NOT ALL EXPERIENCE TRANSFERS: TOWARD BOUNDARY-AWARE SELF-EVOLUTION IN TASK STREAMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model agents can self-evolve by converting accumulated experience into persistent improvements. However, existing methods are predominantly developed and validated in isolated, homogeneous settings, whereas realistic deployment presents heterogeneous task streams in which tasks from interleaved domains arrive without explicit domain labels. This leaves a heterogeneity gap between where self-evolution is studied and where it is deployed. In this work, we ask how self-evolution should operate under such realistic task streams. We propose BORDER (**B**oundary-**OR**iented **D**istillation and **E**volution **R**egularization), a self-evolving framework that combines boundary-aware hierarchical experience curation with selective internalization into policy parameters. Specifically, BORDER extracts local skills from successful interactions and curates a global experience pool by separating reusable decision rules from instance-specific details. Local and global experience guide new interactions, whose outcomes drive reinforcement learning (RL). Meanwhile, on-policy self-distillation (OPSD) is restricted to verified successful trajectories whose local skill contexts increase their likelihood. These mechanisms enable experience to accumulate in both the external pool and policy parameters, supporting selective reuse while mitigating interference. Extensive experiments on three diverse benchmarks with Qwen3-4B-Instruct and Qwen2.5-7B-Instruct under two streaming scenarios show that BORDER achieves the highest average cumulative accuracy in all four backbone–scenario settings, outperforming the strongest parameter-based baseline in each setting by 2.68 percentage points on average.

1 INTRODUCTION

Large language model (LLM) agents are increasingly deployed as long-lived systems that continue to interact with users and environments after release, motivating a growing body of work on self-evolution, where agents improve from experience accumulated during deployment (Fang et al., 2025; Gao et al., 2026; Lin et al., 2026a; Wang et al., 2026). Existing methods are commonly categorized by the components they evolve (Gao et al., 2026), including model parameters (Zhou et al., 2025; Wang et al., 2025; Wu et al., 2026), context (Chhikara et al., 2025; Zhang et al., 2026b), tools (Wang et al., 2024; Qu et al., 2025; Zhang et al., 2026c), and agent architectures (Zhang et al., 2025b; Hu et al., 2025b). However, self-evolution has predominantly been studied and evaluated in isolated, homogeneous settings (Xu et al., 2025; Zhang et al., 2026d). In real-world deployment, agents instead evolve over heterogeneous task streams, where prior experience may be transferable, irrelevant, or even harmful to subsequent tasks. This raises a central question: *How can LLM agents continually evolve over heterogeneous task streams?*

Realistic task streams place multiple demands on a self-evolving agent, as highlighted by recent streaming evaluations (Yan et al., 2026; Guan et al., 2026b). We view these as three complementary requirements for continual self-evolution. ❶ *Experience Accumulation*: the agent should continually retain and refine incoming experience to support sustained performance improvements throughout the stream, rather than relying on benchmark-wise training to convergence. ❷ *Experience Transfer*: the agent should selectively reuse prior experience when its knowledge, reasoning patterns, or strategies are relevant to subsequent tasks, distinguishing reusable guidance from details tied to its

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.

originating interaction. ❸ *Experience Interference Suppression*: the agent should limit the negative effects of inappropriate experience reuse or internalization, so that irrelevant guidance or uninformative hindsight does not undermine current decisions or capabilities needed by other tasks. Together, these requirements capture the central challenge of learning from task streams: continually accumulating experience while regulating when, how, and to what extent it should shape future behavior.

Streaming self-evolution therefore requires not only persistent experience accumulation, but equally careful regulation of the scope over which that experience should influence future tasks. Motivated by this insight, we propose **BORDER** (**B**oundary-**OR**iented **D**istillation and **E**volution **R**egularization), a hierarchical experience learning framework built on reinforcement learning (RL) and on-policy self-distillation (OPSD) (Zhao et al., 2026). Rather than treating all hindsight skills extracted from interaction trajectories as uniformly transferable, BORDER uses the evolving policy to assess their transferability and abstract reusable guidance into a global experience pool for subsequent cross-task learning, without relying on domain labels. Freshly extracted local skills are filtered through success conditioning and evidence gating, providing OPSD supervision only for successful trajectories whose hindsight contexts offer additional support for the observed behavior. Since source-task success alone does not establish the applicability of global skills to new tasks, retrieved global skills guide resampling, with environment feedback evaluating the resulting behavior and guiding RL updates. These complementary learning signals are jointly optimized to consolidate accumulated experience into policy parameters. Together, these mechanisms enable continual accumulation and selective transfer while limiting interference, preserving experience in both the global pool and the evolving policy.

