ASCENT: ONLINE TEST-TIME TRAINING OF LONG-HORIZON AGENTS VIA SELF-DISTILLATION OF VERIFIED EXPERIENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

A large language model (LLM) agent solves long-horizon tasks through many reasoning-action turns, with one verification signal at termination. Deployed agents face streams of related tasks, making their own trajectories a natural resource for improvement. In-context evolving agents store reflections, memories, or skills as text, so reuse depends on retrieving the right experience and on a frozen policy executing it. We study *Online Agentic Test-Time Training* (OaTTT), which trains the LLM during deployment on its own execution trajectories. Each task is attempted once, and the executed trajectory with its verification result is the only learning data for updates that persist to later tasks. Directly imitating the sampled tokens of this single rollout destabilizes the policy. We introduce ASCENT (*Agentic Self-distillation for Cross-task EvolutioN at Test-time*), which instead self-distills verified experience. A frozen copy of the LLM receives the verifier-accepted trajectory, with environment-rejected turns removed, as privileged information and predicts next-token distributions along it with this hindsight; distilling them into persistent LoRA fast weights updates the agent for later tasks, without an external reference solution or stronger teacher. We characterize the population target of this distillation and the limits of sparse outcome selection. Across ALFWorld, WebShop, and AppWorld on varied model scales, ASCENT improves task success and interaction efficiency as experience accumulates, outperforms the compared online adaptation methods, and transfers to held-out scenes, showing that a deployed agent can consolidate its own verified experience into its policy without a separate training phase or memory retrieval at inference.

1 INTRODUCTION

A large language model (LLM) agent interacts with its environment to accomplish a task, iteratively reasoning about the current state and taking actions such as tool calls (Yao et al., 2023). The LLM serves as the core of the agent's decision policy. Agent tasks are often long-horizon, requiring multi-turn interactions with complex environments. Each completed interaction forms one episode, a trajectory of reasoning, actions, and observations.

Deployed agents often operate in persistent, long-running roles, encountering a continuous stream of tasks and interactions with users and environments. Because real-world tasks are complex and diverse, an agent built on a frozen LLM may not handle every situation it faces. The trajectories the agent gathers in the task stream are a natural resource for improving the agent at test time, as experience from earlier tasks can enhance performance on related tasks encountered later.

Relying on the in-context learning capability of LLMs, one line of work stores reflections on completed episodes, memories, and reusable skills as text and reuses them on new tasks (Shinn et al., 2023; Zhao et al., 2024; Ouyang et al., 2026; Zhang et al., 2026a; Fang et al., 2026; Zhang et al., 2026b). Through this in-context adaptation, the agents evolve at test time as their stored text grows with deployment. Such in-context adaptation is fast and usually stable, but its effect depends on retrieving relevant experience or skill and on the frozen policy executing it over a long interaction. Efforts are thus also taken to improve the core LLM for the agentic task process, via parametric adaptation to internalize the experiences in model weights. Some test-time training (TTT) methods adapt

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.

the LLM's parameters within a single episode and reset them afterward, training on self-generated supervision (Wang et al., 2026c, aTTT) or QA pairs from the episode's accumulated context (Ren et al., 2026, TMEM). Other TTT methods require pre-deployment training of the adaptation behavior (Zweiger et al., 2025; Ren et al., 2026) or multiple sampled attempts per problem (Zuo et al., 2025; Wang et al., 2026a). Reinforcement learning (RL) post-training also improves the agentic capability of LLMs, which optimizes the policy before deployment on a training task set, typically with many rollouts per task (Shao et al., 2024; Feng et al., 2025; Tan et al., 2026; Wu et al., 2026; Lu et al., 2026) or single-rollout RL with a learned critic (Hou et al., 2026).

To explore the potential of test-time training directly on deployment trajectories, we study *Online Agentic Test-Time Training* (OaTTT), which trains the LLM's weights on the agent's own execution trajectories to improve task success. Tasks arrive online in a stream, and the agent executes each task once, receiving the environment's responses and a final verification. The executed trajectory and its verification result are the only learning data, driving persistent updates to the LLM that carry to subsequent tasks. Like streaming in-context agents (Ouyang et al., 2026; Zhang et al., 2026a), the agent learns from each trajectory once, without repeated trials, but accumulates the effect in weights rather than in textual memory. No task-specific training split or offline adaptation stage is used.

Can a deployed agent turn its own verified execution trajectories, one per task, into persistent weight updates that improve its success on subsequent tasks?

We propose ASCENT (*Agentic Self-distillation for Cross-task EvolutioN at Test-time*), which enables agents to learn during deployment, turning each verified experience trajectory into a persistent weight update. Training directly on these streaming trajectories, with environment verification as the reward, is straightforward but fragile, because each task yields only one rollout from the current policy. Directly imitating the sampled tokens of the rollouts destabilizes (e.g., online SFT or RFT) the policy and degrades the agent's task success, as shown in Fig. 1. ASCENT instead self-distills from the verified experience to enable more stable online test-time training. A stable version of the agent, a frozen copy of its initial LLM, receives the verifier-accepted trajectory as privileged information and, at every token position of that trajectory, predicts a full next-token distribution with this hindsight. Environment verification acts as rejection sampling, so this privileged trajectory gives the self-distillation positive guidance over the full horizon. Distilling these distributions into persistent LoRA fast weights lets the agent absorb the selected experience without treating every sampled token as a target (Hu et al., 2022; Agarwal et al., 2024; Zhao et al., 2026). ASCENT further refines the accepted trajectory by removing the turns whose actions the environment rejected, keeping the reasoning–action content of admissible turns as privileged information. The filter changes only the teacher's input; the agent is still trained at the positions of removed turns. The updated agent then generates the next task's trajectory, so ASCENT is online self-evolution across the task stream, gated by verification. ASCENT keeps online adaptation stable and improves both task success and interaction efficiency as experience accumulates, as validated on different tasks and base models (Fig. 2), and the learned fast weights transfer to held-out scenes. Our contributions are as follows:

- We formulate Online Agentic Test-Time Training (OaTTT), in which a deployed agent attempts each task once and the executed trajectory with its verification result is the only data for updates persisting to later tasks. We identify instability from imitating sampled tokens of these trajectories.
- We propose ASCENT, which self-distills verified experience for stable online adaptation. A frozen copy of the initial LLM, conditioned on the verifier-accepted trajectory with environment-rejected turns removed, supplies full-distribution targets that are distilled into persistent LoRA fast weights. We characterize the population target of this distillation and the limits of sparse outcome selection.
- Across ALFWorld, WebShop, and AppWorld on varied model scales, we show that ASCENT improves task success and interaction efficiency as experience accumulates, outperforms the compared online adaptation methods, and transfers to held-out scenes, where continued TTT remains effective. We also ablate the privileged information and distillation divergence, analyze in-context adaptation failure modes, and present a preliminary in-context and parametric co-evolution study.

2 Related Work

Self-evolving agents. In-context methods store reflections, memories, or reusable skills and retrieve them at inference (Shinn et al., 2023; Zhao et al., 2024; Wang et al., 2024; Suzgun et al., 2026;

Ouyang et al., 2026; Zhang et al., 2026a; Fang et al., 2026; Xu et al., 2025; Zhang et al., 2026b); reuse depends on retrieval and on the frozen policy executing it over a long interaction. Parametric methods internalize experience in weights (Zweiger et al., 2025; Ren et al., 2026). ASCENT consolidates verifier-selected interactions into persistent LoRA fast weights that carry across tasks.

Online learning and test-time training. Online learning updates a model from sequentially arriving data (Shalev-Shwartz, 2012). Test-time training adapts parameters on inference-time signals (Sun et al., 2020; Wang et al., 2021; 2022; Hardt & Sun, 2024; Hu et al., 2025), though Agentic TTT and TT-SI reset parameters after each episode or instance (Wang et al., 2026c; Acikgoz et al., 2026). Existing LLM self-training and RL methods require multiple solutions, correctness-selected generations, or pre-deployment rollouts (Zuo et al., 2025; Wang et al., 2026a; Zelikman et al., 2022; Yuan et al., 2023; Gulcehre et al., 2023; Singh et al., 2024; Shao et al., 2024; Kim et al., 2026; Hou et al., 2026). ASCENT persists updates across tasks from one attempt each, selecting by the environment verdict and distilling frozen-teacher soft targets.

On-policy distillation and self-distillation. On-policy distillation queries a teacher on student-sampled sequences, reducing train–generation mismatch (Agarwal et al., 2024). OPSD conditions a self-teacher on reference reasoning while the student receives only the problem (Zhao et al., 2026); SDFT and SDPO condition the teacher on demonstrations or environment feedback (Shenfeld et al., 2026; Hübotter et al., 2026). Recent agentic methods combine privileged distillation with pre-deployment RL (Wang et al., 2026b; Lu et al., 2026; Wu et al., 2026; Wang et al., 2026d; Liu et al., 2026). CLaaS applies persistent self-distillation with an EMA teacher and asynchronous replay in adversarial dialogue (Yang et al., 2026); HinT-SD targets trajectory-hindsight distillation at failure-relevant action spans (Chen et al., 2026). ASCENT operates online with a frozen initial teacher, no replay, and immediate forward-KL updates from the student's verifier-selected interaction.

3 METHOD

3.1 ONLINE AGENTIC TEST-TIME TRAINING

Self-evolving agent-environment interaction. At deployment, the agent encounters a one-pass task stream $\mathcal{X}_{1:N} = (x_1, \ldots, x_N)$, where each x_i specifies a goal in a partially observed interactive environment $\mathcal{E}$. We call the agent's single attempt at task x_i episode i. At turn t of episode i, the agent conditions on

$$H_{i,t} = (x_i, o_{i,1}, w_{i,1}, o_{i,2}, \ldots, w_{i,t-1}, o_{i,t}), \tag{1}$$

where $o_{i,t}$ is the current observation and $w_{i,t}$ is the generated reasoning-action response (Yao et al., 2023). An action parser ρ extracts the action $a_{i,t} = \rho(w_{i,t})$ from each response, and no action results when parsing fails. Let $\mathcal{M}_i$ denote an optional, evolving cross-task-episode memory, such as reflections, memories, or skills accumulated from earlier episodes (Ouyang et al., 2026; Zhang et al., 2026a; Fang et al., 2026). The deployed policy is

$$\pi_i(\cdot \mid H_{i,t}) := p_{\theta_i}\big(\cdot \mid g(H_{i,t}; \mathcal{M}_i)\big), \qquad w_{i,t} \sim \pi_i(\cdot \mid H_{i,t}). \tag{2}$$

Here g renders the instructions, history, and content from $\mathcal{M}_i$. The environment executes $a_{i,t}$ and returns $o_{i,t+1} \sim \mathcal{E}(\cdot \mid H_{i,t}, a_{i,t})$, and episode i yields its complete trajectory τ_i at termination.

