}^{\mathrm{OPD}}$.
    end for
  end for
  Masked-whiten the broadcast A_k^{RL} over valid generated tokens for the PPO loss only.
  Update π_θ by minimizing $\mathcal{L}_{\mathrm{PPO}} + \frac{1}{K}\sum_k \frac{1}{L_k}\sum_t g_{k,t}\ell_{k,t}^{\mathrm{OPD}}$.
  Update V_ϕ with the PPO critic objective and set $\theta_{\mathrm{old}} \leftarrow \theta$ for the next batch.
end for

A.6 Benchmark and Evaluation Protocol

We evaluate all methods on the same fixed benchmark splits. Following the controlled setup used in recent agentic RL work built on the same training stack (Feng et al., 2025; Zhang et al., 2025; Wang et al., 2026b), all locally trained methods use environment seed 0 and dataloader seed 1. Validation environments use disjoint seeds beginning at 1000.

ALFWorld. ALFWorld (Shridhar et al., 2021) evaluates text-based household tasks with sparse terminal rewards. We use the in-distribution evaluation split and report success separately for Pick, Look, Clean, Heat, Cool, and Pick2. Episodes allow at most 30 environment steps and retain the previous five interaction turns. The category columns in Table 1 report the success rate within each task category, while Avg is the overall success rate across the 128 validation environments rather than an unweighted average of the six category scores.

WebShop. WebShop (Yao et al., 2023) requires an agent to search a product catalog and purchase an item satisfying a natural-language goal. We use the 1,000-product files with synthetic goals, and report both task score and success rate. Episodes allow at most 15 steps with a two-turn history. Score is the mean environment score, reported on a 0–100 scale, and Acc is the percentage of episodes that fully satisfy the requested product attributes.

Models and teachers. We use Qwen2.5-3B-Instruct, Qwen2.5-7B-Instruct, and Qwen3-1.7B as students. For the main experiments, the Qwen2.5 students use same-scale step-150 GRPO teachers, while Qwen3-1.7B uses a step-150 Qwen3-4B GRPO teacher. Teachers are frozen and queried only on student-visited trajectories. Table 6 summarizes the shared evaluation protocol.

Table 5: **Baseline implementation and evaluation protocol.** Method-specific objectives and hyperparameters follow the corresponding papers; shared controls refer to our common training and evaluation setup.

Methods	Method-specific configuration	Training and evaluation controls
PPO; GRPO	Standard reward-only objectives from the corresponding papers	Student model, benchmark split, interaction budget, training horizon, and evaluation procedure
Fixed OPD; Annealed OPD	Constant or training-dependent OPD coefficient following the cited OPD formulations	Student and teacher checkpoints, reverse-KL objective, rollout source, and evaluation procedure
KL-weighted; Entropy-weighted OPD	Local weighting rule based on teacher–student divergence or student entropy	Student and teacher checkpoints, training horizon, rollout budget, and evaluation procedure
ATOD; KDRL; RLSD; SDAR; AgentOPSD; SERL	Original objectives and author-provided method-specific hyperparameters	Shared model, benchmark, training, rollout, and evaluation settings

Table 6: **Benchmark and evaluation protocols.** All locally evaluated methods use the same split and environment settings within each benchmark.

Benchmark	Evaluation setting	Reported metrics	Interaction budget
ALFWorld	128 in-distribution validation environments	Per-category and overall task success (%)	30 environment steps; five-turn history
WebShop	1,000-product catalog with synthetic goals	Mean task score and exact success rate (%)	15 environment steps; two-turn history

Baseline configurations. Table 5 summarizes the baseline configurations. All locally trained methods follow the shared controls in Tables 6 and 9; method-specific objectives and hyperparameters follow the corresponding papers.

Table 7: **Sensitivity to gate sharpness** β with Qwen2.5-3B at training step 150. Entries are success rates (%); bold marks the best result in each column.

β	ALFWorld	WebShop
6	90.6	**71.8**
8	89.1	68.8
10	**91.4**	71.1
12	88.2	71.1
14	89.1	70.3
Spread	3.2	3.0

A.7 Sensitivity to β.

The hyperparameter β controls the sharpness of the sigmoid gate in Eq. 8, with larger values amplifying the contrast between positive and negative consistency scores. We evaluate $\beta \in \{6, 8, 10, 12, 14\}$ using Qwen2.5-3B on ALFWorld and WebShop, measuring success rates at training step 150. As shown in Table 7, performance remains relatively stable across the tested values, with spreads of 3.2 and 3.0 percentage points on ALFWorld and WebShop, respectively. Although the best setting differs across tasks, $\beta = 10$ achieves the highest success rate on ALFWorld and remains within 0.7 percentage points of the best result on WebShop. These results suggest limited sensitivity to β within the tested range and support $\beta = 10$ as a balanced choice across the two tasks.

