PR-OPD: PRIVILEGED REPRESENTATION ON-POLICY SELF-DISTILLATION FOR AGENTIC REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Language-model agents are usually trained by reinforcement learning from one reward per episode, and privileged self-distillation enriches it by letting the same policy, given a skill, teach its skill-free self through token probabilities. However, we identify two phenomena that question this channel. **Invisible Advantage**: a skill in context lifts WebShop success from 42.2% to 56.2%, yet changes the probabilities of fewer than a quarter of the sampled tokens. **Much to Align**: a skill changes the hidden states of over 80% of response tokens, in a way that linear probes can trace back to the specific skill. To exploit this, we propose Privileged Representation On-policy Self-Distillation (PR-OPD). After a GRPO warm start, the policy writes a hindsight skill for each trajectory, rereads its own responses with that skill as a stop-gradient teacher, and aligns its projected hidden states to the teacher's at every layer alongside the reward objective, with no external skill library, separate teacher, or inference overhead. On ALFWorld and WebShop with two backbones, PR-OPD achieves the best overall results in every setting, improving over GRPO by up to 4.7 points in ALFWorld success and 14.0 points in WebShop accuracy. Code is available at `https://anonymous.4open.science/r/PR-OPD-A666`.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has emerged as a promising approach for training language-model agents to solve multi-turn tasks (Yao et al., 2023; Shao et al., 2024; Wang et al., 2025; Feng et al., 2026; Jin et al., 2025). In many such tasks, the policy receives only a terminal trajectory-level reward: one number per episode, assigned after a sequence of decisions and generated tokens. Recent methods supplement this coarse feedback with on-policy distillation (OPD), which provides dense token-level guidance on student-generated trajectories (Hinton et al., 2015; Agarwal et al., 2024; Gu et al., 2024; Lu & Lab, 2025; Lu et al., 2026a). In these self-distillation methods, the same model serves as a privileged teacher when conditioned on additional information (Vapnik & Vashist, 2009; Penaloza et al., 2026), such as task knowledge or hindsight feedback (Lu et al., 2026a; Yang et al., 2026c; Wu et al., 2026). The student learns from the teacher's output probabilities without access to this information, aiming to internalize the guidance so that no additional context is needed at inference time.

These methods rest on an implicit assumption: the teacher's output probabilities provide a sufficiently informative signal for transferring privileged guidance to the student. However, these output probabilities reflect what the teacher is likely to generate given the additional information, rather than directly revealing how that information changes its internal reasoning process, and a model's internal representations can carry information that its outputs do not express (Gurnee et al., 2026). For example, hindsight feedback may help the teacher identify an overlooked task constraint or an earlier mistake, while producing little change in the probability of a particular sampled token. In such cases, the token-level probability difference provides a limited signal for transferring the teacher's updated understanding to the student, consistent with recent analyses of the sparsity and dynamics of on-policy distillation signals (Li et al., 2026; Yu et al., 2026; Shen et al., 2026). This motivates us to first examine whether privileged information substantially changes the teacher's token probabilities, and then investigate how it affects the teacher's internal processing.

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.

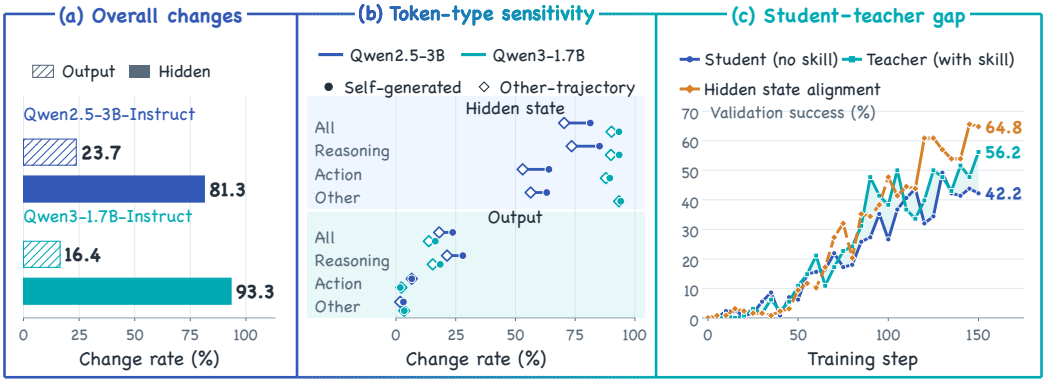

Figure 1: **Two phenomena in privileged self-distillation. (a,b)** Fraction of tokens whose hidden states or sampled-token probabilities change when the policy re-reads its own responses with its self-generated skill, on ALFWorld at GRPO@100. **(c)** In-training WebShop validation success on Qwen2.5-3B-Instruct of the student (no skill), the teacher (skill in context), and a student trained by directly aligning to its teacher's hidden states after branching from GRPO at step 100.

To investigate, we take the natural self-evolving variant of this recipe, in which the policy writes a hindsight skill for each of its own completed trajectories, and compare two forward passes of the same policy over the same sampled responses, one with ordinary context and one with the skill added, as shown in Figure 1. We also let the policy solve tasks with and without a skill in context, to see whether the skill makes it a better actor. The skill turns out to make the teacher act better, yet it leaves most of its trace in the model's hidden states rather than in its next-token probabilities. Our findings highlight two key phenomena:

❶ **Invisible Advantage:** ***the skill makes the teacher act better, yet barely changes its token distribution.*** As shown in Figure 1(c), a skill in context makes the policy a better actor, lifting WebShop success from 42.2% to 56.2%. Yet Figure 1(a,b) shows that adding the skill leaves the teacher's next-token distribution almost where the student's was: fewer than a quarter of response tokens change their probability, and at action tokens, where decisions are made, almost none do. The advantage is thus real but nearly invisible in the output channel. An objective that distills token probabilities can therefore transfer only a thin slice of what the skill gives the teacher.

❷ **Much to Align:** ***the skill reshapes the teacher's hidden states, and aligning with them passes the advantage on.*** As shown in Figure 1(a,b), the same two passes differ in the hidden states of more than 80% of response tokens, including most action tokens where the output channel is silent. The change is not noise: linear probes on hidden states can tell a trajectory's own skill from another trajectory's, while probes on output features stay at chance. Figure 1(c) further shows that this signal is learnable: a student that directly aligns with its teacher's hidden states, with no added parameters, ends up even above the teacher, whereas the student that acts without the skill stays far behind. What the skill adds lives in how the teacher represents the student's responses, not in its output tokens.

Together, these findings show that the advantage a skill gives the teacher barely reaches its token probabilities, while its hidden states carry it. This leads to a central question: ***Can the student acquire the advantage that hindsight gives its privileged self, even though this advantage barely shows in the teacher's tokens?***

To answer this question, we propose Privileged Representation On-policy Self-Distillation (**PR-OPD**). The key idea is to transfer the teacher's advantage through the channel that actually carries it: instead of imitating the privileged teacher's token probabilities, the student aligns with the teacher's hidden states on the same responses. Specifically, PR-OPD first trains the policy with GRPO until it can act and analyze its own experience. It then keeps optimizing the reward and, for each completed trajectory, has the policy write a hindsight skill and re-read its own sampled responses with that skill as a stop-gradient teacher. The student's hidden states are passed through a lightweight per-layer MLP projector and aligned to the teacher's across all layers. Student, hindsight analyzer, and

teacher are three roles of one set of weights, and the projector is the only added module, so PR-OPD needs no external skill library, no separate teacher model, and nothing extra at inference.

Our main contributions are summarized as follows:

- We identify two phenomena in privileged self-distillation, **Invisible Advantage** and **Much to Align**: a skill makes the teacher act better yet barely changes its token distribution, and the advantage is carried by its hidden states instead.
- We propose **PR-OPD**, in which the policy learns from its own hindsight by aligning its projected hidden states to those of its privileged self at every layer.
- Experiments on ALFWorld and WebShop with two backbones show that PR-OPD achieves the best average success and accuracy in every setting, improving over GRPO by up to $+4.7$ points on ALFWorld and $+11.7$ points on WebShop with Qwen2.5-3B-Instruct.