Extensive experiments across two backbone models, two streaming scenarios, and three benchmarks spanning diverse domains demonstrate consistent performance gains, indicating that BORDER can robustly accumulate experience from task streams and consolidate it into policy parameters. Fine-grained ablation studies further demonstrate the benefits of selectively regulating how experience is reused across tasks and incorporated into policy updates. **Our contributions are summarized as follows:**

- We study continual self-evolution under heterogeneous task streams, framing the challenge around three complementary requirements: experience accumulation, selective transfer, and interference suppression.
- We propose **BORDER**, a boundary-aware framework that organizes experience into local and global skills for contextual reuse and selective internalization into policy parameters. It couples experience-guided reinforcement learning with filtered on-policy self-distillation to regulate the influence of experience without requiring task identities.
- Extensive experiments on three diverse benchmarks with two backbone models under both Isolated and Interleaved task streams demonstrate improved average performance over strong baselines.

2 Related Work

2.1 Learning from Streaming Data

Continual learning and test-time adaptation provide complementary perspectives on learning under changing data distributions. Continual learning studies how models acquire knowledge from sequential tasks while retaining previously learned capabilities (Lopez-Paz & Ranzato, 2017). Test-time training updates model parameters using self-supervision on unlabeled test inputs to improve robustness to distribution shifts (Sun et al., 2020), while related source-free domain adaptation methods adapt a pretrained model to an unlabeled target domain without accessing the original training data (Liang et al., 2020). For LLM agents, learning across tasks additionally involves retaining interaction experience and carrying its influence into subsequent decisions. AgentStream formalizes this evaluation through Isolated, Sequential, and Interleaved task streams, systematically comparing self-evolving methods across backbone models and streaming scenarios (Yan et al., 2026). Its findings show that the benefits of self-evolution depend on both the underlying model and the streaming scenario. Related analyses distinguish the ability to produce useful harness updates from the ability to benefit from them during task execution (Lin et al., 2026b). Within this broader setting, our work

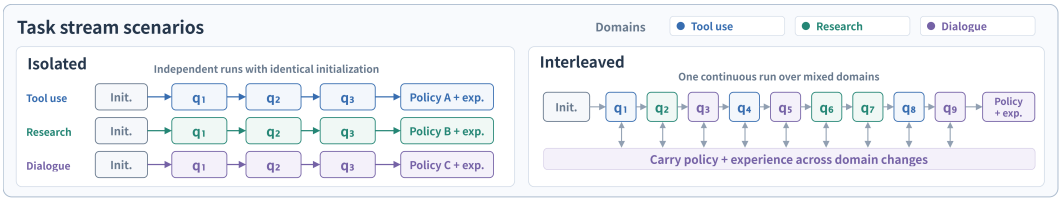


Figure 1: Task stream scenarios for continual self-evolution. Isolated (left): each domain is evaluated in an independent run, starting from the same initial policy and experience state and retaining updates within that run. Interleaved (right): tasks from multiple domains are shuffled into a single stream, with a shared evolving policy and accumulated experience carried across domain changes. Colors indicate task domains.

studies how parameter-based self-evolution can accumulate experience, selectively transfer it across tasks, and limit interference during continual policy updates.