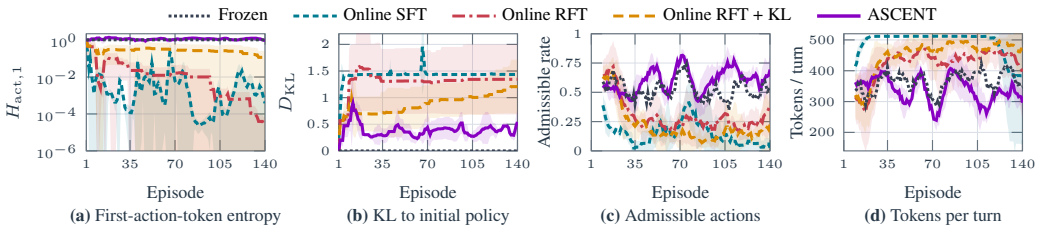
OaTTT as self-evolution. In online evolution, there is no separate task-specific training split or pre-deployment adaptation. We aim to update the policy LLM model from its own interaction experience during deployment. The current policy π_i executes each task x_i once, producing a single episode:

$$\tau_i \sim \mathbb{P}_{\pi_i, \mathcal{E}}(\cdot \mid x_i), \qquad v_i = \mathcal{V}(x_i, \tau_i) \in \{0, 1\}, \tag{3}$$

where v_i is the sparse, episode-level verification outcome, returned when episode i ends and recorded before any update. The trajectory τ_i and outcome v_i then update the policy:

$$(\theta_{i+1}, \mathcal{M}_{i+1}) = \mathsf{Evolve}(\theta_i, \mathcal{M}_i; x_i, \tau_i, v_i). \tag{4}$$

The update operator Evolve is specific to each method. In-context adaptation methods evolve the agent by updating its memory $\mathcal{M}_i$ while keeping the model fixed; because the policy in Eq. (2) conditions on $\mathcal{M}_i$, this memory update also changes the policy. Parametric methods evolve θ_i (Zweiger et al., 2025; Ren et al., 2026); joint methods may evolve both; and frozen methods evolve neither. This ordering follows a prequential online protocol (Dawid, 1984; Blum et al., 1999): each reported



(a) First-action-token entropy **(b)** KL to initial policy **(c)** Admissible actions **(d)** Tokens per turn

Figure 1: **Direct supervised OaTTT loses executable behavior.** Mean Qwen3.5-4B trends with SD shading on the ALFWorld seen stream: (a) first-action-token entropy; (b) KL to the initial policy; (c) accepted-action rate; and (d) generated tokens per turn.

outcome is computed before its own trajectory can influence the policy. In OaTTT, that scored trajectory and its verification result are also the only learning data, with no retries, additional rollouts, or replay buffer. Reported success follows each benchmark's criterion, which can be stricter than the one defining v_i. Success on later tasks thus reflects how much earlier experience helps. Observations describe the resulting states but give no turn-level reward or causal credit for the verified outcome (Sutton & Barto, 2018; Zeng et al., 2025; Tan et al., 2026).

3.2 Instability of Sampled-Token Self-Imitation in OaTTT

A straightforward way to perform OaTTT is self-imitation of the policy's own completed interactions. We formulate and examine three such methods, Online SFT, outcome-gated Online RFT, and Online RFT+KL, each of which retains a likelihood term on the agent's sampled tokens. Under per-episode updates, all three become unstable and progressively lose executable behavior. In all parametric variants, we freeze the base θ_0 and carry a trainable LoRA adapter ϕ_i across episodes (Hu et al., 2022). We call ϕ_i the fast-weight state entering episode i; the initial state produces zero adapter output, and within an episode update, ϕ denotes the adapter variable initialized at ϕ_i.

Sampled-token self-imitation reinforces every generated token. The policy entering episode i is p_{θ_0,ϕ_i}. Let $y_{i,t,1:L_{i,t}}$ denote the tokens of response $w_{i,t}$, where $L_{i,t}$ is its token length. Let $\widehat{H}_{i,t} = g(H_{i,t}; \mathcal{M}_i)$ denote the rendered rollout context used during collection. Define the saved training prefix $c_{i,t,j} = \widehat{H}_{i,t} \oplus y_{i,t,<j}$. Thus the prefix contains the rendered rollout context, including any retrieved context that conditioned collection. Let $\mathcal{I}_i$ index the retained turns with nonempty responses. We average tokens within each turn and then average across turns by setting $\alpha_{i,t,j} = (|\mathcal{I}_i| L_{i,t})^{-1}$. With an episode gate $b_i \in \{0, 1\}$, sampled-token self-imitation minimizes

$$\mathcal{L}_i^{\text{NTP}}(\phi; b_i) = -b_i \sum_{t \in \mathcal{I}_i} \sum_{j=1}^{L_{i,t}} \alpha_{i,t,j} \log p_{\theta_0,\phi}(y_{i,t,j} \mid c_{i,t,j}) \,. \tag{5}$$

Online SFT sets $b_i = 1$ and updates after every episode. Online RFT sets $b_i = v_i$, so only verifier-accepted trajectories become immediate training targets (Yuan et al., 2023; Shao et al., 2024). Online RFT+KL also anchors the adapted policy to the frozen base distribution:

$$\mathcal{L}_i^{\text{OnlineRFT+KL}}(\phi) = \mathcal{L}_i^{\text{NTP}}(\phi; v_i) + \lambda v_i \sum_{t \in \mathcal{I}_i} \sum_{j=1}^{L_{i,t}} \alpha_{i,t,j} D_{\text{KL}}(p_{\theta_0}(\cdot \mid c_{i,t,j}) \,\|\, p_{\theta_0,\phi}(\cdot \mid c_{i,t,j})) \,. \tag{6}$$

Outcome gating changes which episodes become training data, while reference KL adds a soft anchor to the base distribution. Both retain maximum-likelihood imitation of every sampled token, including incidental reasoning, formatting, and detours. Observations identify visited states but not which decisions produced the episode outcome.

Per-episode self-imitation collapses into non-executable behavior. In a controlled ALFWorld study (Fig. 1), Online SFT and Online RFT lose first-action-token entropy as KL to the initial policy rises (a,b); reference KL retains some entropy but still drifts substantially. All three increasingly propose invalid actions while responses stay long or lengthen (c,d), often producing fluent planning text without a parseable action. Over the final ten episodes, success falls below that of the original agent with the frozen LLM.

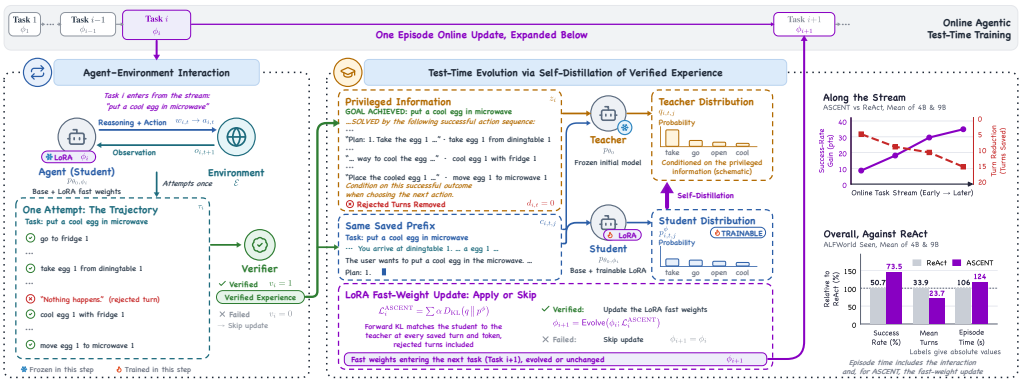


Figure 2: **ASCENT's online test-time training.** The student attempts each task once; a verified trajectory becomes privileged information for the frozen teacher, whose soft targets update the LoRA fast weights for the next task. Excerpts from Fig. 8; charts show ALFWorld seen-stream means of 4B and 9B runs (per-scale in Fig. 4(a) and Table 1); episode time includes the update.

These results separate selecting experience from learning from it. The verifier marks a trajectory as successful but not which decisions caused the success, so imitating that single rollout also reinforces its incidental reasoning, formatting, and detours. Each episode thus pulls the weights toward one specific trajectory, and because the updated policy generates the next one, these aggressive updates compound and can erode the base LLM's existing behavior, consistent with the collapse above. An online learner therefore needs guidance that does not treat every generated token as a correct target.

3.3 ASCENT: OaTTT via Self-Distillation of Verified Experience

Overview. Rather than imitating its own sampled tokens, ASCENT lets the agent absorb verified experience through a more stable, regularized target, namely the hindsight-informed predictions of its own initial model. Given an LLM agent with base model p_{θ_0}, ASCENT keeps θ_0 fixed and updates low-rank fast weights ϕ_i (Hu et al., 2022), so the agent p_{θ_0,ϕ_i} evolves in its parameters during OaTTT. In episode i, the agent attempts task x_i once, producing a complete long-horizon trajectory τ_i and its verification outcome v_i. If the episode is accepted ($v_i = 1$), a light self-distillation update turns ϕ_i into ϕ_{i+1}; otherwise $\phi_{i+1} = \phi_i$. In this self-distillation, a stable version of the LLM, the frozen base model p_{θ_0} conditioned on privileged information z_i drawn from the verified and filtered trajectory, serves as the teacher, and the current agent serves as the student. The updated agent can then produce better trajectories on later tasks, which in turn supply better privileged information, so the agent self-evolves online along the stream (Fig. 2).

3.3.1 On-Policy Self-Distillation Objective

On-policy distillation trains on student-sampled prefixes using a teacher's next-token distributions (Agarwal et al., 2024). OPSD conditions a self-teacher on privileged information unavailable to the student (Zhao et al., 2026). ASCENT applies this asymmetry to each verifier-accepted episode, with the frozen base model p_{θ_0} as the teacher.

The verification outcome v_i selects trajectories for adaptation but assigns no credit to individual turns. Updating only on accepted episodes ($v_i = 1$) acts as rejection sampling (Yuan et al., 2023; Shao et al., 2024), so the teacher's hindsight always comes from a trajectory that passed verification. From this trajectory, without an external reference solution, we construct the privileged information z_i (Sec. 3.3.2) and obtain the following next-token distributions at each saved student prefix $c_{i,t,j}$:

$$q_{i,t,j}(\cdot) = \mathrm{stopgrad}[p_{\theta_0}(\cdot \mid z_i \oplus c_{i,t,j})]\,, \tag{7}$$

$$p^{\phi}_{i,t,j}(\cdot) = p_{\theta_0,\phi}(\cdot \mid c_{i,t,j})\,. \tag{8}$$

The turn set $\mathcal{I}_i$, token lengths $L_{i,t}$, and weights $\alpha_{i,t,j}$ are those of Eq. (5), so ASCENT trains at the same prefixes as sampled-token self-imitation but replaces its target. Starting from $\phi = \phi_i$,

ASCENT minimizes the forward KL to the detached teacher distributions:

$$\mathcal{L}_i^{\mathrm{ASCENT}}(\phi) = \sum_{t \in \mathcal{I}_i} \sum_{j=1}^{L_{i,t}} \alpha_{i,t,j} D_{\mathrm{KL}}\left(q_{i,t,j} \,\middle\|\, p^{\phi}_{i,t,j}\right). \tag{9}$$

The fast weights entering episode $i+1$ are

$$\phi_{i+1} = \begin{cases} \mathsf{Evolve}(\phi_i; \mathcal{L}_i^{\mathrm{ASCENT}}), & v_i = 1, \\ \phi_i, & v_i = 0, \end{cases} \tag{10}$$

where Evolve is the operator from Eq. (4), here a few optimizer steps on Eq. (9) (Appendix B.3). The teacher stays fixed at p_{θ_0}, so self-distillation injects each verified trajectory's experience into the student's fast weights through a stable version of the LLM. Student updates never change the teacher's targets for a given trajectory, which helps mitigate student drift during online learning (Appendix C.1 examines teachers that track the student). Because z_i covers the full horizon of the verified trajectory, the teacher sees at each turn how the episode continued to verified success, giving the student better-informed turn-level guidance. We next describe how the verified trajectory supplies this hindsight and the student's training prefixes.

3.3.2 Distilling Verified and Filtered Experience Trajectories

The complete trajectory provides hindsight without turn-level credit. Episode i, with T_i turns, yields

$$\tau_i = \left(x_i, o_{i,1}, (w_{i,t}, o_{i,t+1})_{t=1}^{T_i}\right). \tag{11}$$

It contains the task, each generated response, and the observation returned after its action. At a saved prefix $c_{i,t,j}$, the student had not yet seen the rest of $w_{i,t}$ or the later turns and observations. The trajectory thus offers a hindsight view of earlier decisions (Harutyunyan et al., 2019; Tan et al., 2026), although it does not identify which of them caused the verified outcome.

Admissibility filtering removes rejected turns only from the teacher's input. Even an accepted trajectory can contain turns whose response yields no parseable action or whose action the environment rejects as invalid. ASCENT marks them with the environment's action-validity check, without separately filtering execution errors:

$$d_{i,t} = \begin{cases} 1, & \text{if } \rho(w_{i,t}) \text{ returns an action accepted as valid,} \\ 0, & \text{otherwise, including when no action is parsed.} \end{cases} \tag{12}$$

The teacher's privileged information is then

$$z_i = h_{\mathrm{ASCENT}}\left(x_i,\ (w_{i,t})_{t:\, d_{i,t}=1}\right), \tag{13}$$

where the fixed map h_{ASCENT} serializes the task goal, a success statement, and the reasoning–action responses of admissible turns, with actions parsed by ρ; by default, z_i omits observation text. The verified trajectory thus plays two roles. Its admissible turns build z_i, while all its nonempty responses, including those of rejected-action turns, supply the student's training prefixes $c_{i,t,j}$. Sec. 4.4 ablates the content of z_i and the filter.