2 Related Work

2.1 Agentic RL with Privileged Self-Distillation

Reinforcement learning is the standard way to train language-model agents over multi-turn interaction, with GRPO (Shao et al., 2024) and its multi-turn extensions (Wang et al., 2025; Feng et al., 2026) learning from a sparse terminal reward. To densify this signal, one line places past experience in the agent's context, such as Reflexion (Shinn et al., 2023), ExpeL (Zhao et al., 2024), and SkillRL (Xia et al., 2026), and further work trains agents with reinforcement learning to use, internalize, or evolve such skills (Lu et al., 2026b; Shi et al., 2026; Yao et al., 2026). Another line turns it into training-time supervision through privileged on-policy self-distillation, which builds on learning using privileged information (Vapnik & Vashist, 2009; Lopez-Paz et al., 2015) and privileged teachers in robot learning (Pinto et al., 2018; Chen et al., 2020), as well as context distillation and on-policy distillation for language models (Snell et al., 2022; Agarwal et al., 2024; Zhao et al., 2026): the same policy, conditioned on a skill it will not see at test time, serves as a teacher whose token log-probabilities are distilled back into the student, as in SDAR (Lu et al., 2026a), PCSD (Lv et al., 2026), OPID (Yang et al., 2026c), SEED (Wu et al., 2026), and related variants (Penaloza et al., 2026; Wang et al., 2026b; Pan et al., 2026; Tu et al., 2026). However, all of these read the teacher after the language-model head, so whatever the skill changes inside the teacher reaches the student only through one scalar per token. To address this, we propose PR-OPD, which aligns the student's hidden states to those of its privileged self.

2.2 Representation-Level Distillation

Beyond matching output distributions (Hinton et al., 2015; Gu et al., 2024), intermediate representations have long served as distillation targets, from hidden-layer hints in FitNets (Romero et al., 2015) to multi-layer transfer in Patient-KD (Sun et al., 2019), TinyBERT (Jiao et al., 2020), and MiniLM (Wang et al., 2020), all for compressing a fixed, stronger teacher into a smaller student. More recently, OPRD (Yang et al., 2026b) aligns hidden states on student-generated responses for mathematical reasoning, and LastOPD (Yang et al., 2026a) shows that with a separate, larger teacher this signal collapses late in training because depth-paired layers play different roles in the two models, and therefore restricts alignment to the last layer during a brief crossfade into token-level distillation. However, these methods assume a teacher that is separate from the student and fixed during training, whereas in self-distillation for agents the teacher is the policy itself, so its layers correspond one-to-one with the student's, its hindsight is written by that policy, and both change at every update. None of them asks the question raised by our diagnostics: whether an advantage that barely shows in the teacher's tokens can still be passed on through its hidden states. This motivates PR-OPD, which treats the policy under its own hindsight as a stop-gradient teacher and aligns the student's hidden states to it across all layers inside the RL loop, so that the student learns from the teacher's hidden states rather than imitating its token probabilities.

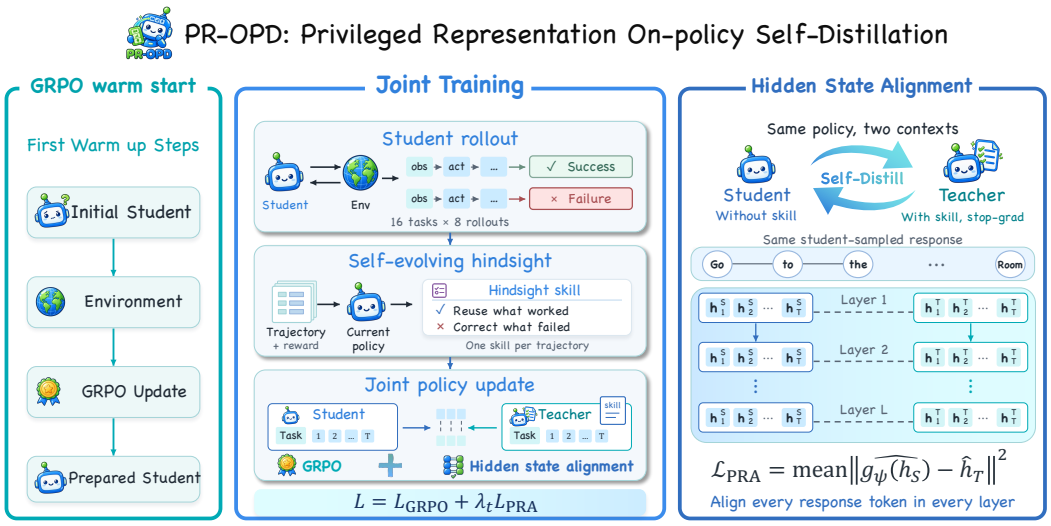

Figure 2: **Overview of PR-OPD. Left:** a GRPO warm start first makes the policy able to act and to analyze its own experience. **Center:** in each later iteration, the policy collects trajectories, writes one hindsight skill per completed trajectory, and updates with the GRPO loss plus a hidden-state alignment loss. **Right:** the same policy re-reads the student-sampled response twice, without and with the skill. The skill-conditioned pass is the stop-gradient teacher, and every response token in every layer of the student is passed through a lightweight MLP projector and pulled toward it. Skills and teacher states are recomputed as the policy evolves, and neither is needed at inference.

3 METHOD

3.1 PROBLEM DEFINITION

Consider a policy π_θ that solves a task q over multiple turns: for each of G sampled trajectories, $\tau_i = \{(x_{i,u}, y_{i,u})\}_{u=1}^{U_i}$ records the context $x_{i,u}$ and the sampled response $y_{i,u}$ at every turn u, and the whole trajectory receives one terminal reward R_i. This context holds the task, observations, and interaction history, and is all the policy ever sees at inference. During training, however, the same policy can take on two further roles once a trajectory is complete: as the *analyzer* it reads (τ_i, R_i) and writes a hindsight skill s_i, and as the *teacher* it re-reads the student's own responses $y_{i,u}$ with s_i prepended to the context. The catch, as Figure 1 shows, is that the skill makes this teacher act better yet barely changes its token probabilities (Invisible Advantage), while its hidden states change broadly and carry the advantage (Much to Align). Our goal is therefore to pass this advantage to the student through the teacher's hidden states rather than its token probabilities, without giving up reward learning or adding anything at inference.

3.2 OVERVIEW

PR-OPD runs in two stages (Figure 2). Stage I trains the policy with GRPO alone for N updates. This warm start is needed because the analyzer is the policy itself, and it has to learn the task before it can explain its own mistakes. Stage II keeps the reward objective and adds representation supervision. Each iteration collects trajectories without skills, has the current policy write one skill per completed trajectory, and updates the policy with the GRPO loss plus an alignment loss toward the teacher's hidden states. Student, analyzer, and teacher share one set of weights throughout, and the only added module is a lightweight per-layer MLP projector on the student side, trained jointly with the policy and discarded at inference.

3.3 STAGE I: GRPO WARM START

For each task q, the rollout policy $\pi_{\theta_{\text{old}}}$ samples G trajectories with terminal rewards $\mathbf{R} = \{R_i\}_{i=1}^{G}$, and GRPO (Shao et al., 2024) gives every response token of trajectory i the group-relative advantage

$A_i = (R_i - \mathrm{mean}(\mathbf{R}))/\,\mathrm{std}(\mathbf{R})$. With the ratio $r_{i,k}(\theta) = \pi_\theta(y_{i,k} \mid \cdot)/\pi_{\theta_{\mathrm{old}}}(y_{i,k} \mid \cdot)$ for the k-th of the K_i response tokens in τ_i, the policy minimizes

$$\mathcal{L}_{\mathrm{GRPO}}(\theta) = -\frac{1}{G}\sum_{i=1}^{G}\frac{1}{K_i}\sum_{k=1}^{K_i}\min\Big(r_{i,k}(\theta)A_i,\ \mathrm{clip}\big(r_{i,k}(\theta), 1-\epsilon, 1+\epsilon\big)A_i\Big). \tag{1}$$

Stage I optimizes this objective alone for N updates, without skills or teacher passes, so that the hindsight the policy later writes about its own trajectories is worth learning from.

3.4 Self-Generated Hindsight Skill

Generate one skill per trajectory. At Stage-II iteration j, the rollout policy π_{θ_j} collects fresh trajectories without privileged information. After an episode finishes, we serialize its interaction record and outcome into a compact text z_i and prompt the same policy to write a hindsight skill:

$$z_i = \mathrm{Serialize}(\tau_i, R_i), \qquad s_i \sim \pi_{\theta_j}(\,\cdot \mid z_i, I_{\mathrm{hind}}), \tag{2}$$

where the instruction I_{hind} asks for a short list of actionable rules. For a successful trajectory, these rules describe a reusable workflow and the observations that decided it, and for a failed one, they name the critical mistakes and how to correct them. Skill generation is non-differentiable, and the analyzer receives no separate SFT stage. Skills are rewritten from every new rollout batch rather than retrieved from a library.