2.2 Self-Evolving LLM Agents

Self-evolving LLM agents improve their policies or surrounding systems using interaction experience and feedback. Recent surveys organize these approaches by the components being evolved, including model parameters, memory, tools, and agent architectures (Gao et al., 2026; Fang et al., 2025). A prominent line of work retains experience outside model parameters. ExpeL extracts natural-language insights from past trajectories, while ReasoningBank derives reusable reasoning strategies from self-judged successful and failed experiences (Zhao et al., 2024; Ouyang et al., 2026). ACE maintains evolving contextual playbooks, and A-Mem organizes and updates interconnected external memories (Zhang et al., 2026d; Xu et al., 2025). MemRL uses environment feedback to improve memory selection, while AutoSkill maintains reusable skills derived from interaction traces; both keep the underlying model weights fixed (Zhang et al., 2026e; Yang et al., 2026b). Other approaches evolve executable skills or agent structures: Voyager and SkillWeaver construct reusable skill libraries through environment interaction (Wang et al., 2024; Zheng et al., 2025), ADAS and AFlow automate agent design and workflow optimization (Hu et al., 2025a; Zhang et al., 2025b), and Darwin Gödel Machine iteratively modifies and evaluates agent code (Zhang et al., 2025a). Complementary approaches internalize experience through parameter updates, with RAGEN and AgentEvolver studying reinforcement learning in interactive environments (Wang et al., 2025; Zhai et al., 2025). More closely related to our work, SEED jointly optimizes outcome-based reinforcement learning and on-policy distillation from self-generated hindsight skills (Wu et al., 2026), while OPD-Evolver combines rapid evolution of hierarchical memory with policy learning through on-policy self-distillation (Zhang et al., 2026a).

3 Method

We first introduce the task stream setup (Sec. 3.1), covering the Isolated and Interleaved scenarios illustrated in Fig. 1. We then present the three complementary modules of BORDER, illustrated in Fig. 2. **Experience Accumulation** combines outcome-based reinforcement learning with local hindsight distillation to continually consolidate interaction experience into policy parameters (Sec. 3.2). **Experience Transfer** constructs global guidance by abstracting and screening local skills, then uses this guidance for cross-task exploration whose outcomes provide the RL learning signal (Sec. 3.3). **Experience Interference Suppression** regulates local distillation through success conditioning and evidence gating, limiting unreliable imitation pressure during continual policy updates (Sec. 3.4).

3.1 Problem Setup: Learning from Heterogeneous Task Streams

We consider an agent processing a task stream $\mathcal{Q} = (q_1, \ldots, q_N)$, where tasks arrive sequentially and future tasks cannot be accessed before their arrival (Yan et al., 2026; Guan et al., 2026a). Starting from an initial policy π_{θ_0}, the agent uses interaction trajectories and environment feedback to update its policy, retained experience, or both. These updates are carried forward to subsequent tasks, allowing experience to accumulate throughout the stream. To examine how task heterogeneity affects self-evolution, we treat each benchmark as a task domain and consider two streaming

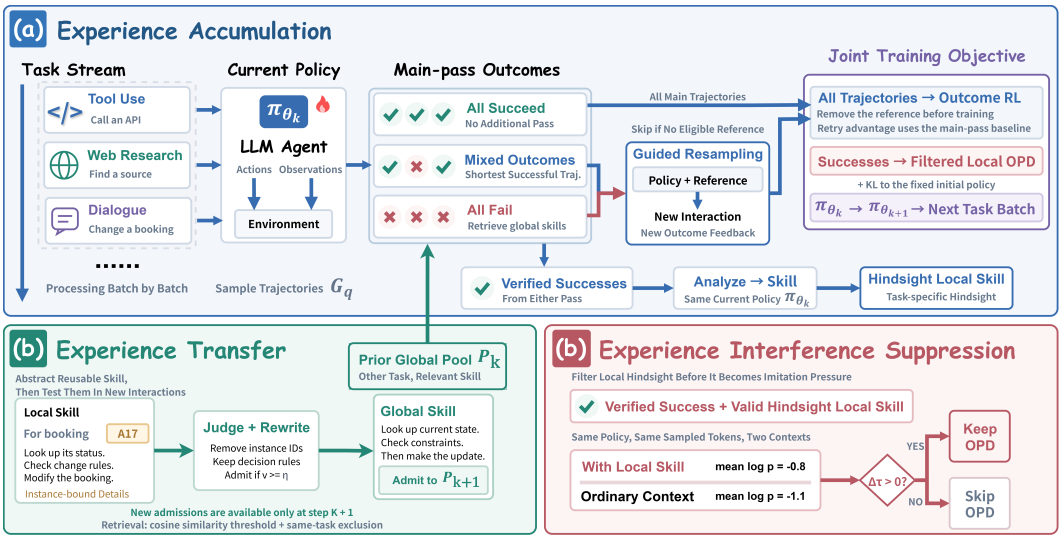


Figure 2: Overview of BORDER. (a) Experience Accumulation combines outcome-based reinforcement learning with local hindsight distillation to consolidate interaction experience into policy parameters. (b) Experience Transfer abstracts and screens local skills into global guidance for cross-task exploration, with new environment feedback providing the learning signal. (c) Experience Interference Suppression regulates local distillation through success conditioning and evidence gating, limiting unreliable imitation during continual policy updates.

scenarios, illustrated in Fig. 1. These scenarios distinguish within-domain evolution in independent runs from evolution over mixed domains with a shared policy and experience state.