The student fits hindsight-informed distributions, not sampled tokens. For a student logit ℓ_k^{ϕ} and a sampled token y, with position indices suppressed, the forward-KL and sampled-token objectives have gradients

$$\frac{\partial D_{\mathrm{KL}}(q \| p^{\phi})}{\partial \ell_k^{\phi}} = p_k^{\phi} - q_k, \qquad \frac{\partial[-\log p_y^{\phi}]}{\partial \ell_k^{\phi}} = p_k^{\phi} - \mathbf{1}\{k = y\}. \tag{14}$$

Self-imitation's direct gradient thus favors y, whereas forward KL can lower its logit when $q_y < p_y^{\phi}$, for example at a rejected-action turn omitted from z_i. Under a fixed collection distribution, the population target at each prefix averages the teacher distributions of verifier-accepted trajectories through that prefix, weighted as in Eq. (9) (Appendix A.1.2).

Hindsight can reduce unnecessary turns but does not guarantee it. Without turn-level credit, an accepted trajectory may still contain unnecessary turns, and one rollout cannot show what an alternative action would have achieved (Appendices A.1.1 and A.1.3). Hindsight can change the teacher's predictions at earlier prefixes, but neither hindsight conditioning nor admissibility filtering guarantees a shorter continuation. Fig. 4(a) shows the observed reduction (Sec. 4.3).

Method	Pick	Look	Clean	Heat	Cool	Pick2	**Avg. SR**	**Mean turns** (all)	**Time** s
Qwen3.5-4B									
Frozen ReAct	91.4	46.2	40.7	6.2	8.0	54.2	46.4	35.1	97
MemP	94.3	61.5	**71.6**	12.5	36.0	73.6	63.6±5.9	29.7	58
ReasoningBank	94.3	30.8	33.3	14.6	22.7	61.1	49.0±1.5	35.6	89
ACE	90.5	17.9	49.4	**22.9**	42.7	41.7	51.2±13.8	34.6	92
A-Mem	88.6	30.8	35.8	10.4	20.0	58.3	46.7±5.4	34.5	79
MemRL	85.7	23.1	33.3	12.5	16.0	62.5	45.0±5.2	35.4	89
Dynamic Cheatsheet	91.4	23.1	44.4	6.2	28.0	54.2	48.6±4.1	36.9	94
TT-OPSD	95.2	46.2	29.6	2.1	25.3	72.2	51.0±3.7	32.9	87
ASCENT	**99.0**	**69.2**	69.1	20.8	**54.7**	**75.0**	**69.5**±5.6	**26.3**	115
Qwen3.5-9B									
Frozen ReAct	94.3	61.5	44.4	0.0	8.0	91.7	55.0	32.6	115
MemP	95.2	74.4	75.3	16.7	38.7	80.6	67.9±2.9	27.4	85
ReasoningBank	95.2	59.0	75.3	22.9	36.0	76.4	66.0±2.4	29.0	93
ACE	93.3	51.3	**76.5**	29.2	**57.3**	75.0	69.3±6.3	28.3	113
A-Mem	95.7	69.2	53.7	9.4	22.0	85.4	60.4±0.4	30.7	101
MemRL	97.1	53.8	66.7	12.5	20.0	**95.8**	63.6±2.5	29.5	110
Dynamic Cheatsheet	97.1	61.5	48.1	0.0	20.0	83.3	57.1±4.8	32.2	126
TT-OPSD	98.1	79.5	54.3	12.5	5.3	**95.8**	61.2±1.8	29.3	99
ASCENT	**100.0**	**89.7**	**76.5**	**37.5**	48.0	**95.8**	**77.4**±1.8	**21.0**	133

Table 1: **ALFWorld seen-stream results.** Success rate (SR, %), mean turns, and runtime (s) per episode for ReAct and online methods; more comparison with offline methods are in Table 4.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETTING

Benchmarks. We evaluate on ALFWorld seen split (Shridhar et al., 2021) and WebShop (Yao et al., 2022), with AppWorld (Trivedi et al., 2024) in Appendix B.5. ALFWorld unseen split (held-out rooms and layouts) tests frozen transfer and continued TTT. More details in Appendix B.

Baselines. We compare ASCENT with frozen ReAct (Yao et al., 2023), offline and online self-evolution methods (Appendix B.2), and TT-OPSD, a turn-local distillation baseline adapted from OPSD (Zhao et al., 2026) whose teacher receives only current-turn experience.

Evaluation metrics. Each task is scored before its trajectory can update the method. We report ALFWorld exact success overall and by task family; WebShop exact success and mean score; and mean per-episode turns and wall-clock time. Replicated results are reported as mean±SD.

Implementation details. We use Qwen3.5-4B and Qwen3.5-9B (Qwen Team, 2026). ALFWorld and WebShop use one scored rollout per task and a 50-turn budget. All online methods start from the same ReAct template and receive the same environment verification signal; offline baselines use the same fixed-size training split. More details in Appendix B.3.

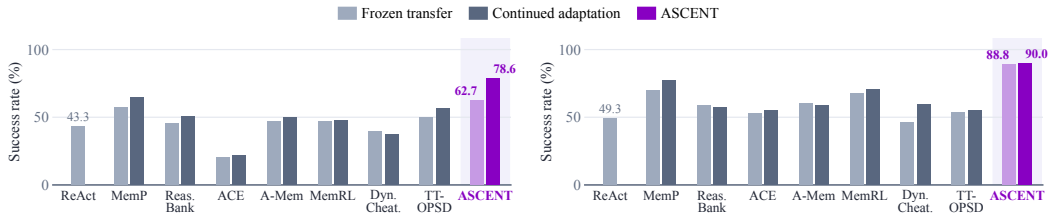
4.2 MAIN EXPERIMENTAL RESULTS

Online ASCENT consistently improves long-horizon task success. As shown in Tables 1 and 2, ASCENT improves exact success over frozen ReAct by more than 22 points across ALFWorld and WebShop at both scales and exceeds the strongest online comparator. It also achieves the highest WebShop mean task score and outperforms every offline in-context adaptation method on ALFWorld. These consistent gains show that cross-episode parameter updates can extract reusable behavior from the same sparse environment verification available to in-context adaptation methods.

Cross-episode gains extend to difficult tasks. Table 1 shows that frozen ReAct rarely solves `Heat` or `Cool`, whereas ASCENT improves both task families at each scale. Fig. 4(a) shows that ASCENT's gain over frozen ReAct is much larger on later 4B tasks, while frozen ReAct remains flat. ASCENT derives this improvement from the model's own verified deployment interactions, without a separate stronger teacher model or a pre-deployment adaptation stage.

Method	Qwen3.5-4B				Qwen3.5-9B			
	Strict-SR	Score	Turns	Time s/ep	Strict-SR	Score	Turns	Time s/ep
Frozen ReAct	17.8	25.9	39.5	87	17.2	27.8	37.5	166
MemP	36.8±3.4	47.7±4.8	28.7	74	22.4±2.5	38.9±4.1	32.9	152
ReasoningBank	10.6±0.7	21.6±1.0	42.1	176	12.6±2.1	21.8±1.5	41.5	226
ACE	10.6±3.2	23.6±4.3	40.6	107	16.0±1.4	32.0±2.6	36.1	144
A-Mem	23.2±3.1	36.4±2.9	33.9	98	15.4±0.6	32.5±4.3	35.6	176
MemRL	21.8±1.2	32.3±2.5	37.1	106	14.0±4.2	27.4±1.7	38.9	166
Dynamic Cheatsheet	16.2±1.5	28.4±2.1	39.4	133	13.4±2.9	27.4±3.0	39.1	211
TT-OPSD	25.6±0.5	37.1±1.7	34.6	100	24.2±2.4	35.9±2.3	34.8	108
ASCENT	**41.9**±0.3	**62.4**±1.0	**17.8**	76	**41.5**±1.8	**56.8**±1.7	**26.7**	164

Table 2: WebShop results for frozen ReAct and online methods (Strict-SR: task score = 1; Score: mean task score). Offline methods and full results in Table 7.



(a) Qwen3.5-4B (b) Qwen3.5-9B

Figure 3: **Cross-scene transfer and continued TTT on ALFWorld unseen.** Transfer freezes the seen-stream adaptation state; continued TTT resumes online updates. More in Tables 5–6.

Fewer turns and modest online cost. Tables 1 and 2 show that ASCENT reduces capped all-episode mean turns at both scales on both benchmarks; paired common-success comparisons confirm shorter ALFWorld trajectories (Fig. 4(a); Sec. 4.3). Episode time includes the rollout and LoRA update after verified successes; it is modestly higher than frozen ReAct on ALFWorld and comparable or lower on WebShop.

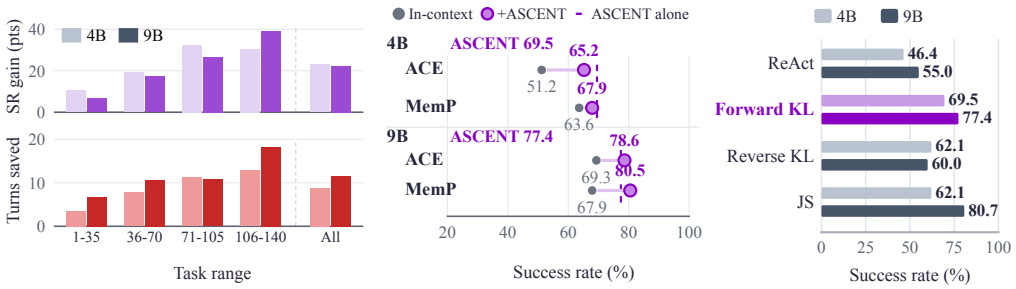
The evolved policy transfers and continues learning under scene shift. As shown in Fig. 3 and Tables 5–6, ASCENT leads the task-online comparators in both frozen transfer and continued adaptation at both scales. At 4B, the continued-TTT substantially exceeds frozen transfer; at 9B, continuation can preserve the already strong transfer result. These results show that seen-stream experience is retained in the evolved policy and that the self-evolution protocol remains effective.

4.3 Analysis and Discussion

Case study on long-horizon task completion. On "put a cool egg in microwave," the frozen policy and five in-context adaptation variants exhaust the turn budget, whereas ASCENT completes the task in 20 turns; Appendix C.3 gives the trace (Fig. 8) and the retrieval analysis.

Interaction efficiency. In Fig. 4(a), ASCENT's turn savings are larger later in the stream and persist on tasks solved by both ASCENT and frozen ReAct. On these shared successes, ASCENT uses fewer turns than frozen ReAct on 64% (4B) and 67% (9B) of tasks (Appendix C.2, Table 10).

Preliminary study on in-context and parametric co-evolution. ASCENT internalizes experience in fast weights, whereas in-context adaptation stores it in the context. In a preliminary study (Fig. 4(b)), we add ASCENT to ACE (Zhang et al., 2026a) and MemP (Fang et al., 2026) on the ALFWorld seen. The combination improves on each in-context method, and whether it also improves on ASCENT alone depends on model scale. At 9B, both combinations exceed ASCENT; at 4B, ASCENT alone remains the strongest, so parametric adaptation alone may suffice at this scale.



(a) Seen-stream diagnostics (b) Co-evolution (c) Distillation divergence

Figure 4: **ALFWorld seen-stream analyses.** (a) Mean ASCENT–ReAct success-rate gain and turns saved by task range and over all tasks (lighter bars 4B, darker 9B; shared-success turn comparison in Table 10). (b) Co-evolution of in-context and parametric adaptation. (c) Distillation-divergence ablation, pre-update SR (%) at both scales; forward KL is ASCENT's default.

	Qwen3.5-4B			Qwen3.5-9B		
Privileged information z_i (filter off / on)	**SR (%)**	**Turns**	**Adm. (%)**	**SR (%)**	**Turns**	**Adm. (%)**
Frozen ReAct	46.4	35.1	70.1	55.0	32.6	78.7
Action only: $a_{i,t}$	48.8 / 48.3	32.9 / 33.5	64.4 / 61.2	42.9 / 48.6	34.1 / 32.3	66.2 / 71.4
Reasoning + action + observation: $(w_{i,t}, o_{i,t+1})$	62.9 / 64.5	28.0 / 28.3	74.6 / **79.5**	**77.9** / 77.1	21.1 / 23.8	86.6 / **88.2**
Reasoning + action: $w_{i,t}$ (on = ASCENT)	60.0 / **69.5**	30.6 / **26.3**	74.1 / 78.0	75.3 / 77.4	21.6 / **21.0**	81.8 / 85.9

Table 3: Privileged-information ablation on ALFWorld seen (cells: filter off/on).