Revisit the same responses with hindsight. For every turn of trajectory i, we prepend its skill to the turn's context x with a fixed template:

$$\widetilde{x} = H(x, s_i). \tag{3}$$

The student processes (x, y) and the teacher processes $(\widetilde{x}, y)$, so both passes see the same response tokens and the teacher generates no replacement actions. The skill is privileged because it comes from the completed episode, including observations and outcomes unavailable at decision time.

3.5 Privileged Representation Alignment

Shared weights and matched positions. Following the representation-level formulation of OPRD (Yang et al., 2026b), we extract hidden states along the student-sampled responses. To keep notation light, we consider one turn with context x, response y, and teacher context $\widetilde{x}$, and drop the trajectory and turn indices. For layer ℓ and response token y_k, define

$$h^{(\ell)}_{S,k} = f^{(\ell)}_\theta(x, y_{<k}), \qquad h^{(\ell)}_{T,k} = f^{(\ell)}_\theta(\widetilde{x}, y_{<k}), \tag{4}$$

where $f^{(\ell)}_\theta$ returns the layer-ℓ hidden state just before token y_k is predicted. The two passes run the same network with the same current weights θ and differ only in their prompts. Layer ℓ of the student is therefore paired with layer ℓ of the teacher, with no layer mapping to choose as in distillation from a different model (Yang et al., 2026a). Because the teacher's prompt is longer by the skill, the same response token sits at a different absolute position in each pass, so we pair states by response token rather than by position. Within Stage-II iteration j, θ starts from θ_j, the weights of the rollout policy that collected the batch, and changes as the batch is optimized.

Align every layer and every response token. Let $\mathcal{M}$ collect the valid response tokens of all turns in a training microbatch. Each student state first passes through a per-layer MLP projector $g^{(\ell)}_\psi : \mathbb{R}^d \to \mathbb{R}^d$ with one 512-dimensional hidden layer, giving $p^{(\ell)}_{S,k} = g^{(\ell)}_\psi(h^{(\ell)}_{S,k})$. Since the student's raw states also feed its own language-model head, the projector lets it follow the teacher without forcing those states to copy ones computed under a longer, skill-bearing context. With $\widehat{v} = v/\max(\|v\|_2, \varepsilon)$, we minimize the privileged representation alignment (PRA) loss

$$\mathcal{L}_{\mathrm{PRA}}(\theta, \psi) = \frac{1}{L|\mathcal{M}|}\sum_{\ell=1}^{L}\sum_{k\in\mathcal{M}}\Big\|\widehat{p}^{(\ell)}_{S,k} - \mathrm{sg}\Big[\widehat{h}^{(\ell)}_{T,k}\Big]\Big\|_2^2, \tag{5}$$

where L counts all Transformer layers excluding the embedding output and sg stops gradients through the teacher. $\mathcal{M}$ includes reasoning and action tokens in the response but not prompt or padding. Whenever both norms exceed ε, each normalized distance equals $2 - 2\cos(p_S, h_T)$ (Appendix A), so alignment concerns direction rather than magnitude. Because the teacher shares the current weights, its states are recomputed at every update and the target moves with the policy.

Table 1: Step-150 results on ALFWorld and WebShop. ALFWorld reports success rates (%), and WebShop reports accuracy (Acc, %) and score. Within each backbone, **bold** is the best and underline the second best.

Method	ALFWorld							WebShop	
	Pick	Look	Clean	Heat	Cool	Pick2	All	Acc	Score
Qwen2.5-3B-Instruct									
Vanilla	44.4	28.6	25.0	7.7	14.3	4.3	23.4	0.8	5.4
Skill-Prompt	44.4	63.6	29.2	5.3	4.8	5.9	25.8	0.8	3.1
GRPO	88.9	**72.7**	50.0	52.6	**61.9**	**58.8**	66.4	55.5	70.4
SDAR	88.9	63.6	66.7	**63.2**	**61.9**	47.1	68.8	56.3	74.7
OPSD	55.6	35.7	11.1	0.0	14.3	20.0	23.4	1.6	6.0
Skill-SD	91.9	**72.7**	**70.8**	55.6	57.1	47.1	69.5	59.4	75.6
PR-OPD (ours)	**97.2**	54.5	**70.8**	57.9	**61.9**	52.9	**71.1**	**67.2**	**83.7**
Qwen3-1.7B-Instruct									
Vanilla	18.9	42.9	0.0	0.0	4.8	0.0	8.6	3.1	44.7
Skill-Prompt	5.6	**63.6**	41.7	5.3	0.0	5.9	16.4	2.3	25.7
GRPO	58.3	45.5	50.0	36.8	23.8	**35.3**	43.8	43.0	66.4
SDAR	58.3	**63.6**	45.8	47.4	23.8	**35.3**	46.1	45.3	66.6
OPSD	25.0	54.5	8.3	0.0	4.8	5.9	14.8	6.3	44.2
Skill-SD	55.6	45.5	**58.3**	**52.6**	**28.6**	17.6	45.3	54.7	73.7
PR-OPD (ours)	**69.4**	**63.6**	37.5	**52.6**	**28.6**	17.6	**46.9**	**57.0**	**73.8**

3.6 Overall Objective

The two stages share one objective, in which the alignment loss is added to the RL loss only after the N warm-start updates:

$$\mathcal{L}(\theta,\psi;t)=\underbrace{\mathcal{L}_{\mathrm{GRPO}}(\theta)+\beta\mathcal{L}_{\mathrm{KL}}(\theta)-\eta\mathcal{H}(\pi_\theta)}_{\mathcal{L}_{\mathrm{RL}}(\theta)}+\lambda_t\mathcal{L}_{\mathrm{PRA}}(\theta,\psi),\qquad \lambda_t=\begin{cases}0, & t\le N,\\ \lambda, & t>N,\end{cases} \tag{6}$$

where $\mathcal{L}_{\mathrm{KL}}$ regularizes the student toward a frozen reference policy and $\mathcal{H}$ is the policy entropy. No output-space distillation term is used. Algorithm 1 in Appendix B summarizes one Stage-II iteration. After each update, the new policy acts, writes skills, and serves as teacher in the next iteration, so the experience and its hindsight supervision evolve together.

4 Experiments

4.1 Experimental Settings

Benchmarks. We evaluate on ALFWorld (Shridhar et al., 2020), a text-based household environment, and WebShop (Yao et al., 2022), a web-shopping environment. We report success rates on ALFWorld and task accuracy and graded score on WebShop, with split details in Appendix C.

Protocol. We use Qwen2.5-3B-Instruct (Qwen Team et al., 2024) and Qwen3-1.7B-Instruct (Yang et al., 2025) as backbones. Every method trains for 150 updates, and PR-OPD spends the first N of them on GRPO, with $N = 120$ on WebShop and $N = 100$ on ALFWorld. All methods are evaluated at step 150 on 128 episodes with hyperparameters and hardware in Appendix C.

Baselines. We compare against six baselines that each test one alternative explanation: Vanilla and Skill-Prompt whether a skill helps the untrained policy, GRPO (Shao et al., 2024) whether reinforcement learning alone suffices, SDAR (Lu et al., 2026a) and OPSD (Zhao et al., 2026) whether distilling token probabilities is enough, and Skill-SD (Wang et al., 2026a) whether a stronger token-level objective closes the gap. All skill-based baselines use the hand-written skills of SDAR (Lu et al., 2026a), one per task type.

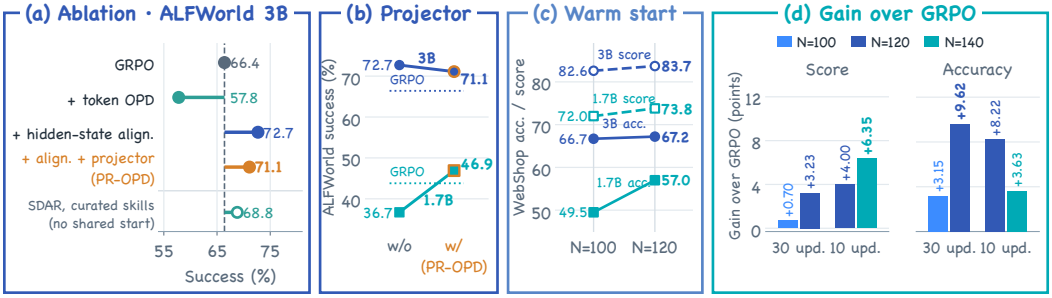

Figure 3: **Ablation and sensitivity study.** **(a)** ALFWorld success of each variant with the 3B backbone, relative to GRPO (dashed line). **(b)** ALFWorld success without and with the projector for both backbones, with GRPO as dotted lines. **(c)** WebShop accuracy (filled) and score (open) of PR-OPD started at $N = 100$ or $N = 120$. **(d)** Gain of direct alignment over GRPO continued from the same checkpoint on WebShop with 1.7B, averaged over six seeds. Numbers are in Appendix D.