A.8 Implementation Details of Ablation Variants

We detail the four ablation variants reported in Table 3. Each variant modifies either the consistency score or the adaptive OPD weight. Unless otherwise specified, the weight is computed as $g_{k,t} = \sigma(\beta\, \mathrm{sg}(M_{k,t}))$.

Without action-level return credit (trajectory-level reward). We replace the action-level advantage A_k^{RL} in the consistency score with the final trajectory reward used in GRPO:

$$M_{k,t} = R(\tau) A_{k,t}^{\mathrm{OPD}} \tag{15}$$

where $R(\tau)$ denotes the terminal reward of the sampled trajectory τ. The same reward is broadcast to all interaction steps and tokens in the trajectory, replacing action-level return credit with trajectory-level feedback.

Without adaptive weighting. We fix the adaptive OPD weight to

$$g_{k,t} = 1 \tag{16}$$

for all tokens throughout training. Each token-level OPD loss therefore receives a constant unit weight, removing adaptive modulation based on teacher–return consistency.

Without return credit in the OPD weight. We remove the RL advantage from the consistency score:

$$M_{k,t} = A_{k,t}^{\mathrm{OPD}}. \tag{17}$$

The adaptive OPD weight is thus determined solely by the teacher's relative support for the sampled token, without incorporating action-level return credit.

Without teacher support in the OPD weight. We remove the teacher-support signal from the consistency score:

$$M_{k,t} = A_k^{\mathrm{RL}}. \tag{18}$$

The adaptive OPD weight depends only on the action-level advantage and is shared by all tokens within the same action. The fixed teacher still provides supervision through the base OPD loss; this ablation removes its support signal only from the weighting mechanism.

A.9 Compute-cost accounting

All methods in Table 2 are run on one node with eight NVIDIA H800 GPUs under the same software and hardware configuration. We compute GPU-hours as elapsed wall-clock hours multiplied by the eight allocated GPUs. For each target, defined as 25/50/100% of the final step-150 GRPO teacher success rate, we report the cumulative cost at the first evaluated checkpoint that reaches or exceeds the target; if one checkpoint crosses multiple targets, those targets share the same cost. The measurement includes student optimization, environment interaction, rollout generation, frozen-teacher inference, critic updates, and periodic evaluation, as applicable to each method. It excludes queueing, one-time data preparation, and construction of the frozen GRPO teacher checkpoints, which are shared starting assets for the teacher-guided methods. Accordingly, this comparison measures the downstream student-training cost given an available teacher, rather than the total cost of training both teacher and student from scratch.

A.10 Implementation and Evaluation Details

Hyperparameters. Following the common setup in recent agentic RL work (Feng et al., 2025; Lu et al., 2026; Wang et al., 2026a; Zhang et al., 2025; Wang et al., 2026b), we use the same optimizer, PPO, rollout, runtime, and evaluation settings for all locally trained methods. For each baseline, method-specific hyperparameters follow the values provided by its authors; all remaining settings are held fixed. The VALVE-specific settings are action-level GAE with $\gamma = 0.95$ and $\lambda_{\mathrm{GAE}} = 1$, gate sharpness $\beta = 10$, and the relative teacher-support rescaling in Eq. 14. The teacher remains frozen and is evaluated on student-generated trajectories; no additional reference-policy KL or curriculum term is used. Table 8 lists the complete shared and method-specific configuration. Table 9 gives the benchmark-specific batch, context, and environment settings. The main-table launchers use a single node with eight NVIDIA H800 GPUs. The trainer runs for 160 steps and the reported checkpoint is evaluated at step 150. Vanilla entries are validation-only.

PPO optimization details. Our PPO implementation follows the `verl-agent` PPO trainer. For the PPO actor loss, we masked-whiten the broadcast action-level GAE over valid generated tokens and denote the result by $\widehat{A}_k^{\mathrm{RL}}$. For a generated token $y_{k,t}$, we use the token-level importance ratio $\rho_{k,t} = \pi_\theta(y_{k,t} \mid h_{k,t}) / \pi_{\theta_{\mathrm{old}}}(y_{k,t} \mid h_{k,t})$ and define

$$\ell_{k,t}^{\mathrm{clip}} = \max\Big(-\rho_{k,t}\widehat{A}_k^{\mathrm{RL}}, -\operatorname{clip}(\rho_{k,t}, 1-\epsilon, 1+\epsilon)\widehat{A}_k^{\mathrm{RL}}\Big), \qquad \epsilon = 0.2. \tag{19}$$