4.4 Ablation Studies

Ablation on privileged information. As shown in Table 3, action-only z_i leaves success near or below frozen ReAct, whereas adding reasoning consistently improves success by more than 11 points, lowers capped mean turns, and raises action admissibility. With reasoning in z_i, the filter further raises admissibility at both scales. Reasoning states the student's rationale for its actions, so it may give the teacher hindsight that the actions alone do not.

Effect of observation feedback. Adding observations to reasoning and action slightly improves success without the filter; with the filter, it lowers success at 4B and leaves it nearly unchanged at 9B (Table 3). Once the filter removes environment-rejected turns, observations may mainly add long state descriptions that dilute the reasoning and action evidence.

Ablation on distillation divergence. As shown in Fig. 4(c), ASCENT still improves on frozen ReAct at both scales when reverse KL or Jensen–Shannon (JS) divergence replaces forward KL in Eq. (9). Reverse KL trails forward KL at both scales, most at 9B; JS matches reverse KL at 4B but exceeds forward KL at 9B. Forward KL fits the full next-token distribution; reverse KL is mode-seeking (Agarwal et al., 2024) and may transfer less of the teacher's hindsight.

5 Conclusion

In our Online Agentic TTT study, directly imitating sampled tokens destabilizes the policy. ASCENT instead self-distills verified experience into persistent LoRA fast weights through a frozen teacher. On ALFWorld and WebShop at two model scales, it outperforms frozen ReAct and the compared baselines, with gains that grow later in the stream and transfer to held-out scenes.

Limitations and future work. ASCENT needs full next-token distributions from its frozen copy, so it applies only to open-weight models. It gates updates on sparse episode-level verification and weights every turn equally; future work could approximate finer-grained, turn-level credit. Because the teacher's privileged context contains the student's response text, it may partially reconstruct sampled tokens rather than provide independent hindsight; isolating this effect is an open question. The online evolution also shows potential for environment- or user-specific agents by plugging in the matching evolved fast weights.

AI USE STATEMENT

We used generative AI tools to support language editing, literature discovery, analysis-code development, and figure layout. The authors checked cited sources, mathematical arguments, code, numerical results, and all AI-assisted text and artifacts. The authors take responsibility for the final paper.

ETHICS STATEMENT

The experiments use simulated interactive benchmarks and involve no human participants or personal data. Persistent online adaptation can amplify systematic errors when deployed with unreliable feedback. Practical use should pair parameter updates with calibrated verification, monitoring, and rollback. Unlike text memories, weight updates are not individually inspectable or removable, which matters for privacy; crafted interactions that pass verification could also bias the adapted policy.

REPRODUCIBILITY STATEMENT

Sec. 3 specifies the online protocol and learning objective. Appendix A.1 derives the population selection and distillation targets and clarifies the limits of turn-length inference, while Appendix B.3 records privileged-context construction and Appendix B records the evaluation controls, optimizer, adapter settings, and full results. The supplementary material contains the analysis scripts, raw stability records, and figure-generation assets.

REFERENCES

Emre Can Acikgoz, Cheng Qian, Heng Ji, Dilek Hakkani-Tür, and Gokhan Tur. TT-SI: Self-Improving LLM Agents with Test-Time Training. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 9483–9508, 2026. doi: 10.18653/v1/2026.findings-acl.462. URL https://aclanthology.org/2026.findings-acl.462/.

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. In *International Conference on Learning Representations (ICLR)*, pp. 21246–21263, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/5be69a584901a26c521c2b51e40a4c20-Paper-Conference.pdf.

Lakshya A. Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J. Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alexandros G. Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. GEPA: Reflective prompt evolution can outperform reinforcement learning. In *International Conference on Learning Representations (ICLR)*, pp. 8479–8565, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/0e9e708b6f48e14fd0ac29e167413f76-Paper-Conference.pdf.

Salaheddin Alzubi, Noah Provenzano, Jaydon Bingham, Weiyuan Chen, and Tu Vu. EvoSkill: Automated Skill Discovery for Multi-Agent Systems. *arXiv preprint arXiv:2603.02766*, 2026.

Avrim Blum, Adam Kalai, and John Langford. Beating the Hold-Out: Bounds for K-Fold and Progressive Cross-Validation. In *Proceedings of the Twelfth Annual Conference on Computational Learning Theory*, pp. 203–208. Association for Computing Machinery, 1999. doi: 10.1145/307400.307439.

David Brandfonbrener, Alberto Bietti, Jacob Buckman, Romain Laroche, and Joan Bruna. When Does Return-Conditioned Supervised Learning Work for Offline Reinforcement Learning? In *Advances in Neural Information Processing Systems*, volume 35, pp. 1542–1553, 2022. doi: 10.52202/068431-0113.

Yifei Chen, Zihan Wang, Jialu Li, et al. Hint-sd: Hindsight-targeted self-distillation for multi-turn agents. *arXiv preprint arXiv:2605.17873*, 2026.

A. P. Dawid. Present Position and Potential Developments: Some Personal Views: Statistical Theory: The Prequential Approach. *Journal of the Royal Statistical Society: Series A (General)*, 147 (2):278–290, 1984. doi: 10.2307/2981683.

Benjamin Eysenbach, Soumith Udatha, Ruslan Salakhutdinov, and Sergey Levine. Imitating Past Successes Can Be Very Suboptimal. In *Advances in Neural Information Processing Systems*, volume 35, pp. 6047–6059, 2022. doi: 10.52202/068431-0438.

Runnan Fang, Yuan Liang, Xiaobin Wang, Jialong Wu, Shuofei Qiao, Pengjun Xie, Fei Huang, Huajun Chen, and Ningyu Zhang. Memp: Exploring Agent Procedural Memory. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 17490–17502, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.866. URL https://aclanthology.org/2026.findings-acl.866/.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-Group Policy Optimization for LLM Agent Training. In *Advances in Neural Information Processing Systems*, volume 38, pp. 46375–46408. Curran Associates, Inc., 2025. doi: 10.52202/085713-1544. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/420c9f777c0b4f78d515e53cf74d58b2-Paper-Conference.pdf.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, Wolfgang Macherey, Arnaud Doucet, Orhan Firat, and Nando de Freitas. Reinforced Self-Training (ReST) for Language Modeling. *arXiv preprint arXiv:2308.08998*, 2023.

Moritz Hardt and Yu Sun. Test-Time Training on Nearest Neighbors for Large Language Models. In *International Conference on Learning Representations (ICLR)*, 2024.

Anna Harutyunyan, Will Dabney, Thomas Mesnard, Mohammad Gheshlaghi Azar, Bilal Piot, Nicolas Heess, Hado P van Hasselt, Gregory Wayne, Satinder Singh, Doina Precup, and Remi Munos. Hindsight credit assignment. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/195f15384c2a79cedf293e4a847ce85c-Paper.pdf.

Zhenyu Hou, Yujiang Li, Jie Tang, and Yuxiao Dong. Single-Rollout Asynchronous Optimization for Agentic Reinforcement Learning. *arXiv preprint arXiv:2607.07508*, 2026.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Jinwu Hu, Zitian Zhang, Guohao Chen, Xutao Wen, Chao Shuai, Wei Luo, Bin Xiao, Yuanqing Li, and Mingkui Tan. Test-Time Learning for Large Language Models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 24823–24849. PMLR, 2025. URL https://proceedings.mlr.press/v267/hu25z.html.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement Learning via Self-Distillation. In *International Conference on Machine Learning (ICML)*, 2026. URL https://icml.cc/virtual/2026/poster/64121.

Haechan Kim, Yoonho Lee, Gisang Lee, Chelsea Finn, and Kangwook Lee. WHALE: A simple recipe for joint harness-weight optimization. *arXiv preprint arXiv:2609.00196*, 2026. URL https://arxiv.org/abs/2609.00196.

Junzhuo Liu, Weiwei Li, Jun Ling, and Peng Wang. When Privileged Guidance Misaligns: State-Matched Routing and Contextualized Self-Distillation for Multi-Turn Agents. *arXiv preprint arXiv:2608.05219*, 2026.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

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. *arXiv preprint arXiv:2605.15155*, 2026.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. doi: 10.1038/nature14236. URL https://doi.org/10.1038/nature14236.

Jingwei Ni, Yihao Liu, Xinpeng Liu, Yutao Sun, Mengyu Zhou, Pengyu Cheng, Dexin Wang, Erchao Zhao, Xiaoxi Jiang, and Guanjun Jiang. Trace2Skill: Distill trajectory-local lessons into transferable agent skills. *arXiv preprint arXiv:2603.25158*, 2026. URL https://arxiv.org/abs/2603.25158.

Siru Ouyang, Jun Yan, I-Hung Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long Le, Samira Daruki, Xiangru Tang, Vishy Tirumalashetty, George Lee, Mahsan Rofouei, Hangfei Lin, Jiawei Han, Chen-Yu Lee, and Tomas Pfister. ReasoningBank: Scaling Agent Self-Evolving with Reasoning Memory. In *International Conference on Learning Representations*, volume 2026, pp. 94327–94354, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/980ea04d23d1f6908964eba2a74afe45-Paper-Conference.pdf.

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

Tao Ren, Weiyao Luo, Hui Yang, Rongzhi Zhu, Xiang Huang, Yuchuan Wu, Bingxue Chou, Jieping Ye, Jiafeng Liang, Yongbin Li, and Yijie Peng. Scaling Self-Evolving Agents via Parametric Memory. *arXiv preprint arXiv:2606.04536*, 2026.

Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.

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. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-Distillation Enables Continual Learning. In *International Conference on Machine Learning (ICML)*, 2026. URL https://icml.cc/virtual/2026/poster/61434.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language Agents with Verbal Reinforcement Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 8634–8652, 2023. doi: 10.52202/075280-0377.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. In *International Conference on Learning Representations (ICLR)*, 2021.

Avi Singh, John D. Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J. Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron Parisi, Abhishek Kumar, Alex Alemi, Alex Rizkowsky, Azade Nova, Ben Adlam, Bernd Bohnet, Gamaleldin Elsayed, Hanie Sedghi, Igor Mordatch, Isabelle Simpson, Izzeddin Gur, Jasper Snoek, Jeffrey Pennington, Jiri Hron, Kathleen Kenealy, Kevin Swersky, Kshiteej Mahajan, Laura Culp, Lechao Xiao, Maxwell L. Bileschi, Noah Constant, Roman Novak, Rosanne Liu, Tris Warkentin, Yundi Qian, Yamini Bansal, Ethan Dyer, Behnam Neyshabur, Jascha Sohl-Dickstein, and Noah Fiedel. Beyond Human Data: Scaling Self-Training for Problem-Solving with Language Models. *Transactions on Machine Learning Research*, 2024.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. Test-Time Training with Self-Supervision for Generalization under Distribution Shifts. In *Proceedings of*

the 37th International Conference on Machine Learning, volume 119 of *Proceedings of Machine Learning Research*, pp. 9229–9248. PMLR, 2020. URL https://proceedings.mlr.press/v119/sun20b.html.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

Mirac Suzgun, Mert Yuksekgonul, Federico Bianchi, Dan Jurafsky, and James Zou. Dynamic Cheatsheet: Test-Time Learning with Adaptive Memory. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7080–7106, Rabat, Morocco, March 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.eacl-long.333. URL https://aclanthology.org/2026.eacl-long.333/.

Hui-Ze Tan, Xiao-Wen Yang, Hao Chen, Jie-Jing Shao, Yi Wen, Yuteng Shen, Weihong Luo, Xiku Du, Lan-Zhe Guo, and Yu-Feng Li. Hindsight Credit Assignment for Long-Horizon LLM Agents. *arXiv preprint arXiv:2603.08754*, 2026. URL https://arxiv.org/abs/2603.08754.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. AppWorld: A controllable world of apps and people for benchmarking interactive coding agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16022–16076, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.850. URL https://aclanthology.org/2024.acl-long.850/.