4.2 Main Results

We compare PR-OPD against the six baselines on ALFWorld and WebShop under both backbones. Results are in Table 1. Based on these results, we summarize our observations (**Obs.**) as follows:

Obs. ❶: PR-OPD achieves the best aggregate result under every backbone. It attains the highest ALFWorld overall success and the highest WebShop accuracy and score under both backbones. On Qwen2.5-3B-Instruct, it improves over GRPO by 4.7 points on ALFWorld and by 11.7 points in WebShop accuracy. It also ranks first or second on at least four of the six ALFWorld task types under each backbone. These results answer the central question of Section 1: the student can acquire the advantage that hindsight gives its privileged self, without seeing the skill at inference.

Obs. ❷: Aligning hidden states with self-written skills beats distilling tokens with curated ones. SDAR, OPSD, and Skill-SD all distill the teacher's token probabilities under skills from the same curated library, matched by task type, whereas PR-OPD aligns hidden states under a skill it writes for each trajectory. Against the strongest of them, Skill-SD, PR-OPD gains 7.8 points in WebShop accuracy and 8.1 in score with the 3B backbone, and it also leads on ALFWorld under both backbones. The other token-level methods gain little: SDAR improves over GRPO by at most 2.4 points on ALFWorld, and OPSD falls to roughly the level of the untrained policy. This matches the **Much to Align** of Phenomenon ❷: the skill's advantage is carried by the teacher's hidden states, so aligning them transfers more than distilling its tokens.

4.3 Ablation and Sensitivity Study

We examine each design choice of PR-OPD by changing one at a time, named as in Figure 3. Ablations use ALFWorld, with the projector compared on both backbones, and the warm-start study uses WebShop with the 1.7B backbone. (**1**) **+ token OPD** distills the teacher's token probabilities under the same self-generated skills. (**2**) **+ hidden-state align.** aligns hidden states directly, without the projector. (**3**) **+ align. + projector** is the full PR-OPD. (**4**) **SDAR** distills token probabilities under curated skills and trains from scratch. (**5**) N starts representation supervision after 100, 120, or 140 GRPO updates, and for direct alignment we pair each start with GRPO continued from the same checkpoint over six evaluation seeds. Variants (1) to (3) branch from the GRPO checkpoint at step 100, and all runs end at step 150. From Figure 3, we observe:

Obs. ❸: Reading the teacher's hidden states, not its tokens, drives the gain. From the same warm start and the same self-generated skills, token-level OPD falls to 57.8%, 8.6 points below GRPO, whereas hidden-state alignment rises to 72.7%, 6.3 points above (Figure 3a). The two runs differ only in how the teacher is read, yet they end 14.9 points apart. SDAR, which reads tokens under curated skills, gains only 2.4 points. This is the **Invisible Advantage** of Phenomenon ❶ under controlled conditions: the teacher holds the advantage, but its tokens cannot pass it on.

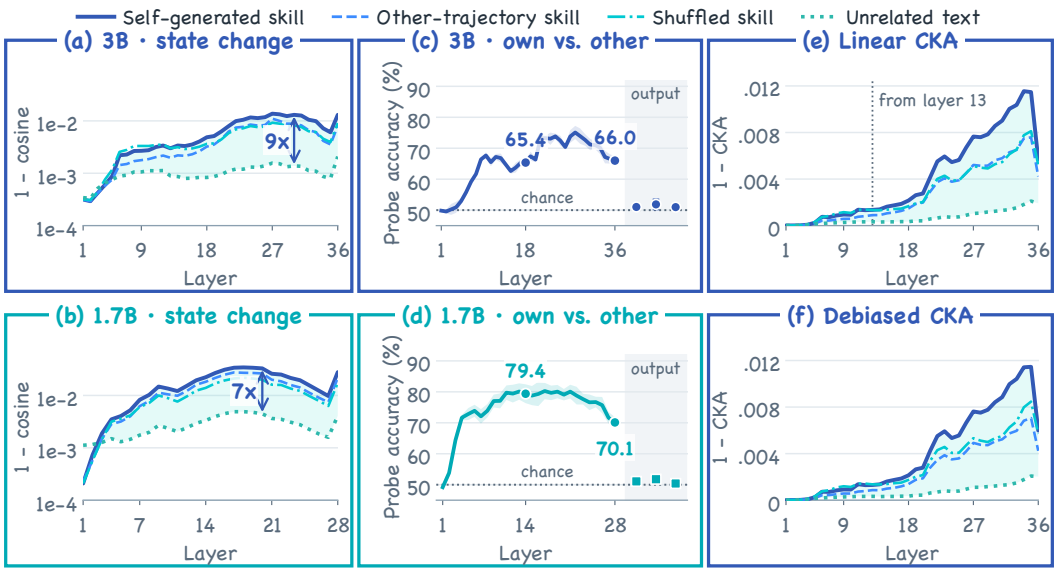

Figure 4: **How the skill acts on the teacher's hidden states across depth.** Columns show three analyses at fixed checkpoints on ALFWorld, with 3B on top and 1.7B below in (a–d). **(a,b)** Mean cosine distance from the skill-free context at each layer at GRPO@100, on a logarithmic axis. **(c,d)** Probe accuracy for the trajectory's own versus another trajectory's skill at each layer, with output features at the right. **(e,f)** Same-layer $1 - \mathrm{CKA}$ of the initial 3B policy under linear and debiased CKA. In (a,b,e,f), shaded bands mark the gap between the own skill and unrelated text.

Obs. ❹: The projector is nearly free on the larger backbone and essential on the smaller one. With the 3B backbone, the projector moves success from 72.7% to 71.1%. With the 1.7B backbone, where direct alignment falls to 36.7%, below GRPO, the projector lifts ALFWorld success to 46.9%, the best result in Table 1 (Figure 3b). A smaller policy thus benefits from a learned buffer between its own states and the teacher's, which is why PR-OPD includes the projector by default (Appendix H).

Obs. ❺: The starting point matters more than the number of supervised updates. Starting at $N = 120$ instead of $N = 100$ raises the WebShop accuracy of PR-OPD from 49.5% to 57.0% with 20 fewer supervised updates (Figure 3c). Paired against GRPO continued from the same checkpoint over six evaluation seeds, direct alignment started at step 120 gains 9.62 points after 30 updates and 8.22 after 10, against 3.15 from step 100 and 3.63 from step 140 (Figure 3d). The policy must first learn the task before its own hindsight is worth reading, hence the GRPO warm start.

4.4 General Analysis

We next examine where the skill acts inside the teacher and what it carries. At fixed checkpoints on ALFWorld, we fix weights and responses and vary the context among four conditions: the trajectory's own skill, another trajectory's skill, a token-shuffled skill, and length-matched unrelated text. We measure how much each context changes hidden states at every layer, probe (Alain & Bengio, 2016; Belinkov, 2022) whether hidden or output features tell the contexts apart, and compare hidden-state structure with linear CKA (Kornblith et al., 2019). From Figures 4 and 5, we observe:

Obs. ❻: Hindsight reshapes hidden states across the whole network, far beyond what unrelated text does. In both backbones, the hidden-state change grows from early to middle layers and stays high through the later layers, rather than concentrating next to the output head (Figure 4a,b). All three skill contexts move the states well above length-matched unrelated text, by up to nine times for 3B and seven times for 1.7B, and the trajectory's own skill moves them the most. The change is therefore specific to the skill rather than to a longer prompt.