Isolated. The agent processes a stream from a single domain, with each domain evaluated in a separate run starting from the same initial policy and experience state. Experience is retained only within that run, allowing us to assess within-domain self-evolution.

Interleaved. The agent processes a shuffled stream of tasks from multiple domains, carrying a shared evolving policy and accumulated experience across domain transitions. Experience acquired in one domain can therefore benefit or interfere with learning and performance in another.

In both scenarios, domain labels are provided to the agent during training. These labels identify task domains, while the applicability of individual experiences to subsequent tasks still needs to be assessed. Our focus is on continually accumulating experience while regulating how it is reused and incorporated into the shared policy.

3.2 Experience Accumulation via Hierarchical Learning

Continual improvement requires agents to learn from specific interactions while reusing relevant experience across tasks. BORDER addresses these two scopes through hierarchical learning: local hindsight skills support filtered distillation from their source trajectories, while abstracted global skills guide new interactions whose outcomes provide reinforcement learning signals. This distinction aligns experience internalization with the evidence available at each level, combining outcome-based reinforcement learning with local hindsight distillation (Wu et al., 2026).

Interaction and Experience Extraction. At evolution step k, the current policy π_{θ_k} processes the next batch of tasks from the stream and collects a main-pass trajectory group $\mathcal{G}_q$ for each task q. Experience-guided interactions can supplement these initial attempts with further exploration (Sec. 3.3). The main-pass and additional trajectories, together with their environment feedback, form the training batch $\mathcal{D}_k$.

For each trajectory $\tau \in \mathcal{D}_k$ verified as successful by feedback $\mathcal{F}_\tau$, the policy extracts a local hindsight skill:

$$z_\tau^{\text{local}} = \text{Analyze}_{\theta_k}(\tau, \mathcal{F}_\tau). \tag{1}$$

Local skills summarize decision rules and procedures associated with their source interactions. Selected skills are further abstracted into global guidance for reuse across tasks. This hierarchy preserves task-specific guidance for local learning while making potentially reusable knowledge available to future tasks. Its applicability to new interactions must still be assessed.

Hierarchical Experience Internalization. We use environment outcomes from the collected trajectories to optimize the policy through reinforcement learning. For experience-guided interactions, the policy receives an auxiliary skill as guidance when generating the trajectory. During RL optimization, this added guidance is removed, while the task, generated actions, and environment observations are retained. The resulting behavior is therefore learned from its environment reward under the ordinary interaction context, encouraging the policy to internalize useful behavior without relying on the guiding skill as an additional input.

Local skills complement outcome rewards with guidance about how successful behavior was achieved. We internalize this guidance through OPSD. For a sampled response token $y_{\tau,j}$ and its ordinary interaction history $h_{\tau,j}$, including preceding response tokens, the skill-conditioned policy serves as teacher and the ordinary-context policy as student:

$$\begin{aligned} u_{\tau,j}(\theta) &= \log \pi_\theta\big(y_{\tau,j} \mid H(h_{\tau,j}, z_\tau^{\mathrm{local}})\big), \\ s_{\tau,j}(\theta) &= \log \pi_\theta\big(y_{\tau,j} \mid h_{\tau,j}\big), \end{aligned} \tag{2}$$

where H augments the ordinary context with the local skill. Both branches score the same sampled tokens and share policy parameters; teacher scores are detached when supervising the student branch. Since a successful outcome does not establish that its hindsight explanation is informative, this supervision is restricted to successful trajectories whose skill contexts provide additional support for the observed behavior (Sec. 3.4).

Joint Training Objective. We combine the two learning paths through a clipped GRPO objective $\mathcal{L}_{\mathrm{PG}}$, filtered local distillation, and KL regularization. The overall objective is

$$\mathcal{L}_{\mathrm{evo}}(\theta) = \mathcal{L}_{\mathrm{PG}}(\theta) + \lambda_{\mathrm{OPD}}\mathcal{L}_{\mathrm{OPD}}^{\mathrm{filtered}}(\theta) + \lambda_{\mathrm{KL}}\mathcal{L}_{\mathrm{KL}}(\theta), \tag{3}$$

where $\mathcal{L}_{\mathrm{OPD}}^{\mathrm{filtered}}$ is defined in Sec. 3.4, $\mathcal{L}_{\mathrm{KL}}$ regularizes the policy against its fixed initialization, and λ_{OPD} and λ_{KL} weight the corresponding terms. The updated policy $\pi_{\theta_{k+1}}$ and global pool $\mathcal{P}_{k+1}$ are carried forward to the next evolution step, preserving accumulated experience in both parameters and reusable guidance.