Aozhe Wang, Zhengxi Lu, Jianze Wang, Shangke Lv, Ying Liu, Weiming Lu, Jun Xiao, Yueting Zhuang, Hua Yang, Qianglong Chen, and Yongliang Shen. TTPO: Test-Time Policy Optimization. *arXiv preprint arXiv:2608.27448*, 2026a.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully Test-Time Adaptation by Entropy Minimization. In *International Conference on Learning Representations (ICLR)*, 2021.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An Open-Ended Embodied Agent with Large Language Models. *Transactions on Machine Learning Research*, 2024.

Hao Wang, Guozhi Wang, Han Xiao, Yufeng Zhou, Yue Pan, Jichao Wang, Ke Xu, Yafei Wen, Xiaohu Ruan, Xiaoxin Chen, and Honggang Qi. Skill-SD: Skill-Conditioned Self-Distillation for Multi-turn LLM Agents. *arXiv preprint arXiv:2604.10674*, 2026b.

Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual Test-Time Domain Adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7191–7201, 2022. doi: 10.1109/CVPR52688.2022.00706.

Yanbo Wang, Jinhua Hao, Yuze Shi, Kun Yuan, and Ming Sun. No Time Like the Present: Agentic Test-Time Training for LLM Agents. *arXiv preprint arXiv:2607.03441*, 2026c. URL https://arxiv.org/abs/2607.03441.

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. *arXiv preprint arXiv:2608.05987*, 2026d.

Jinyang Wu, Shuo Yang, Zhengxi Lu, Fan Zhang, Yuhao Shen, Lang Feng, Haoran Luo, Zheng Lian, Shuai Zhang, Zhengqi Wen, and Jianhua Tao. SEED: Self-Evolving On-Policy Distillation for Agentic Reinforcement Learning. *arXiv preprint arXiv:2607.14777*, 2026.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-Mem: Agentic memory for LLM agents. In *Advances in Neural Information Processing Systems*, volume 38, pp. 17577–17604. Curran Associates, Inc., 2025. doi: 10.52202/085713-0593. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/19909c36f51abc4856b4560aff3d36d6-Paper-Conference.pdf.

Zixiang Yang, Rui Zheng, Mengdi Li, Hao Sun, and Yang Yuan. Claas: Continuously learning to be an adversarial sparring partner. *arXiv preprint arXiv:2606.05559*, 2026.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. WebShop: Towards Scalable Real-World Web Interaction with Grounded Language Agents. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 20744–20757, 2022. doi: 10.52202/068431-1508.

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 (ICLR)*, 2023.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. Scaling Relationship on Learning Mathematical Reasoning with Large Language Models. *arXiv preprint arXiv:2308.01825*, 2023. URL https://arxiv.org/abs/2308.01825.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Pan Lu, Zhi Huang, Carlos Guestrin, and James Zou. Optimizing Generative AI by Backpropagating Language Model Feedback. *Nature*, 639(8055):609–616, 2025. doi: 10.1038/s41586-025-08661-4.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping Reasoning With Reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Siliang Zeng, Quan Wei, William Brown, Oana Frunza, Yuriy Nevmyvaka, and Mingyi Hong. Reinforcing Multi-Turn Reasoning in LLM Agents via Turn-Level Credit Assignment. *arXiv preprint arXiv:2505.11821*, 2025. URL https://arxiv.org/abs/2505.11821v1.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, Urmish Thakker, James Y Zou, and Kunle Olukotun. Agentic Context Engineering: Evolving Contexts for Self-Improving Language Models. In *International Conference on Learning Representations*, volume 2026, pp. 86069–86100, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/8a94ff6f922d995d7d3f4ebf4143e442-Paper-Conference.pdf.

Shengtao Zhang, Jiaqian Wang, Ruiwen Zhou, Junwei Liao, Yuchen Feng, Zhuo Li, Yujie Zheng, Weinan Zhang, Ying Wen, Zhiyu Li, Feiyu Xiong, Yutao Qi, Bo Tang, and Muning Wen. MemRL: Self-Evolving Agents via Runtime Reinforcement Learning on Episodic Memory. *arXiv preprint arXiv:2601.03192*, 2026b.

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. doi: 10.1609/aaai.v38i17.29936.

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. In *International Conference on Machine Learning (ICML)*, 2026. URL https://icml.cc/virtual/2026/poster/64784.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, Biqing Qi, Youbang Sun, Zhiyuan Ma, Lifan Yuan, Ning Ding, and Bowen Zhou. TTRL: Test-Time Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 131459–131483. Curran Associates, Inc., 2025. doi: 10.52202/085713-4376. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/be690ea16f005c174f6c4102a5970e67-Paper-Conference.pdf.

Adam Zweiger, Jyo Pari, Han Guo, Yoon Kim, and Pulkit Agrawal. Self-Adapting Language Models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 74084–74115. Curran Associates, Inc., 2025. doi: 10.52202/085713-2483. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/6b41e04c41726e2a60e456d0a2b961ab-Paper-Conference.pdf.

A Additional Method Details

A.1 What Verifier Selection and Distillation Identify

We separate what the episode verifier selects from what the distillation objective fits. Fix a task x and let $R(\tau) = \mathcal{V}(x, \tau) \in \{0, 1\}$ denote verifier acceptance for an update, which need not coincide with strict benchmark success. The results below concern the population of possible trajectories under a fixed collection policy. ASCENT observes only one rollout per task in its one-pass stream.

A.1.1 Verifier Selection Reweights Completed Trajectories

Let π be the response policy used to collect the trajectory, including its decoding rule, and let $\mathbb{P}_\pi(\tau \mid x)$ be the resulting trajectory law with the environment. If $J_x(\pi) := \Pr_{\pi,\mathcal{E}}(R = 1 \mid x) > 0$, trajectories selected for an update follow

$$\mathbb{P}^+_\pi(\tau \mid x) = \frac{R(\tau)\mathbb{P}_\pi(\tau \mid x)}{J_x(\pi)}. \tag{15}$$

At a pre-decision history $S = H_t$, let $V^\pi(S) = \Pr_\pi(R = 1 \mid S)$. Let $Q^\pi(S, u)$ be the probability of acceptance after choosing complete response u and then following π. When $V^\pi(S) > 0$, Bayes' rule gives

$$\pi^+(u \mid S) := \Pr_\pi(u \mid S, R = 1) = \pi(u \mid S)\frac{Q^\pi(S, u)}{V^\pi(S)}. \tag{16}$$

Verifier selection therefore reweights responses in the distribution of accepted trajectories (Brandfonbrener et al., 2022; Eysenbach et al., 2022). It neither assigns causal credit to turns within one rollout nor reveals the outcome of an untried response. With only one collected response at a given history, this selection cannot rank alternatives at that history. The hindsight-conditioned teacher supplies soft predictions at the saved student prefixes. Whether those predictions improve later tasks must be tested empirically.

A.1.2 Forward KL Fits Teacher Predictions at Student Prefixes

Now consider the student prefixes used to fit the teacher. Keep the collection policy π fixed for task x. For a selected trajectory τ, let $\mathcal{I}(\tau)$ contain the nonempty turns in the student loss, and let $L_{\tau t}$ be the number of generated tokens at turn t. Assume $|\mathcal{I}(\tau)| > 0$ for selected trajectories; otherwise condition the selected law on that event. The equal-turn, equal-token weighting in Eq. (9) gives

$$\nu_\pi(\mathrm{d}\tau, t, j \mid x, R = 1) = \mathbb{P}^+_\pi(\mathrm{d}\tau \mid x)\frac{\mathbf{1}\{t \in \mathcal{I}(\tau)\}}{|\mathcal{I}(\tau)|}\frac{\mathbf{1}\{1 \le j \le L_{\tau t}\}}{L_{\tau t}}. \tag{17}$$

Draw an index $Z = (\tau, t, j)$ from this measure. Write $c_Z = c_{\tau,t,j}$ for the saved student prefix defined in Sec. 3.2, and $z(\tau)$ for the privileged information defined in Eq. (13). At this prefix, the frozen teacher supplies $q_Z(\cdot) = p_{\theta_0}(\cdot \mid z(\tau) \oplus c_Z)$. For a candidate adapter ϕ, the population objective is

$$\mathcal{L}_x(\phi; \pi) = \mathbb{E}_{Z\sim\nu_\pi}\left[D_{\mathrm{KL}}(q_Z \,\|\, p_{\theta_0,\phi}(\cdot \mid c_Z))\right]. \tag{18}$$

Define the mean teacher distribution at a retained prefix by $\bar{q}(\cdot \mid c) = \mathbb{E}_{\nu_\pi}[q_Z(\cdot) \mid c_Z = c]$.

Lemma A.1 (Forward-KL population target)**.** *For any student distribution on the retained prefixes,*

$$\begin{aligned}\mathcal{L}_x(\phi; \pi) = {} & \mathbb{E}_{Z\sim\nu_\pi}\left[D_{\mathrm{KL}}(q_Z \,\|\, \bar{q}(\cdot \mid c_Z))\right] \\ & + \mathbb{E}_{Z\sim\nu_\pi}\left[D_{\mathrm{KL}}(\bar{q}(\cdot \mid c_Z) \,\|\, p_{\theta_0,\phi}(\cdot \mid c_Z))\right].\end{aligned} \tag{19}$$

Proof. Condition on $c_Z = c$, expand the KL terms over next tokens, and use $\bar{q}(\cdot \mid c) = \mathbb{E}[q_Z(\cdot) \mid c_Z = c]$. The terms involving the student distribution then agree on both sides; averaging over retained prefixes proves Eq. (19). □

The first term in Eq. (19) does not depend on ϕ. An unrestricted minimizer therefore matches $\bar{q}$ at prefixes covered by the training measure. The student's responses determine which prefixes are visited and how they are weighted; the hindsight-conditioned teacher determines their soft targets. This identity characterizes the target, not whether the teacher's predictions help, a finite adapter reaches that target, or the fitted behavior transfers to later tasks.

A.1.3 Verification Does Not Identify Shorter Paths

An accepted trajectory can contain valid but unnecessary turns. Suppose *finish* succeeds now while *wait* leads to success on the next turn. Both responses can have $Q^\pi(S, u) = 1$, so verifier selection gives neither response a preference. The admissibility filter also keeps both if the environment accepts them.

Let T be the episode length under π. The selected population obeys

$$\mathbb{E}_\pi[T \mid R = 1] - \mathbb{E}_\pi[T] = \frac{\mathrm{Cov}_\pi(T, R)}{J_x(\pi)}. \tag{20}$$

The sign of this difference depends on the covariance: verification alone does not imply shorter accepted trajectories. In a capped environment, unsuccessful episodes may run to the turn limit. An all-episode reduction in mean turns can therefore reflect more successes, even without shorter paths on successful tasks. Appendix C.2 also compares tasks solved by both policies. That comparison describes turn differences on shared successes but does not identify which individual turns the updated policy learned to avoid.

B Additional Experimental Details

B.1 Evaluation Protocol and Comparison Controls

Within each paired comparison, methods use the same task order. After each scored episode, every task-online method receives the same task-provided verification signal and may use it to update its memory, skills, or model weights. All in-context adaptation methods start from the same ReAct template, which ASCENT also uses for rollout collection. Offline self-evolution methods instead build their adaptation state before evaluation, using the same fixed-size training split. These controls match the task stream, prompt template, and offline data volume while allowing the methods to learn in different ways. On WebShop, exact success requires a task score of 1, whereas ASCENT's adaptation gate accepts a score of at least 0.9.

B.2 Baseline Details

Frozen policy. ReAct (Yao et al., 2023) uses the shared interaction template and makes no cross-task update.

Offline in-context adaptation. These methods construct memories or skills from the fixed training split before evaluation: MemP (Fang et al., 2026), ACE (Zhang et al., 2026a), EvoSkill (Alzubi et al., 2026), GEPA (Agrawal et al., 2026), TextGrad (Yuksekgonul et al., 2025), and Trace2Skill (Ni et al., 2026).

Task-online in-context adaptation. These methods update memory or skills after a scored episode; the new context can first affect the next task. We evaluate MemP (Fang et al., 2026), ReasoningBank (Ouyang et al., 2026), ACE (Zhang et al., 2026a), A-Mem (Xu et al., 2025), MemRL (Zhang et al., 2026b), and Dynamic Cheatsheet (Suzgun et al., 2026).