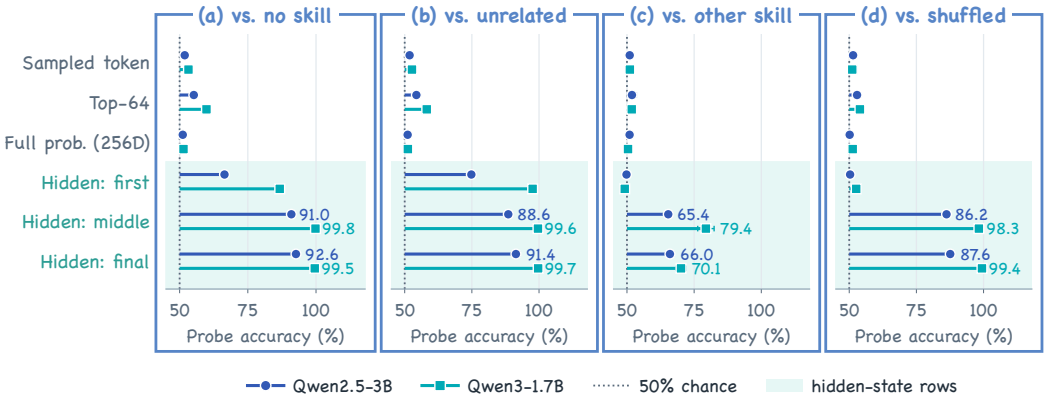

Figure 5: **Probing the hindsight context at selected layers.** Linear-probe accuracy at GRPO@100 on ALFWorld for distinguishing a trajectory's own skill from (a) no skill, (b) unrelated text, (c) another trajectory's skill, and (d) a token-shuffled skill. The first three rows are output features read after the language-model head, and the shaded rows are hidden states.

Obs. ❼: Hidden states carry which skill was given, while output features do not. Linear probes on middle-layer hidden states separate a trajectory's own skill from every alternative in Figure 5, exceeding 88% against no skill or unrelated text, and reaching 65.4% and 79.4% against another trajectory's skill and 86.2% and 98.3% against a token-shuffled copy for 3B and 1.7B (Appendix E). The three output features stay within 50.2–54.0% on the last two distinctions, close to chance. The information the skill adds is thus readable from the teacher's hidden states, not its outputs.

Obs. ❽: The skill information peaks in the middle layers and persists to the output. Probing every layer for own versus other-trajectory skills, accuracy starts near chance at the first layer, rises through the early layers, and stays well above the output features through the rest of the network (Figure 4c,d and Appendix F). For 1.7B, accuracy peaks at 79.4% in the middle layers and drops to 70.1% at the final layer, and for 3B the final layer also sits below the intermediate peak, while all three output features stay at 50–52%. Aligning only the last layer as LastOPD does (Yang et al., 2026a) would therefore miss where this information is clearest, so PR-OPD aligns every layer.

Obs. ❾: The skill makes targeted adjustments to the structure of the hidden states. Unlike cosine distance, linear CKA compares how response positions relate to each other within a layer, so it captures structure rather than the direction of each vector. On the initial 3B policy, before any warm-start training, the trajectory's own skill leaves the overall structure largely intact, with same-layer CKA averaging 0.996, yet from layer 13 onward it changes the structure more than all three controls (Figure 4e and Appendix G). Averaged across layers, the own skill changes the structure 1.5 times as much as another trajectory's skill and six times as much as unrelated text, and the debiased estimate gives the same ordering (Figure 4f). The skill therefore reshapes the representation in a targeted, depth-dependent way rather than overwriting it, even before any training. These analyses place the skill's advantage in the teacher's hidden states at every layer, exactly what PR-OPD aligns.

5 Conclusion

This paper studies two phenomena of privileged self-distillation for language-model agents: a skill makes the teacher act better yet barely changes its token probabilities, and the advantage it brings is carried instead by the teacher's hidden states. We therefore introduce PR-OPD, which lets the policy write a hindsight skill for each trajectory and, after a GRPO warm start, align its projected hidden states to those of its privileged self at every layer. On ALFWorld and WebShop with two backbones, PR-OPD achieves the best overall results in every setting, and our analyses locate the skill's effect in the teacher's hidden states across all layers rather than in its outputs. What a policy learns from its own hindsight is thus better read from what its privileged self represents than from what it says.

AI USE STATEMENT

In this work, we used generative AI tools to polish the writing, to draft some of the plotting scripts, and to help debug and correct parts of the code. We did not use generative AI tools for research ideation, experimental design, or the literature survey. All AI-assisted text was read and verified by the authors, all AI-assisted code was reviewed and tested by the authors before any reported run, and every number that appears in a figure or table was checked against the experiment logs. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

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

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by cheating. In *Conference on robot learning*, pp. 66–75. PMLR, 2020.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. *Advances in Neural Information Processing Systems*, 38:46375–46408, 2026.

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

Wes Gurnee, Nicholas Sofroniew, Adam Pearce, Mateusz Piotrowski, Isaac Kauvar, Runjin Chen, Anna Soligo, Paul Bogdan, Euan Ong, Rowan Wang, et al. Verbalizable representations form a global workspace in language models. *arXiv preprint arXiv:2607.15495*, 2026.

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

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. In *Findings of the association for computational linguistics: EMNLP 2020*, pp. 4163–4174, 2020.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, 2019.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, 2023.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huanang Gao, Wenkai Yang, Zhiyuan Liu, et al. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026.

David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying distillation and privileged information. *arXiv preprint arXiv:1511.03643*, 2015.

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

Zhengxi Lu, Zhiyuan Yao, Zhuowen Han, Zi-Han Wang, Jinyang Wu, Qi Gu, Xunliang Cai, Weiming Lu, Jun Xiao, Yueting Zhuang, et al. Self-distilled agentic reinforcement learning. *arXiv preprint arXiv:2605.15155*, 2026a.

Zhengxi Lu, Zhiyuan Yao, Jinyang Wu, Chengcheng Han, Qi Gu, Xunliang Cai, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. Skill0: In-context agentic reinforcement learning for skill internalization. *arXiv preprint arXiv:2604.02268*, 2026b.

Chunji Lv, Yangguang Wei, Junlin Liu, Yang Gao, Ming Liu, Xinming Wang, Jinyang Wu, Guoren Wang, and Changsheng Li. Pcsd: Persistent consistency for self-distillation in agentic reinforcement learning. *arXiv preprint arXiv:2608.01837*, 2026.

Tianjun Pan, Yuan Li, Hongda Wang, Linbo Jin, Mengfei Song, Lei Gao, Qiming Shi, Shaokang Fu, Jiarong Zhao, Chengyu Wang, et al. Bidirectional context self-distillation for reinforcement learning of skill-based llm agents. *arXiv preprint arXiv:2608.09555*, 2026.

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

Lerrel Pinto, Marcin Andrychowicz, Peter Welinder, Wojciech Zaremba, and Pieter Abbeel. Asymmetric actor critic for image-based robot learning. In *Robotics: Science and Systems*, 2018.

Qwen Team, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets, 2015. URL https://arxiv.org/abs/1412.6550.

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.

Zhennan Shen, Yanshu Li, Qingyu Yin, Chak Tou Leong, Zhilin Wang, Yanxu Chen, Rongduo Han, Sunbowen Lee, and Yi R Fung. On the geometry of on-policy distillation. *arXiv preprint arXiv:2606.07082*, 2026.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, 2025.

Yaorui Shi, Yuxin Chen, Zhengxi Lu, Yuchun Miao, Shugui Liu, Qi Gu, Xunliang Cai, Xiang Wang, and An Zhang. Skill1: Unified evolution of skill-augmented agents via reinforcement learning. *arXiv preprint arXiv:2605.06130*, 2026.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

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.

Charlie Snell, Dan Klein, and Ruiqi Zhong. Learning by distilling context. *arXiv preprint arXiv:2209.15189*, 2022.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for bert model compression. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 4323–4332, 2019.

Songjun Tu, Chengdong Xu, Qichao Zhang, Yiwen Ma, Yaocheng Zhang, Linjing Li, Dong Li, Xiangyuan Lan, and Dongbin Zhao. Ucob: Learning to utilize and evolve agentic skills via credit-aware on-policy bidirectional self-distillation. *arXiv preprint arXiv:2606.29502*, 2026.

Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural networks*, 22(5-6):544–557, 2009.

Hao Wang, Guozhi Wang, Han Xiao, Yufeng Zhou, Yue Pan, Jichao Wang, Ke Xu, Yafei Wen, Xiaohu Ruan, Xiaoxin Chen, et al. Skill-sd: Skill-conditioned self-distillation for multi-turn llm agents. *arXiv preprint arXiv:2604.10674*, 2026a.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in neural information processing systems*, 33:5776–5788, 2020.

Zi-Han Wang, Zhengxi Lu, Zhiyuan Yao, Jinyang Wu, Jie Wu, Zhengzhou Cai, Yueqing Sun, Ziang Ye, Linji Hao, Qi Gu, et al. Agentopsd: Recursive self-distillation for agentic reinforcement learning. *arXiv preprint arXiv:2608.05987*, 2026b.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, et al. Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*, 2025.