The following sections detail how global experience is curated for cross-task learning (Sec. 3.3) and how local hindsight supervision is filtered (Sec. 3.4).

3.3 Experience Transfer via Boundary-Aware Curation

The global level extends learning beyond the source interactions of local skills. Since successful procedures may remain entangled with instance-specific details, BORDER abstracts and screens global guidance, then grounds its contribution to policy learning in new environment outcomes.

Boundary-Aware Experience Curation. We use *boundary* to describe the scope over which experience may remain applicable. Transferring instance-specific details may encourage actions that were appropriate only in the source interaction. To distinguish reusable procedures from such content, the evolving policy assesses each candidate local skill and proposes an abstracted global version:

$$(v_\tau, z_\tau^{\mathrm{global}}) = \mathrm{JudgeRewrite}_{\theta_k}(z_\tau^{\mathrm{local}}), \tag{4}$$

where $v_\tau \in [0, 1]$ estimates the candidate's reuse potential. The global version preserves decision rules and action order while removing source-specific entities and values. Screening determines which candidates enter the pool, while abstraction determines what information they carry forward. Candidates with $v_\tau \geq \eta$ are retained, where η is the admission threshold; their original local versions remain available for hindsight distillation. The admission threshold limits the accumulation of weakly reusable guidance as experience grows throughout the stream. The global pool $\mathcal{P}_k$ contains skills retained before evolution step k. These abstractions provide candidates for transfer whose usefulness must still be assessed through their application to new tasks.

Outcome-Grounded Cross-Task Transfer. When the main-pass group $\mathcal{G}_q$ contains no successful trajectory, we retrieve global guidance from $\mathcal{P}_k$ to support further exploration. Since its applicability

to the current task remains uncertain, the retrieved guidance, when available, is used to generate additional rollouts whose environment rewards provide the RL learning signal (Sec. 3.2). Global experience is thus incorporated into policy parameters through the outcomes of new interactions. Successful reuse also supports continued retention in the pool, while newly successful trajectories provide local skills for subsequent distillation and abstraction. Pool maintenance details are provided in Appendix A.4.

3.4 Experience Interference Suppression via Selective Hindsight Distillation

At the local level, hindsight distillation should reinforce behavior supported by both successful outcomes and informative skill contexts. Otherwise, it may imitate unsuccessful behavior or add supervision without useful guidance. Success conditioning and trajectory-level evidence gating regulate this local learning path in Eq. 3, while outcome-based policy learning remains active.

Success-Conditioned Distillation. For the hindsight OPD surrogate, detaching the teacher score and using a positive detached token weight $g_{\tau,j}(\theta)$ gives the token-level gradient $-g_{\tau,j}(\theta)\nabla_\theta s_{\tau,j}(\theta)$, where the teacher and student scores are defined in Eq. 2. This signal increases the probability of sampled tokens even when the hindsight context lowers their likelihood. On failed trajectories, such imitation pressure can oppose reward-driven learning. We therefore extract hindsight skills and apply distillation only to verified successful trajectories. Let $\mathcal{V}_k \subseteq \mathcal{D}_k$ contain successful trajectories for which a local skill is obtained. Failed trajectories remain available for outcome-based learning. Success conditioning determines which behavior is eligible for imitation; evidence gating then assesses the supervision provided by its local skill.