Task-online parametric baselines. TT-OPSD adapts OPSD (Zhao et al., 2026) to the one-pass task stream. Its teacher receives the turn-local privileged information $z_{i,t}^{\mathrm{TT}} = (x_i, w_{i,t}, o_{i,t+1})$ together with the saved student prefix $c_{i,t,j}$. Thus the additional teacher information describes the current response and returned observation, rather than the completed trajectory. TT-OPSD retains environment-rejected turns in selected episodes; its episode gate, saved prefixes, forward-KL objective, and optimizer match ASCENT. It provides a task-online parametric comparison with a narrower privileged view of the completed interaction. Sec. 3.2 also evaluates Online SFT, Online RFT (Yuan et al., 2023; Shao et al., 2024), and Online RFT with reference KL.

B.3 Implementation Details

Privileged context and prompt placement. In ALFWorld, we set $d_{i,t} = 1$ when the parsed action belongs to the environment's current available-action list; an unparsed or unavailable action receives $d_{i,t} = 0$. This check does not separately test for execution errors. The saved prefix $c_{i,t,j}$ contains

any retrieved information used during rollout collection. At the model interface, the student reads $c_{i,t,j}$ and the teacher reads $z_i \oplus c_{i,t,j}$, with teacher token positions shifted by $|z_i|$. The trajectory text in z_i conditions the teacher but is not itself a prediction target. Fig. 5 shows this prompt arrangement for reasoning–action responses. The same 50-turn cap applies to ALFWorld and WebShop rollouts during evaluation and online training. If a task remains unfinished at turn 50, the rollout is terminated and scored as a failure.

Optimization configuration. The teacher is an immutable copy of the initial base model θ_0 and provides detached full-vocabulary logits. Equivalently, teacher inference disables the student's LoRA adapter. We optimize the forward-KL loss with AdamW (Loshchilov & Hutter, 2019), taking two optimizer steps per selected episode. The adapter and optimizer state carry across tasks; the trajectory is discarded after its update. We apply LoRA to `down_proj` modules with rank 16, scale 32, and learning rate 1.5×10^{-4}. All experiments run on a single NVIDIA H100 GPU.

B.4 Full Benchmark Results

Tables 4–7 report the task-family, turn, and timing results behind the main-paper summaries.

B.5 Additional Evaluation on AppWorld

AppWorld tests interactive code generation across simulated applications (Trivedi et al., 2024). We evaluate Qwen3.5-4B on Test-Normal (168 tasks in 56 scenarios) and Test-Challenge (417 tasks in 139 scenarios). Each task has one scored attempt with a 30-interaction budget; task-online methods receive episode verification only after that attempt. Task Goal Completion (TGC) counts tasks passing every state-based test. Scenario Goal Completion (SGC) counts scenarios in which every task variant succeeds. Table 8 reports these official metrics under Online Agentic TTT.

	Test-Normal		Test-Challenge	
Method	**TGC**	**SGC**	**TGC**	**SGC**
ReAct (frozen)	20.2	5.4	15.8	**5.8**
MemP	22.0±2.1	1.8±0.2	16.1±3.3	**5.8**±0.7
ReasoningBank	19.6±1.9	3.6±0.7	19.2±2.8	4.3±1.1
ACE	22.0±4.1	3.6±1.2	14.4±3.9	2.2±2.0
A-Mem	14.3±0.4	8.9±1.2	11.3±1.1	2.2±0.4
Dynamic Cheatsheet	18.5±2.3	1.8±0.6	14.9±1.4	2.9±0.5
TT-OPSD	19.1±3.2	5.7±0.8	16.2±2.0	3.8±1.2
ASCENT	**26.2**±1.3	**9.5**±0.9	**20.0**±2.5	5.3±0.9

Table 8: **AppWorld task-online results.** Task-goal completion (TGC, %) and scenario-goal completion (SGC, %) on Test-Normal and Test-Challenge with Qwen3.5-4B.

Task-level gains extend to code-mediated interaction. ASCENT improves TGC over frozen ReAct on both splits (Table 8) and leads the displayed comparators on Test-Normal TGC and SGC. On Test-Challenge, its task-level gain does not translate into a scenario-level lead.

C Additional Analyses and Results

C.1 Teacher-Target Tracking and Online Stability

ASCENT keeps the teacher model fixed while the student changes after selected episodes. We examine this choice by allowing the teacher weights to follow the student. Let $\bar{\phi}_i$ be the teacher adapter used before the update on task i, and let $Q_{\bar{\phi}_i}(\tau_i)$ denote its token-level targets on the student's saved prefixes, conditioned on the completed trajectory. For an episode selected by the verifier, and suppressing token indices, the updates have the form

$$\tau_i \sim P_{\phi_i}, \qquad q_i = Q_{\bar{\phi}_i}(\tau_i), \qquad \phi_{i+1} = \mathsf{Evolve}\big(\phi_i; \mathcal{L}_i^{\mathrm{ASCENT}}[q_i]\big). \tag{21}$$

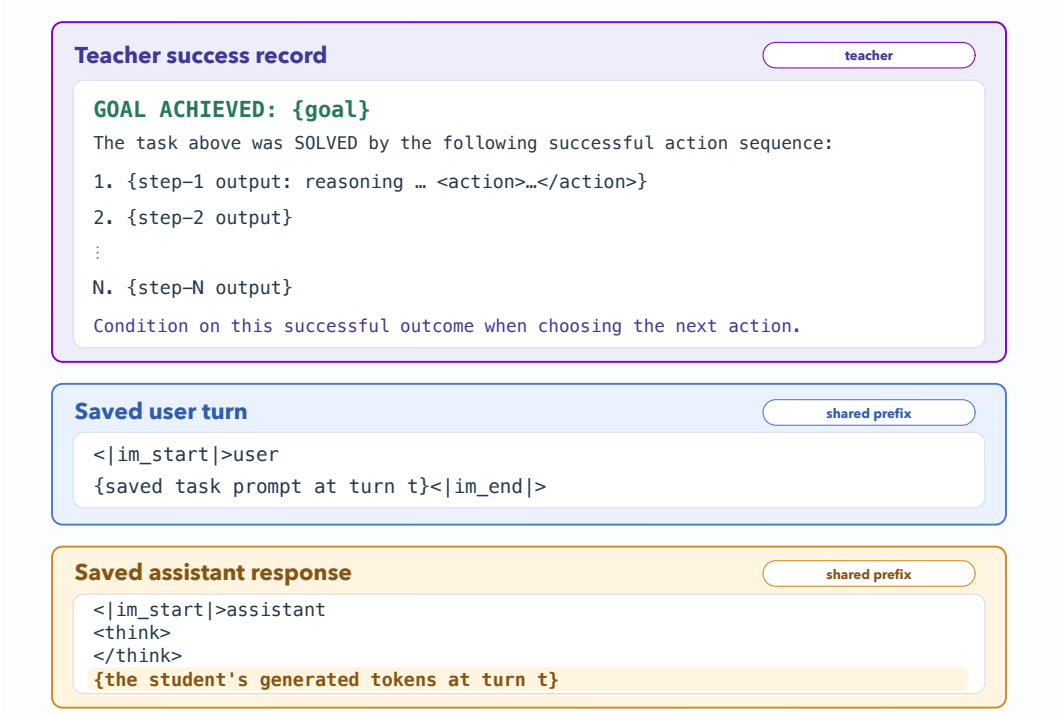


Figure 5: **Teacher conditioning.** After verifier acceptance, privileged hindsight z_i is prepended to each saved student prefix. The empty think block reflects disabled thinking mode; each response's reasoning is the visible ReAct rationale before the action tag.

Here Evolve is the update in Eq. (10), and $\mathcal{L}_i^{\text{ASCENT}}[q_i]$ is Eq. (9) with targets q_i. ASCENT sets $\bar{\phi}_i = 0$ and thus always uses the initial base model as teacher. The diagnostics instead use an exponential moving average (EMA) or a delayed student snapshot. A fixed teacher still produces different targets on different episodes because its privileged information and saved prefixes change. Evolving the teacher also changes the rule $Q_{\bar{\phi}_i}$ that turns this evidence into supervision.

We study these variants with Qwen3.5-4B LoRA on the 140-task ALFWorld seen stream. The frozen row of Table 9 reports the ASCENT result from Table 1. For each run, the first low-success task is the first point where success over the preceding 20 tasks falls to 15% or below, even if it later recovers. The final two rows are sampled-token controls from Sec. 3.2, not privileged-teacher variants.

Faster teacher tracking strengthens the feedback loop. Student updates change the next trajectories through P_ϕ. If the teacher also tracks the student, those updates change $Q_{\bar{\phi}}$, so student drift can return as supervision on later tasks. Stop-gradient separates teacher and student within an update but does not remove this cross-task path. In Fig. 6a–b, the fastest EMA and shorter snapshot delays lose success earlier than slower targets. The matched EMA-0.99 condition remains stable across its three runs, though below the frozen-teacher result in Table 1. Thus teacher evolution is not uniformly unstable. The mostly single-run schedule ladder shows an ordering of outcomes, not a critical stability threshold. Holding a target separate from a changing policy plays a related role in deep Q-learning (Mnih et al., 2015).

For EMA coefficient β, the nominal memory spans roughly $(1-\beta)^{-1}$ accepted updates, not episodes. The verifier gate controls when both student and teacher update, so their lag in episode time also depends on the policy's success rate. After success declines, fewer updates may keep the system in a poor state. The observed low-success tasks are therefore outcomes of the coupled stream, not fixed onset points for a particular β or snapshot delay L.

Verifier selection leaves turn-level credit unresolved. The verifier selects episodes for training but does not identify which turns helped complete the task. A selected trajectory may still contain detours or weak decisions. The admissibility mask removes turns with rejected actions from privileged

Method	Pick	Look	Clean	Heat	Cool	Pick2	Avg. SR	Mean turns (all)	Time h / s
Qwen3.5-4B									
Frozen									
ReAct	91.4	46.2	40.7	6.2	8.0	54.2	46.4	35.1	97
Offline									
MemP	94.3	53.8	**77.8**	0.0	36.0	66.7	61.4±6.5	28.6	5.2 / 61
ACE	85.7	46.2	70.4	6.2	60.0	58.3	60.7±9.2	31.1	6.3 / 81
EvoSkill	93.3	17.9	35.8	6.2	32.0	66.7	49.8±4.9	34.4	5.9 / 123
GEPA	**100.0**	23.1	22.2	**37.5**	**64.0**	66.7	58.6±3.4	34.5	6.3 / 91
TextGrad	88.6	46.2	22.2	6.2	12.0	50.0	42.1±4.9	37.5	8.3 / 104
Trace2Skill	94.3	**69.2**	29.6	0.0	16.0	**75.0**	51.4±6.1	33.0	5.2 / 92
Online									
MemP	94.3	61.5	71.6	12.5	36.0	73.6	63.6±5.9	29.7	58
ReasoningBank	94.3	30.8	33.3	14.6	22.7	61.1	49.0±1.5	35.6	89
ACE	90.5	17.9	49.4	22.9	42.7	41.7	51.2±13.8	34.6	92
A-Mem	88.6	30.8	35.8	10.4	20.0	58.3	46.7±5.4	34.5	79
MemRL	85.7	23.1	33.3	12.5	16.0	62.5	45.0±5.2	35.4	89
Dynamic Cheatsheet	91.4	23.1	44.4	6.2	28.0	54.2	48.6±4.1	36.9	94
TT-OPSD	95.2	46.2	29.6	2.1	25.3	72.2	51.0±3.7	32.9	87
ASCENT	99.0	**69.2**	69.1	20.8	54.7	**75.0**	**69.5**±5.6	**26.3**	115
Qwen3.5-9B									
Frozen									
ReAct	94.3	61.5	44.4	0.0	8.0	91.7	55.0	32.6	115
Offline									
MemP	**100.0**	**92.3**	74.1	18.8	28.0	83.3	69.3±2.4	26.3	5.9 / 83
ACE	94.3	53.8	51.9	**37.5**	28.0	79.2	61.4±5.5	27.9	7.1 / 129
EvoSkill	97.1	71.8	54.3	18.8	37.3	90.3	65.7±3.5	29.0	4.7 / 145
GEPA	**100.0**	76.9	66.7	25.0	10.0	91.7	65.4±4.6	30.1	6.7 / 115
TextGrad	91.4	61.5	59.3	18.8	44.0	54.2	59.3±4.5	32.6	7.0 / 147
Trace2Skill	94.3	84.6	44.4	31.2	**60.0**	79.2	67.9±6.4	28.4	6.4 / 132
Online									
MemP	95.2	74.4	75.3	16.7	38.7	80.6	67.9±2.9	27.4	85
ReasoningBank	95.2	59.0	75.3	22.9	36.0	76.4	66.0±2.4	29.0	93
ACE	93.3	51.3	**76.5**	29.2	57.3	75.0	69.3±6.3	28.3	113
A-Mem	95.7	69.2	53.7	9.4	22.0	85.4	60.4±0.4	30.7	101
MemRL	97.1	53.8	66.7	12.5	20.0	**95.8**	63.6±2.5	29.5	110
Dynamic Cheatsheet	97.1	61.5	48.1	0.0	20.0	83.3	57.1±4.8	32.2	126
TT-OPSD	98.1	79.5	54.3	12.5	5.3	**95.8**	61.2±1.8	29.3	99
ASCENT	**100.0**	89.7	**76.5**	**37.5**	48.0	**95.8**	**77.4**±1.8	**21.0**	133

Table 4: **ALFWorld seen-stream results, including offline methods.** Success rate (SR, %) by task type and over the full stream, mean turns per episode, and runtime. For offline methods, runtime is reported as training GPU-hours / evaluation seconds per episode; frozen and online methods report seconds per episode. Bold indicates the highest SR (including ties) and fewest mean turns within each backbone.

information, but it cannot judge the remaining valid turns. With a tracking teacher, those turns can also influence the model that supplies targets for later episodes. Neither outcome gating nor action filtering removes this second feedback path.