For negative advantages, dual clipping gives the token-level PPO loss

$$\ell_{k,t}^{\mathrm{PPO}} = \begin{cases} \min\Big(-c\widehat{A}_k^{\mathrm{RL}}, \ell_{k,t}^{\mathrm{clip}}\Big), & \widehat{A}_k^{\mathrm{RL}} < 0, \\ \ell_{k,t}^{\mathrm{clip}}, & \widehat{A}_k^{\mathrm{RL}} \geq 0, \end{cases} \qquad c = 3. \tag{20}$$

Let $m_{k,t}$ denote the valid generated-token mask. The PPO actor objective in Eq. 9 is

$$\mathcal{L}_{\mathrm{PPO}} = \frac{\sum_{k,t} m_{k,t}\ell_{k,t}^{\mathrm{PPO}}}{\sum_{k,t} m_{k,t}} - \eta_{\mathcal{H}} \frac{\sum_{k,t} m_{k,t}\mathcal{H}[\pi_\theta(\cdot \mid h_{k,t})]}{\sum_{k,t} m_{k,t}}, \qquad \eta_{\mathcal{H}} = 10^{-3}. \tag{21}$$

Thus, PPO uses token-mean aggregation over valid generated tokens. The adaptive OPD gate and teacher-support rescaling instead use the unnormalized A_k^{RL} defined in Eq. 5. Each rollout batch receives one actor epoch without minibatch shuffling; actor gradients are clipped to norm 1.0. Environment rewards are used without length normalization, with the invalid-action penalty given in Table 9.

The critic is optimized separately from the actor, so no value-loss coefficient enters Eq. 9. It uses one update epoch, token-mean aggregation, gradient clipping at 1.0, and a clipped squared-error value loss with clip range 0.5. Both actor and critic use AdamW with weight decay 0.01 and no learning-rate warmup. No additional reference-policy KL penalty is used in the actor objective. The weighted OPD term is averaged over tokens within each interaction action and then over actions as shown in Eq. 9; this action-balanced reduction prevents longer interaction actions from receiving proportionally larger distillation weight. Before computing $g_{k,t}$, the teacher-support signal is rescaled relative to the corresponding action-level advantage as described in Appendix A.4. Figure 5(b) reports the resulting distribution of adaptive OPD weights throughout training.

Table 8: **Shared training configuration.** Settings are fixed across benchmarks unless stated otherwise.

Setting	Value
Models and runtime	
Student models	Qwen2.5-3B/7B-Instruct, Qwen3-1.7B
Teacher models	same-scale Qwen2.5 GRPO; Qwen3-4B for Qwen3-1.7B
Teacher update	frozen; queried on student trajectories
Optimizer / precision	AdamW / bfloat16
Weight decay / LR warmup	0.01 / none
Rollout / hardware	vLLM; 1 node, 8×NVIDIA H800; vLLM memory utilization 0.5
Training sampling	temperature 1.0; top-p 1.0; top-k disabled
Optimization	
Actor / critic learning rate	1×10^{-6} / 1×10^{-5}
PPO clip / dual clip / value clip	$0.2/3.0/0.5$
Actor / critic epochs	$1/1$
Policy loss / minibatch shuffle	vanilla PPO / disabled
PPO loss aggregation / advantage scaling	token mean / masked whitening
Entropy coefficient / gradient clip	1×10^{-3} / 1.0
λ_{GAE}	1
Discount factor	0.95
Training schedule	160 steps; report step 150
Seed configuration	dataloader 1; training environment 0; validation environments beginning at 1000
VALVE-*specific settings*	
Supervision granularity	action-level return credit; token-level teacher support and OPD weight
Gate return-credit input	unnormalized action-level GAE
Teacher-support rescaling	sign-preserving cap at $\lvert A_k^{\mathrm{RL}} \rvert$
Adaptive gate	sigmoid; $\beta = 10$; $g_{k,t} \in (0, 1)$
Additional KL / curriculum terms	disabled

Table 9: **Benchmark-specific settings.** For prompt batch × rollouts, values before and after the slash apply to GRPO-based methods and to PPO/VALVE, respectively; all other settings are shared across methods.

Setting	ALFWorld	WebShop
Prompt batch × rollouts	16×8 / 128×1	16×8 / 128×1
Validation environments / batch	128	128
Prompt / response length	$2048/512$	$6000/512$
Actor mini-batch / micro-batch	$256/32$	$64/8$
Critic micro-batch	16	8
Tensor parallel size	2	2
Invalid-action penalty	0.1	0.1
Evaluation temperature	0.4	0.4
Environment options	in-distribution	small catalog; synthetic goals