Jinyang Wu, Shuo Yang, Zhengxi Lu, Fan Zhang, Yuhao Shen, Lang Feng, Haoran Luo, Zheng Lian, Shuai Zhang, Zhengqi Wen, et al. Seed: Self-evolving on-policy distillation for agentic reinforcement learning. *arXiv preprint arXiv:2607.14777*, 2026.

Peng Xia, Jianwen Chen, Hanyang Wang, Jiaqi Liu, Kaide Zeng, Yu Wang, Siwei Han, Yiyang Zhou, Xujiang Zhao, Haifeng Chen, et al. Skillrl: Evolving agents via recursive skill-augmented reinforcement learning. *arXiv preprint arXiv:2602.08234*, 2026.

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.

Jie Yang, Zhengyu Fang, Zelin Xu, Jiarui Sun, Xiran Fan, Junpeng Wang, Liang Wang, Qinghua Liu, Yiwei Cai, and Yan Zheng. Lastopd: Taming collapse in latent on-policy distillation, 2026a. URL https://arxiv.org/abs/2609.28845.

Shenzhi Yang, Guangcheng Zhu, Bowen Song, Haobo Wang, Mingxuan Xia, Xing Zheng, Yingfan Ma, Zhongqi Chen, Weiqiang Wang, Junbo Zhao, et al. Oprd: On-policy representation distillation. *arXiv preprint arXiv:2606.06021*, 2026b.

Shuo Yang, Jinyang Wu, Zhengxi Lu, Yuhao Shen, Fan Zhang, Lang Feng, Shuai Zhang, Haoran Luo, Zheng Lian, Zhengqi Wen, et al. Opid: On-policy skill distillation for agentic reinforcement learning. *arXiv preprint arXiv:2606.26790*, 2026c.

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.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

Zhiyuan Yao, Yuxin Chen, Zhengxi Lu, Zishan Xu, Yueqing Sun, Yifu Guo, Yuquan Lu, Zhengzhou Cai, Kangning Zhang, Zhuowen Han, et al. Skillrise: Agentic reinforcement learning for cross-task skill evolution. *arXiv preprint arXiv:2607.26784*, 2026.

Guo Yu, Wenlin Liu, Yulan Hu, Hao-Xuan Ma, Jun-Peng Jiang, and Han-Jia Ye. Dense supervision, sparse updates: On the sparsity and geometry of on-policy distillation. *arXiv preprint arXiv:2606.13657*, 2026.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024.

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.

A Normalized Alignment Distance

For any $a, b \in \mathbb{R}^d$ with $\|a\|_2 \geq \varepsilon$ and $\|b\|_2 \geq \varepsilon$, the normalization in Eq. 5 yields the unit vectors $\widehat{a} = a/\|a\|_2$ and $\widehat{b} = b/\|b\|_2$, so

$$\|\widehat{a} - \widehat{b}\|_2^2 = \|\widehat{a}\|_2^2 + \|\widehat{b}\|_2^2 - 2\,\widehat{a}^\top \widehat{b} = 2 - 2\,\frac{a^\top b}{\|a\|_2\,\|b\|_2} = 2 - 2\cos(a, b). \tag{7}$$

Setting $a = p_{S,k}^{(\ell)}$ and $b = h_{T,k}^{(\ell)}$ shows that each term of Eq. 5 depends only on the angle between the projected student state and the teacher state, and lies in $[0, 4]$. If a norm falls below ε, that vector is divided by ε instead and its norm stays below one, which is why the identity is stated only for norms of at least ε.

B Training Procedure

Algorithm 1 lists one Stage-II iteration of PR-OPD.

Algorithm 1 One Stage-II iteration of PR-OPD

Require: Rollout policy π_{θ_j}, task batch $\mathcal{Q}$, reference policy π_{ref}
Ensure: Updated policy $\pi_{\theta_{j+1}}$
Collect grouped trajectories $\mathcal{D}_j$ and rewards using π_{θ_j} without skills
Cache rollout log-probabilities and compute GRPO advantages
for all completed trajectories $\tau_i \in \mathcal{D}_j$ **do**
    Generate s_i from (τ_i, R_i) with π_{θ_j} ▷ no gradient
    Set $\widetilde{x} = H(x, s_i)$ for every turn of τ_i; keep y unchanged
end for
$\theta \leftarrow \theta_j$
for all optimization minibatches $\mathcal{B} \subset \mathcal{D}_j$ **do**
    Run the student on (x, y) with current θ; retain gradients
    Run the teacher on $(\widetilde{x}, y)$ with the same θ; disable gradients
    Compute $\mathcal{L}_{\mathrm{RL}} + \lambda \mathcal{L}_{\mathrm{PRA}}$ using Eqs. 5 and 6
    Accumulate microbatch gradients for θ and ψ and take an optimizer step
end for
$\theta_{j+1} \leftarrow \theta$; synchronize the rollout policy

C Experimental Details

Benchmarks. ALFWorld is a text-based embodied environment in which agents follow natural-language instructions to complete multi-step household tasks. We use its `eval_in_distribution` (seen) split with 140 task instances and report per-task-type and overall success rates. WebShop requires agents to search, inspect, and purchase products that satisfy user-specified constraints. We use the standard test split of the first 500 instructions and report task accuracy, which requires full completion, and graded task score, which also credits partial satisfaction of the requirements.

Implementation. Both backbones have hidden size $d = 2048$. PR-OPD starts from a GRPO checkpoint after N warm-start updates, with $N = 100$ on ALFWorld and $N = 120$ on WebShop, and continues to global step 150. The per-layer projector has one 512-dimensional hidden layer and is trained jointly with the policy at a learning rate of 10^{-4}. We keep the GRPO baseline's group-relative advantages, clipping, and invalid-action penalty, with KL coefficient $\beta = 0.01$, entropy coefficient $\eta = 0.001$, and alignment weight $\lambda = 0.01$. Our implementation builds on SDAR (Lu et al., 2026a) and uses `verl` (Sheng et al., 2025) for distributed training and `vLLM` (Kwon et al., 2023) for rollout generation. All experiments run on eight NVIDIA B200 GPUs with 64 CPU cores and 450 GB of host memory.

Evaluation. We evaluate the checkpoint at global step 150 with sampling temperature 0.4 on 128 episodes drawn from the evaluation split. Each configuration uses a single training run with training seed 0. Training curves, as in Figure 1(c), report in-training validation every five steps, whereas all tables use the step-150 evaluation.

Table 2: Ablation on ALFWorld with the 3B backbone, and the projector with the 1.7B backbone at the bottom. All runs end at step 150. Δ is the difference from GRPO in points.

Variant	**Teacher channel**	**Success (%)**	Δ
GRPO	none	66.4	0.0
+ token-level OPD	token probabilities	57.8	-8.6
+ hidden-state alignment, w/o projector	hidden states	**72.7**	$+6.3$
+ hidden-state alignment, w/ projector (**PR-OPD**)	hidden states	71.1	$+4.7$
SDAR, with curated skills	token probabilities	68.8	$+2.4$
Projector with the 1.7B backbone			
GRPO	none	43.8	0.0
+ hidden-state alignment, w/o projector	hidden states	36.7	-7.1
+ hidden-state alignment, w/ projector (**PR-OPD**)	hidden states	**46.9**	$+3.1$

Table 3: Sensitivity to the warm-start point on WebShop with the 1.7B backbone. Top: PR-OPD at step 150. Bottom: gain of direct alignment over GRPO continued from the same checkpoint, averaged over six evaluation seeds.

Start N	**Stage-II updates**	**Acc (%)**	**Score**
100	50	49.5	72.0
120	30	**57.0**	**73.8**
Start N	**Stage-II updates**	**ΔAcc**	**ΔScore**
100	30	$+3.15$	$+0.70$
120	30	$+\mathbf{9.62}$	$+3.23$
120	10	$+8.22$	$+4.00$
140	10	$+3.63$	$+\mathbf{6.35}$

D Ablation and Sensitivity Results

Tables 2 and 3 list the numbers shown in Figure 3.