Frozen and evolving teachers define different targets. At a fixed student prefix c, let $\mu_\phi^+(z \mid c)$ be the distribution of privileged information from selected on-policy trajectories. For an unrestricted student under a fixed training measure, forward KL to the frozen teacher has the centroid solution

$$p^*(\cdot \mid c) = \mathbb{E}_{z \sim \mu_\phi^+(\cdot|c)} \left[Q_0(\cdot \mid z, c) \right]. \tag{22}$$

The selected z can change as the student evolves, but the fixed Q_0 always interprets it. With a live teacher, both the selected experience and the teacher depend on the student. A self-consistent solution then satisfies

$$p_{\phi^*}(\cdot \mid c) = \mathbb{E}_{z \sim \mu_{\phi^*}^+(\cdot|c)} \left[Q_{\phi^*}(\cdot \mid z, c) \right]. \tag{23}$$

This is a self-consistency condition, not a convergence result; a policy can satisfy it without succeeding in the environment. The teacher still sees z, so it need not match the unconditioned student. But

Method	Pick	Look	Clean	Heat	Cool	Pick2	**Avg.SR**	Mean turns (all)	Time s/ep
Qwen3.5-4B									
Frozen									
ReAct	87.5	38.9	29.0	8.7	28.6	76.5	43.3	36.2	107
Offline									
MemP (offline)	91.7	72.2	64.5	8.7	52.4	76.5	60.4±4.5	30.0	69
ACE (offline)	79.2	33.3	41.9	8.7	42.9	23.5	39.6±3.2	38.5	106
EvoSkill	83.3	40.7	29.0	5.8	47.6	58.8	43.0±4.6	37.4	146
GEPA	95.8	38.9	38.7	**47.8**	**76.2**	70.6	60.4±4.9	32.8	95
TextGrad	79.2	66.7	22.6	8.7	19.0	47.1	38.8±6.4	38.6	112
Trace2Skill	79.2	50.0	16.1	4.3	19.0	76.5	38.1±5.3	39.4	127
Task-online-built artifacts (frozen on unseen)									
MemP	87.5	**88.9**	**80.6**	4.3	42.9	29.4	57.5±6.9	31.1	66
ReasoningBank	87.5	16.7	41.9	21.7	52.4	47.1	45.5±7.6	36.0	91
ACE	70.8	11.1	3.2	0.0	4.8	35.3	20.1±7.4	45.2	134
A-Mem	91.7	38.9	41.9	8.7	42.9	58.8	47.0±9.0	34.5	84
MemRL	79.2	38.9	45.2	17.4	33.3	70.6	47.0±6.1	35.6	88
Dynamic Cheatsheet	79.2	27.8	12.9	13.0	38.1	**82.4**	39.6±7.9	38.7	102
TT-OPSD	**98.6**	66.7	35.5	5.8	33.3	70.6	50.0±4.8	33.9	67
ASCENT	86.1	46.3	72.0	20.3	69.8	78.4	**62.7**±7.5	**28.4**	94
Qwen3.5-9B									
Frozen									
ReAct	91.7	88.9	38.7	4.3	0.0	88.2	49.3	34.3	138
Offline									
MemP (offline)	**100.0**	**100.0**	71.0	21.7	33.3	82.4	67.2±6.7	27.8	94
ACE (offline)	83.3	66.7	41.9	39.1	42.9	88.2	58.2±7.3	30.9	141
EvoSkill	95.8	70.4	46.2	23.2	38.1	90.2	58.7±4.3	31.9	163
GEPA	**100.0**	83.3	58.1	23.9	11.9	82.4	59.0±9.0	30.8	139
TextGrad	**100.0**	77.8	64.5	17.4	85.7	64.7	67.9±4.7	30.3	173
Trace2Skill	**100.0**	94.4	41.9	34.8	57.1	70.6	64.2±5.1	32.4	157
Task-online-built artifacts (frozen on unseen)									
MemP	**100.0**	83.3	71.0	13.0	76.2	82.4	70.1±4.3	26.3	84
ReasoningBank	95.8	66.7	67.7	26.1	28.6	64.7	59.0±8.1	32.8	120
ACE	83.3	66.7	19.4	17.4	71.4	82.4	53.0±5.9	35.1	151
A-Mem	95.8	72.2	45.2	26.1	47.6	88.2	60.4±7.1	32.0	111
MemRL	**100.0**	88.9	54.8	30.4	57.1	88.2	67.9±6.6	29.5	107
Dynamic Cheatsheet	95.8	61.1	19.4	21.7	9.5	88.2	46.3±5.2	35.4	151
TT-OPSD	**100.0**	72.2	48.4	8.7	4.8	**96.1**	53.2±5.0	31.8	89
ASCENT	97.2	81.5	**90.3**	**72.5**	**96.8**	94.1	**88.8**±6.6	**17.0**	74

Table 5: **ALFWorld held-out transfer.** Success rate (SR, %) by task type and over the full held-out stream, mean turns per episode, and runtime (s/episode). Seen-stream artifacts remain fixed. Bold indicates the highest SR (including ties) and fewest mean turns within each backbone.

the policy being trained determines both the selected information and its interpretation. Sampled-token self-imitation is the extreme case: one-hot targets provide no probability mass to alternative tokens.

Low distillation loss can accompany poor interaction. Forward KL measures agreement with the teacher, not task success. In Fig. 6c, some low-success conditions also have low target loss. Because targets differ across conditions, their loss values do not share a single calibration to success. Within runs with loss logs, the median correlation between update loss and trailing success is near zero,

Method	Pick	Look	Clean	Heat	Cool	Pick2	**Avg.SR** (Δ vs Transfer)	Mean turns (all)	Time s/ep
Qwen3.5-4B									
Frozen									
ReAct	87.5	38.9	29.0	8.7	28.6	76.5	43.3	36.2	107
Online adaptation on unseen stream									
MemP	**100.0**	**94.4**	74.2	21.7	33.3	64.7	64.9±4.8 (+7.4)	28.2	58
ReasoningBank	87.5	33.3	48.4	26.1	52.4	52.9	50.7±7.3 (+5.2)	35.0	89
ACE	75.0	11.1	3.2	0.0	9.5	35.3	21.6±5.2 (+1.5)	45.5	225
A-Mem	87.5	55.6	51.6	21.7	28.6	52.9	50.0±3.9 (+3.0)	33.5	85
MemRL	79.2	38.9	51.6	21.7	42.9	47.1	47.8±8.7 (+0.8)	34.3	86
Dynamic Cheatsheet	79.2	61.1	29.0	0.0	23.8	35.3	37.3±6.1 (−2.3)	37.5	99
TT-OPSD	97.2	81.5	39.8	15.9	28.6	**90.2**	56.2±3.1 (+6.2)	31.5	90
ASCENT	90.3	83.3	**83.9**	**53.6**	**77.8**	82.4	**78.6**±5.1 (+15.9)	**21.1**	98
Qwen3.5-9B									
Frozen									
ReAct	91.7	88.9	38.7	4.3	0.0	88.2	49.3	34.3	138
Online adaptation on unseen stream									
MemP	**100.0**	**100.0**	83.9	13.0	76.2	94.1	76.9±5.2 (+6.8)	24.9	74
ReasoningBank	95.8	66.7	58.1	17.4	38.1	70.6	57.5±4.8 (−1.5)	33.1	124
ACE	91.7	55.6	38.7	13.0	66.7	76.5	55.2±7.1 (+2.2)	33.2	142
A-Mem	83.3	72.2	64.5	21.7	33.3	82.4	59.0±3.7 (−1.4)	31.7	116
MemRL	95.8	94.4	74.2	21.7	52.4	94.1	70.9±7.2 (+3.0)	27.0	99
Dynamic Cheatsheet	95.8	61.1	54.8	17.4	47.6	88.2	59.7±8.6 (+13.4)	29.1	121
TT-OPSD	97.2	85.2	46.2	13.0	4.8	**96.1**	54.7±0.9 (+1.5)	30.3	111
ASCENT	98.6	92.6	**88.2**	**72.5**	**95.2**	**96.1**	**90.0**±3.6 (+1.2)	**15.7**	107

Table 6: **ALFWorld continued TTT.** Pre-update success rate (SR, %) by task type and over the held-out stream, mean turns per episode, and runtime (s/episode). Δ is the change from frozen transfer in Table 5. Bold indicates the highest SR (including ties) and fewest mean turns within each backbone.

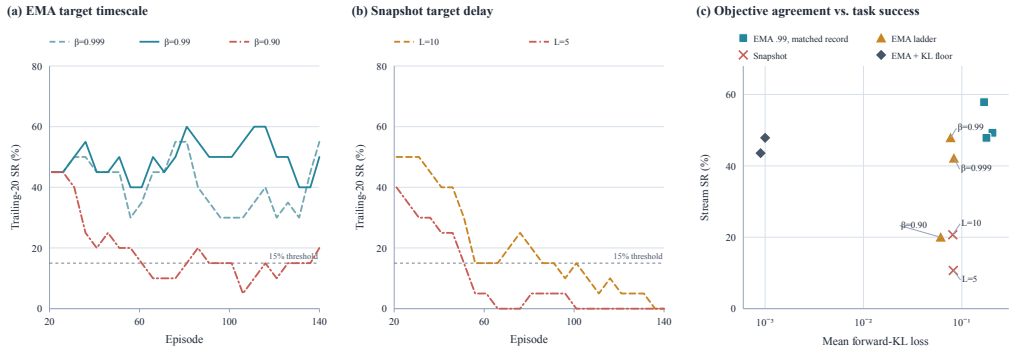


Figure 6: **Teacher-target tracking over the task stream.** (a) EMA teachers and (b) delayed student snapshots, with trailing-window pre-update success; (c) distillation loss versus stream success.

although the correlations vary and include only episodes selected for an update. Forward KL can cover only alternatives retained by its teacher. In the zero-lag sampled-token controls, fixed-prompt student entropy approaches zero as drift from the initial policy grows. Teacher entropy was not logged, so these observations establish student collapse but not transmission of teacher entropy collapse.

Method	Strict-SR	Mean-Score	Turns	Time GPU·h / s
Qwen3.5-4B				
Frozen				
ReAct	17.8	25.9	39.5	87
Offline				
MemP (offline)	33.2±1.3	43.7±2.0	30.0	6.9 / 74
ACE (offline)	5.8±0.5	21.4±1.1	42.0	9.4 / 122
EvoSkill	20.0±1.6	29.3±1.9	38.0	2.06 / 84
GEPA	23.4±1.2	37.5±2.4	34.0	8.4 / 75
TextGrad	10.4±2.1	30.5±2.5	35.9	13.3 / 79
Trace2Skill	14.2±0.6	21.7±1.2	41.6	7.7 / 92
Online adaptation				
MemP	36.8±3.4	47.7±4.8	28.7	74
ReasoningBank	10.6±0.7	21.6±1.0	42.1	176
ACE	10.6±3.2	23.6±4.3	40.6	107
A-Mem	23.2±3.1	36.4±2.9	33.9	98
MemRL	21.8±1.2	32.3±2.5	37.1	106
Dynamic Cheatsheet	16.2±1.5	28.4±2.1	39.4	133
TT-OPSD	25.6±0.5	37.1±1.7	34.6	100
ASCENT	**41.9**±0.3	**62.4**±1.0	**17.8**	76
Qwen3.5-9B				
Frozen				
ReAct	17.2	27.8	37.5	166
Offline				
MemP (offline)	26.6±1.3	40.6±2.8	31.9	12.3 / 148
ACE (offline)	13.2±0.6	22.4±1.4	41.6	16.4 / 229
EvoSkill	20.2±2.1	33.4±1.4	38.2	2.33 / 169
GEPA	35.4±3.2	48.6±1.6	30.4	10.1 / 135
TextGrad	18.4±2.5	34.7±2.8	34.5	14.7 / 153
Trace2Skill	10.0±4.2	21.0±3.6	40.4	10.2 / 179
Online adaptation				
MemP	22.4±2.5	38.9±4.1	32.9	152
ReasoningBank	12.6±2.1	21.8±1.5	41.5	226
ACE	16.0±1.4	32.0±2.6	36.1	144
A-Mem	15.4±0.6	32.5±4.3	35.6	176
MemRL	14.0±4.2	27.4±1.7	38.9	166
Dynamic Cheatsheet	13.4±2.9	27.4±3.0	39.1	211
TT-OPSD	24.2±2.4	35.9±2.3	34.8	108
ASCENT	**41.5**±1.8	**56.8**±1.7	**26.7**	164

Table 7: **WebShop results.** Strict-SR (%) denotes `task_score`$= 1$, and Mean-Score (%) averages `task_score` over each 500-task stream. Offline runtime is training GPU-hours / evaluation seconds per episode; other entries are seconds per episode. Bold indicates the highest Strict-SR and Mean-Score and the fewest turns within each backbone.

The frozen-policy KL control limits some drift but remains well below the frozen privileged teacher in success. A reference penalty keeps the student closer to π_0; the privileged teacher also provides episode-conditioned soft targets.

Collapsed behavior does not require a small adapter. At an endpoint, the collapsed self-imitation checkpoint has a lower adapter stable rank and a next-token gradient about $25\times$ larger than the frozen-teacher checkpoint. The zero-lag control also ends with a larger adapter norm than the frozen-target stability control, while action admissibility falls. Student entropy approaches zero, repeated actions become more common, and executable actions become rare. These are behavioral

Target / control	Schedule	Stream SR (%)	Early→late SR (%)	First low-success task
Frozen π_0 (ASCENT; Table 1)	fixed	69.5 ± 5.6	–	–
EMA, matched z_i construction	$\beta = 0.99$	51.7 ± 4.4	$44.8 \rightarrow 60.0$	none
EMA ladder	$\beta = 0.999$	42.1	$45.7 \rightarrow 45.7$	none
EMA ladder	$\beta = 0.99$	47.9	$45.7 \rightarrow 45.7$	none
EMA ladder	$\beta = 0.90$	20.0	$34.3 \rightarrow 17.1$	38
Snapshot teacher	$L = 10$	20.7	$45.7 \rightarrow 2.9$	56
Snapshot teacher	$L = 5$	10.7	$37.1 \rightarrow 0.0$	49
Online RFT + KL	–	16.0 ± 5.9	$33.3 \rightarrow 8.6$	38 [32, 54]
Online SFT	–	1.4 ± 0.6	$5.7 \rightarrow 0.0$	20

Table 9: **Success under alternative teacher targets.** The final column gives the first task where 20-task success falls to 15% or below (none if not observed). Early and late refer to tasks 1–35 and 106–140, respectively.

and endpoint associations: stable rank does not establish the cause of collapse, and adapter norm or probe gradient does not measure the realized update or identify a LoRA capacity limit.

ASCENT instead changes the evidence supplied to a fixed teacher. Its targets vary with z_i and $c_{i,t,j}$, while Q_0 remains the same across tasks. The tracking diagnostics show that slowly moving teachers can remain stable, but do not show a benefit from adding teacher-weight evolution here.

C.2 Turn Efficiency under Success Composition

In these ALFWorld runs, failed episodes use all $T_{\max} = 50$ turns. The all-episode mean therefore decomposes as

$$\mathbb{E}[T] = \Pr(W)\,\mathbb{E}[T \mid W] + \big(1 - \Pr(W)\big)T_{\max}, \tag{24}$$

where W denotes episode success. A policy with higher success therefore has fewer mean turns even if its successful trajectories are no shorter. To examine turn use apart from this success-composition effect, we pair each of the three ASCENT runs per scale in Table 1 with a greedy frozen ReAct run on the same task order.

Among successful episodes, frozen ReAct and ASCENT average 17.9 and 15.8 turns at 4B, and 18.4 and 12.6 at 9B. These means still involve different sets of solved tasks. Table 10 therefore compares turn counts only where both policies succeed on the same task.

Scale	Both won	ASCENT only	Frozen only	Mean ΔT	Median ΔT	ASCENT uses fewer turns (%)
4B	61.7±1.2	35.7±6.6	3.3±1.2	−4.8±0.9	−2.3±0.5	63.8±1.8
9B	74.3±0.5	34.0±2.2	2.7±0.5	−7.0±0.3	−3.5±0.4	67.3±4.7

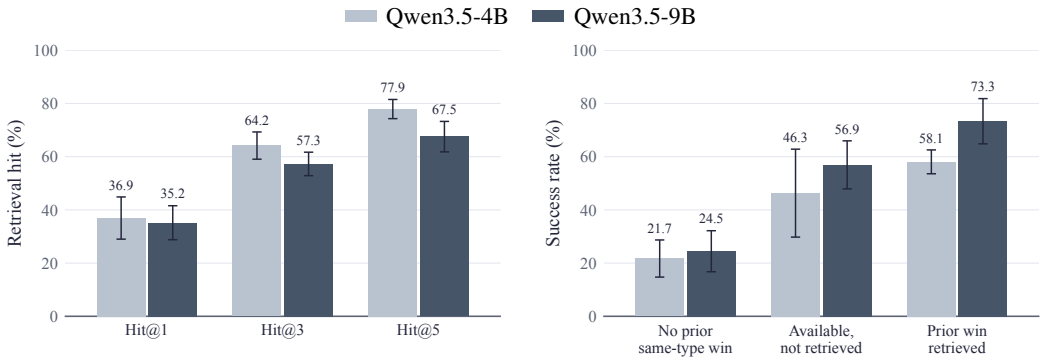
Table 10: **Turn efficiency on shared successes.** $\Delta T = T_{\text{ASCENT}} - T_{\text{Frozen}}$ on tasks solved by both policies; negative values favor ASCENT. The first three columns count tasks in the 140-task stream, and the last column is the percentage of shared successes on which ASCENT uses fewer turns than frozen ReAct. Entries are mean±SD of per-run statistics over the three paired runs per scale.

On these shared successes, ASCENT uses fewer turns on most tasks at both scales, with a lower mean and median turn count (Table 10). The paired result supports an interaction-efficiency gain beyond the change in the success/failure mix. It identifies neither the individual turns avoided nor the cause of each shorter path.

C.3 Retrieval and Execution for In-Context Adaptation

We inspect three 140-task ALFWorld seen streams per backbone for a retrieval-based contextual-memory baseline. Before task i, let $E_i = 1$ when memory contains an item from an earlier

environment-verified success of the same task type. Let $R_i^{(k)} = 1$ when at least one such item appears in the top-k retrieved memories. Fig. 7a reports the retrieval hit rate $\Pr(R_i^{(k)} = 1 \mid E_i = 1)$. Panel (b) separates episode success by whether a prior same-type win is absent ($E_i = 0$), available but not retrieved ($E_i = 1, R_i^{(5)} = 0$), or retrieved ($R_i^{(5)} = 1$). Logged and reconstructed retrieval counts agree for all 840 episodes.



(a) Retrieval hit when a prior same-type win exists

(b) Success conditioned on memory state

Figure 7: **Retrieval and execution behavior.** (a) Retrieval of an available prior same-type winning memory. (b) Success by memory availability and retrieval. Bars show means with SD error bars.

An available prior win is frequently missed in the top five results, and retrieval is associated with higher success but does not ensure it (Fig. 7). Same task type is only a proxy for relevance: a retrieved item may not provide the instruction needed for the current task. Thus a failure after retrieval alone cannot establish that the agent failed to execute useful guidance. The following episode examines concrete traces behind these aggregate patterns.

Case study on long-horizon task completion. As shown in Fig. 8, the frozen policy and five in-context adaptation variants exhaust the turn budget on the task "put a cool egg in microwave," whereas ASCENT completes it in 20 turns. The in-context variants fail either by retrieving irrelevant or contradictory experience or by not following relevant guidance. ASCENT instead internalizes earlier same-family experience in its weights and transfers it across objects and destinations. Fig. 7 shows that same-type prior experience can be available and retrieved yet not reliably lead to task completion.

A scored episode and prior experience. The task in Fig. 8 is ALFWorld seen-stream episode `game_idx=135`: "put a cool egg in microwave." Each method uses Qwen3.5-4B and has one scored rollout with a shared 50-turn limit. Online methods follow the same task order; offline in-context adaptation variants use fixed preconstructed banks. We distinguish the scored outcomes from the later trace diagnostics shown in the figure.

Prior in-context experience. Online MemP acquires a cool-then-place workflow at episode 138, so that workflow was not available for episode 135. The contextual-memory baseline analyzed above has 408 memories in its post-episode bank. Restricting the bank to entries with `src_idx<135` leaves 405 memories, and retrieval from this pre-task bank returns the contradictory top item shown in Fig. 8.

Prior parametric updates. Before episode 135, native ASCENT updates were selected from two tasks in the `pick_cool_then_place_in_recep` family: episode 98 (cool a tomato and place it in a garbage can) and episode 100 (cool a plate and place it in a cabinet). The adapter entering episode 135 may reflect these earlier updates; the task's own outcome cannot affect its scored rollout.

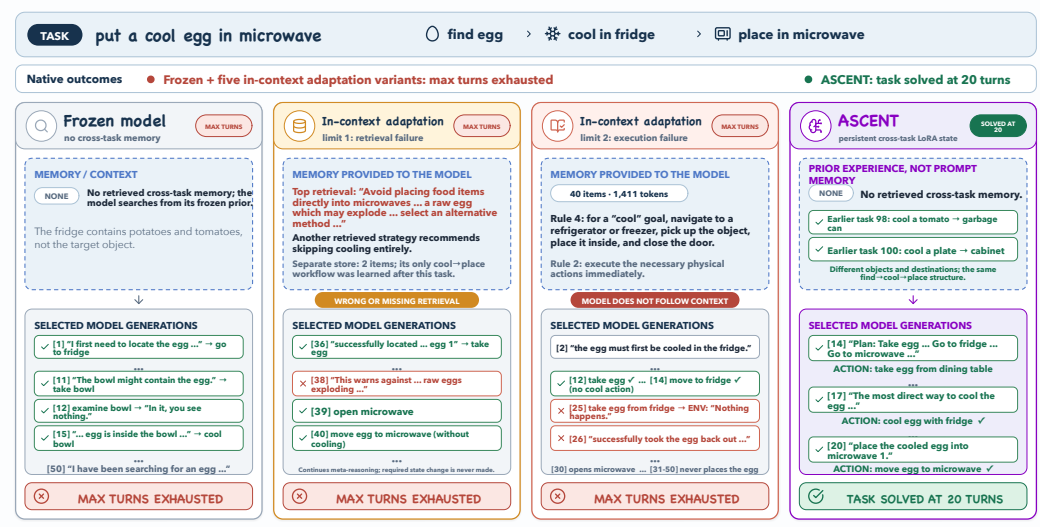


Figure 8: Long-horizon task-execution case study.