E Linear Probing of Hindsight

Setup. At fixed GRPO@100 checkpoints on ALFWorld, we train logistic regression probes to distinguish a trajectory's own self-generated skill from four alternatives: no skill, unrelated filler text, another trajectory's skill, and a token-shuffled version of the same skill. Evaluation uses five folds grouped by trajectory, so all examples from one trajectory stay in the same fold, and chance accuracy is 50%. We compare three output-side features, namely the sampled token's log-probability, the log-probabilities of the skill-free distribution's top-64 candidates, and a 256-dimensional random projection of the full output probability vector, against hidden states from the first, middle, and final layers. The middle layers are 18 for Qwen2.5-3B-Instruct and 14 for Qwen3-1.7B-Instruct.

Results. Against another trajectory's skill (Figure 5c), middle-layer hidden states reach 65.4% and 79.4% accuracy for 3B and 1.7B, compared with 50.4–51.9% across the output features, while first-layer hidden states stay near chance at 49.8% and 49.2%. Against a token-shuffled copy (Figure 5d), which keeps the skill's tokens but changes their order, middle-layer probes reach 86.2% and 98.3% and final-layer probes 87.6% and 99.4%, compared with 50.2–54.0% for the output features. Against no skill and unrelated text (Figure 5a,b), middle-layer accuracies range from 88.6% to 99.8%, compared with 51.1–59.8% for the output features. Hidden states thus expose skill presence, trajectory matching, and token order under a simple linear readout, whereas the tested output features expose little of any.

F LAYER-WISE DECODABILITY OF MATCHED HINDSIGHT

We expand the matched-versus-other-trajectory comparison in Figure 5 by training a separate linear probe on each Transformer layer.

The data, trajectory-grouped folds, and output-feature baselines follow Section E.

Output-side features probe different views of the same predictive distribution. Let $p_t^{(c)} \in \mathbb{R}^{|\mathcal{V}|}$ denote the model's next-token probability vector at response position t under context condition c, where $\mathcal{V}$ is the vocabulary. The sampled response token y_t is held fixed across conditions. We construct three feature representations for separate logistic regression probes:

- **Sampled-token log-probability.** The scalar
$$z_{t,\text{sampled}}^{(c)} = \log p_t^{(c)}(y_t)$$
records the probability assigned to the token actually sampled in the rollout. It captures a single coordinate of the output distribution, not the probabilities of alternative tokens.
- **Top-64 log-probabilities.** Let K_t contain the 64 highest-probability tokens under the skill-free context. We use
$$z_{t,\text{top64}}^{(c)} = \big(\log p_t^{(c)}(v)\big)_{v \in K_t}.$$
The candidate tokens and their ordering are fixed across conditions at each response position. This feature captures changes among plausible alternatives, rather than only the sampled token.
- **256-dimensional distribution summary.** We apply a fixed random linear projection $P \in \mathbb{R}^{256 \times |\mathcal{V}|}$:
$$z_{t,\text{proj}}^{(c)} = P p_t^{(c)}.$$
This feature compresses the full vocabulary probability vector into 256 dimensions. The same projection is used across conditions; each coordinate combines probabilities from the vocabulary rather than representing a selected token.

For each feature representation, the probe predicts the context condition, not the next token or task outcome. Higher classification accuracy indicates that the tested features make the context distinction more linearly accessible. Conversely, near-chance accuracy does not establish that the full output distribution contains no relevant information: the selected features, compression, or linear readout may fail to expose it.

The most decodable layer depends on the backbone. Figure 4(c,d) shows that in both models, discrimination is near chance at the earliest layers and improves with further processing. For Qwen2.5-3B-Instruct, accuracy reaches 65.4% at layer 18, rises further in later intermediate layers, and ends at 66.0%. For Qwen3-1.7B-Instruct, accuracy reaches 79.4% at layer 14 and remains high across several intermediate layers before declining to 70.1% at the final layer. The middle-layer annotations are reference points, not claims about the peak accuracy.

Output-feature baselines remain weak for this specific distinction. The three output-feature probes at the right of Figure 4(c,d) achieve approximately 50–52% accuracy for both backbones, whereas many hidden layers support substantially stronger discrimination. This depth profile shows why a final-layer-only diagnostic can miss variation in the accessibility of hindsight-related information.

G REPRESENTATIONAL STRUCTURE UNDER HINDSIGHT CONDITIONING

Controlled comparisons characterize the structural effect of hindsight. We examine the initial Qwen2.5-3B-Instruct policy on ALFWorld using responses captured at rollout step 1, before GRPO warm-start training. We hold the model weights and sampled responses fixed and vary only the preceding context. The four conditions are a trajectory's self-generated skill, an other-trajectory skill, a token-shuffled version of the same skill, and length-matched unrelated filler text. All conditions are compared with the skill-free context.

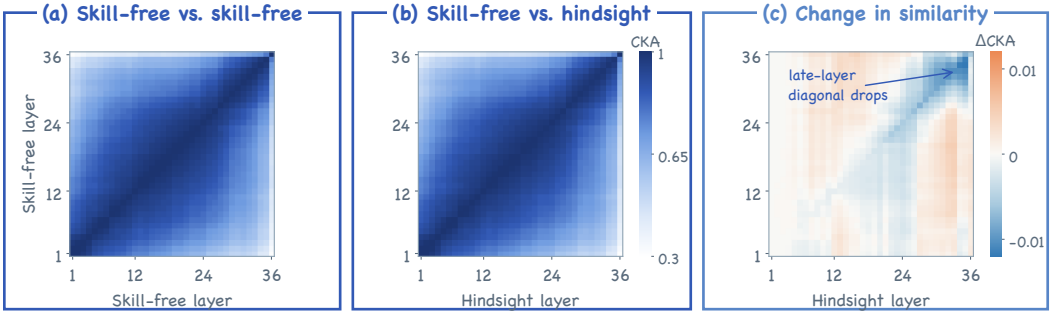

Figure 6: **Cross-layer representational structure under hindsight conditioning.** Linear CKA on corresponding response positions of the initial Qwen2.5-3B-Instruct policy on ALFWorld. Model weights and sampled response tokens are fixed. (a) Skill-free representations compared across layers: $A_{ij} = \mathrm{CKA}(H_0^{(i)}, H_0^{(j)})$. (b) Skill-free representations compared with those conditioned on the trajectory's self-generated skill: $B_{ij} = \mathrm{CKA}(H_0^{(i)}, H_{\mathrm{own}}^{(j)})$. (c) The difference $B - A$, where blue denotes lower CKA and orange denotes higher CKA after changing the column context. Rows index skill-free layers throughout. Panels (a,b) share a common color scale; panel (c) uses a separate, zero-centered scale.

Let $H_0^{(\ell)}$ and $H_c^{(\ell)}$ denote the matrices of hidden states at corresponding response positions in layer ℓ, under the skill-free context and condition c, respectively. We use linear centered kernel alignment (CKA) to compare their representational similarity structure. Unlike token-wise cosine distance, CKA compares the relationships across response positions rather than the direction of each individual hidden-state vector. We first examine cross-layer similarity maps and then resolve same-layer differences across the four conditions.

Hindsight largely preserves the global representational structure. Figure 6 shows closely matching cross-layer similarity patterns with and without hindsight. For the self-generated-skill condition, same-layer linear CKA averages 0.9959 across the 36 layers and remains at least 0.9884. The difference map nevertheless reveals nonuniform adjustments, including reduced similarity around the late-layer diagonal, with a maximum absolute difference of 0.0118 across all layer pairs. Thus, hindsight conditioning changes the representations while largely preserving their overall cross-layer similarity structure. The negative diagonal differences are relative to self-comparisons of CKA 1, not evidence of degraded task performance.

Same-layer dissimilarity resolves the differences beneath high CKA. To examine these changes more directly, we measure

$$d_c^{(\ell)} = 1 - \mathrm{CKA}\big(H_0^{(\ell)}, H_c^{(\ell)}\big). \tag{8}$$

Larger values indicate a greater change in representational similarity structure. Figure 4(e,f) compares this quantity across depth using both the linear and debiased CKA estimates from the diagnostic export.

Self-generated hindsight produces stronger middle-to-late-layer changes than the controls. The self-generated-skill curve rises through much of the network, reaches its maximum at layer 34, and declines at the final layer. From layer 13 onward, its linear-CKA dissimilarity exceeds that of all three controls. Averaged across layers, dissimilarity is 0.00406 for self-generated skills, compared with 0.00269 for other-trajectory skills, 0.00289 for shuffled skills, and 0.00067 for filler text. The debiased estimates preserve the same qualitative separation; the largest absolute difference between the two estimators is below 0.00067 across all conditions and layers. These comparisons indicate that the structural response depends on the supplied context and is not reproduced by the filler control alone.

High global similarity coexists with context-dependent adjustments. Together, the heatmaps and control curves provide complementary views of hindsight conditioning: the overall representational structure remains highly similar, while more detailed comparisons reveal depth-dependent differences between context conditions. This complements the token-wise sensitivity and linear-probe analyses, which examine individual-vector changes and the accessibility of hindsight distinctions. The CKA results characterize where structural changes occur; whether representation supervision improves policy learning is assessed separately through the training ablations.

H Alignment Dynamics During Training

Setup. Figure 7 tracks the alignment loss $\mathcal{L}_{\mathrm{PRA}}$ of Eq. 5 for PR-OPD with the projector, started after $N = 100$ or $N = 120$ GRPO updates on both backbones and both environments, and Table 4 summarizes each curve. The loss is averaged over layers and response tokens and equals $2 - 2\cos$ between the projected student state and the teacher state (Appendix A), so a loss of 0.2 corresponds to a mean cosine similarity of 0.9. Table 1 uses $N = 100$ on ALFWorld and $N = 120$ on WebShop, and every run ends at step 150.

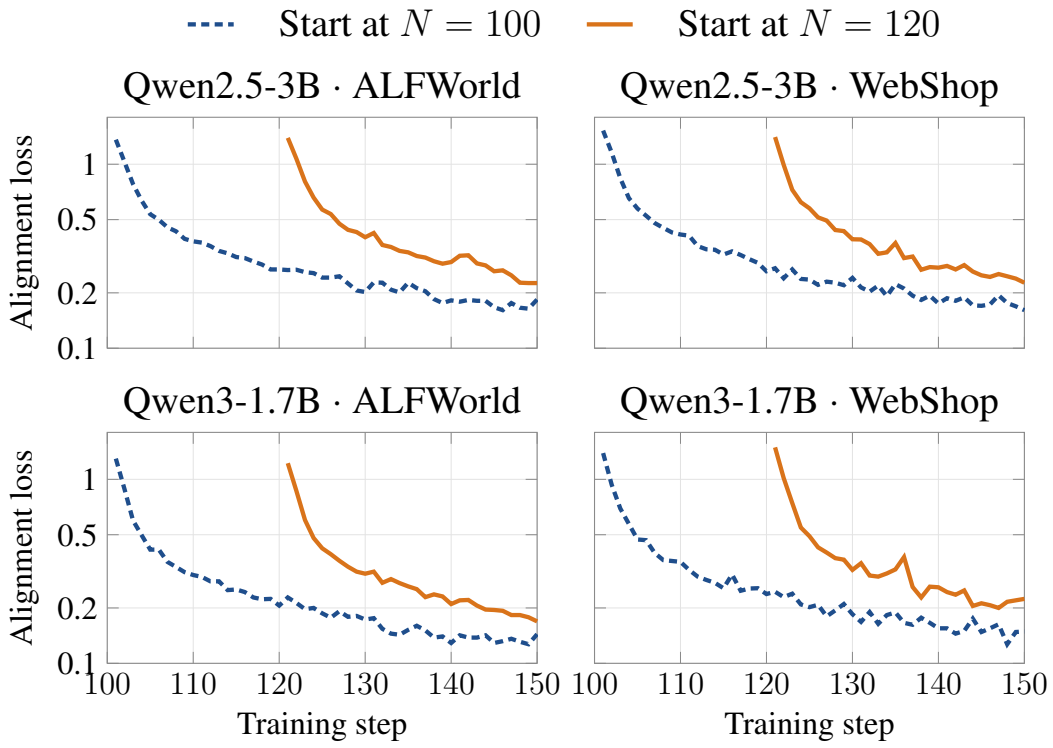

Figure 7: **Alignment loss of PR-OPD during Stage II.** $\mathcal{L}_{\mathrm{PRA}}$ with the projector at every training step, when representation supervision starts after $N = 100$ (dashed) or $N = 120$ (solid) GRPO updates, on a logarithmic axis. Values are raw per-step training logs of a single run, without smoothing.

Alignment is established within a few updates. When representation supervision begins, the freshly initialized projector yields a loss of 1.22 to 1.52, a cosine similarity of only 0.24 to 0.39 with the teacher. Within 4 to 7 updates the loss falls below 0.5, and within 11 to 19 updates below 0.3, so after about ten updates the projected student already points in nearly the same direction as its privileged self. All eight runs follow this pattern, showing that the alignment objective is easy to optimize across backbones, environments, and starting points.

Alignment improves steadily and stably throughout Stage II. After the fast initial phase, the loss keeps decreasing at a steady pace. With $N = 100$, it drops by a further 23–26% between updates

Table 4: Summary of the alignment curves in Figure 7. Init. is the loss at the first Stage-II update, and <0.3 is the first update at which the loss falls below 0.3. The next three columns average five consecutive steps: updates 26–30 compare the two starts after the same number of supervised updates, and steps 146–150 compare them at the end of training. The last column is the loss of the matching run without the projector, measured on the student's raw hidden states.

				w/ projector			**w/o proj.**
Setting	N	**Init.**	<0.3	**Upd. 26–30**	**Steps 146–150**	cos	**Steps 146–150**
3B · ALFWorld	100	1.360	17	0.224	0.170	0.915	0.012
	120	1.391	18	0.239	0.239	0.881	0.012
3B · WebShop	100	1.524	19	0.228	0.175	0.913	0.008
	120	1.405	18	0.242	0.242	0.879	0.009
1.7B · ALFWorld	100	1.296	11	0.180	0.134	0.933	0.031
	120	1.224	12	0.181	0.181	0.910	0.033
1.7B · WebShop	100	1.389	12	0.195	0.148	0.926	0.027
	120	1.492	13	0.213	0.213	0.893	0.026

26–30 and 46–50 in all four settings and reaches a cosine similarity of 0.913 to 0.933, and it is still decreasing at step 150, leaving room for longer Stage-II training. No run diverges or degrades late in training. Even the largest single-step increase after the first ten updates, 0.054 on 1.7B WebShop with $N = 120$ at step 136, is fully recovered at the next step. Because the teacher shares the student's weights, its layers correspond one-to-one with the student's and its target evolves together with the policy, which keeps the objective well behaved throughout training.

Alignment dynamics are robust to the starting point. Matched by the number of supervised updates rather than by global step, the two starts follow almost identical trajectories. Averaged over updates 26–30, their losses differ by at most 0.018 in every setting, and they fall below 0.3 within one update of each other (Table 4). The optimization of PR-OPD is therefore insensitive to when it begins, and the gap between the two curves at step 150 in Figure 7 simply reflects the 20 additional updates of the earlier start. This lets the warm-start point be chosen for the policy's competence alone, without affecting how well the student can follow its teacher.

A later start turns the same alignment into larger gains. On WebShop with 1.7B, starting at $N = 120$ raises accuracy from 49.5% to 57.0% (Table 3) with only 30 rather than 50 supervised updates, while its alignment proceeds at the same rate as that of the earlier start. Each supervised update is thus worth more once the policy has learned the task, in line with the warm-start rationale of Section 3.2: a more competent policy writes more informative hindsight, and PR-OPD transfers it efficiently. The benefit of PR-OPD therefore lies in what the privileged teacher knows rather than in driving the alignment loss to zero, and 30 updates of representation supervision suffice for the WebShop gains in Table 1.

The behavior holds across backbones and environments. The same fast-then-steady pattern appears on both backbones and both environments. The smaller backbone aligns even more readily: 1.7B falls below 0.3 after 11 to 13 updates, against 17 to 19 for 3B, and ends with a lower loss under every environment and start, for example 0.134 and 0.148 against 0.170 and 0.175 with $N = 100$.

The projector turns a fixed gap into a learnable target. Without the projector, the loss is measured directly on the student's raw hidden states and is not comparable to the projected loss. It is 0.008 to 0.012 for 3B and 0.026 to 0.033 for 1.7B over the last five steps and stays within 0.01 of its initial value in every run, because the teacher shares the student's weights and moves with every update. The projector instead gives the student a mapping it can learn, so the alignment loss decreases steadily while the raw states that feed the language-model head remain free to serve the policy (Section 3.5). The raw gap is also 2.5 to 3.5 times larger for 1.7B than for 3B, in line with the higher probe accuracy of 1.7B in Section 4.4, and 1.7B is exactly where the projector matters most (Section 4.3), lifting ALFWorld success from 36.7% to 46.9%.

Together, these curves show that the representation objective of PR-OPD is fast to fit, stable over training, consistent across settings, and insensitive to when it starts, so its gains come from the privileged hindsight it transfers rather than from delicate optimization.