Trajectory-Level Evidence Gating. Even a successful trajectory can yield an uninformative hindsight explanation. We assess the skill's contextual support by comparing the likelihood of the same successful trajectory with and without its local skill. Let $\mathcal{T}_\tau$ index the valid response tokens in τ. Using the policy weights before optimization, we compute

$$\Delta_\tau = \frac{1}{|\mathcal{T}_\tau|} \sum_{j \in \mathcal{T}_\tau} \left[u_{\tau,j}(\theta_k) - s_{\tau,j}(\theta_k) \right], \qquad m_\tau = \mathbf{1}[\Delta_\tau > 0]. \tag{5}$$

Scoring the same trajectory at the same policy weights isolates the likelihood change associated with adding the skill context. This gain provides a measure of contextual support for selecting hindsight supervision. We retain the trajectory's distillation contribution only when this gain is positive. The mask is held fixed throughout the update and applies to the trajectory as a whole. Trajectory-level gating prevents token weighting from retaining imitation pressure when the skill provides no overall likelihood benefit. The resulting filtered objective is

$$\mathcal{L}_{\mathrm{OPD}}^{\mathrm{filtered}}(\theta) = \frac{1}{N_k} \sum_{\tau \in \mathcal{V}_k} m_\tau \sum_{j \in \mathcal{T}_\tau} g_{\tau,j}(\theta) \left[\mathrm{sg}[u_{\tau,j}(\theta)] - s_{\tau,j}(\theta) \right], \tag{6}$$

where sg denotes stop-gradient. Following SEED, the detached confidence weight is $g_{\tau,j}(\theta) = \mathrm{sg}[\sigma(\beta(u_{\tau,j}(\theta) - s_{\tau,j}(\theta)))]$, with sigmoid σ and sharpness $\beta > 0$. The mask selects which trajectories contribute supervision, while the confidence weight adjusts token contributions within retained trajectories. The denominator N_k counts all valid response tokens in $\mathcal{D}_k$, avoiding amplification of the remaining supervision when filtering reduces its coverage.

Together, these filters determine which local experience contributes to the joint learning objective in Sec. 3.2.

4 Experiment

4.1 Experimental Setup

Benchmarks and Evaluation Metrics. We evaluate BORDER on three agentic benchmarks: BFCL (Patil et al., 2025) for multi-step function calling, BrowseComp+ (Chen et al., 2026) for deep web research over a controlled corpus, and Tau2-Bench (Barrès et al., 2025) for conversational agents in dual-control environments. Following Yan et al. (2026), we treat each benchmark as a domain. We construct *Isolated* streams within individual domains and an *Interleaved* stream by mixing tasks across domains without providing explicit domain labels. Each arriving task is evaluated before its experience is incorporated into the agent or used for policy optimization. We report

cumulative accuracy (Cum. Acc.), which averages first-attempt success over the incoming stream, and holdout accuracy (Hold. Acc.), which measures performance on a fixed unseen set excluded from experience accumulation. Both metrics are reported as success rates, and Avg. denotes the corresponding average across the three benchmarks. More details are provided in Appendix A.

Base LLMs and Baseline Methods. We use Qwen3-4B-Instruct (Yang et al., 2025) and Qwen2.5-7B-Instruct (Qwen et al., 2025) as backbone models. Vanilla provides a non-evolving reference without streaming adaptation. We compare BORDER with four parameter-based baselines: GRPO (Shao et al., 2024) for outcome-based RL, OPSD (Zhao et al., 2026) for self-distillation with privileged teacher contexts, RLSD (Yang et al., 2026a) for combining outcome-based RL with self-distillation, and SEED (Wu et al., 2026) for experience-driven on-policy self-distillation. We additionally compare with two harness-based methods on Qwen3-4B-Instruct: ACE (Zhang et al., 2026d), which incrementally updates a contextual strategy playbook through reflection and curation, and ReasoningBank (Ouyang et al., 2026), which extracts reusable reasoning strategies from successful and failed experiences and retrieves relevant memories for subsequent tasks. Both adapt through external context or memory while keeping the backbone parameters fixed. Further baseline details are provided in Appendix A.3.

Implementation Details. Our implementation consists of two stages. *SFT initialization.* For each backbone, we sample 300 training tasks and collect $K = 8$ rollout trajectories per task, yielding 2,400 trajectories. GLM-5.2 (Z.ai, 2026) serves as an external trajectory analyzer to extract hindsight skills. The resulting trajectory–skill pairs are used to fine-tune the backbone for three epochs. *Streaming evolution.* The SFT checkpoint initializes both the policy and the trajectory analyzer used during RL. For the main experiments, we sample 64 tasks from each benchmark. Each domain-specific subset forms an independent *Isolated* stream, while the same 192 tasks are randomly shuffled to construct the *Interleaved* stream. Main results are averaged over three random seeds. For component analysis, we use 32 tasks per benchmark and evaluate only the Interleaved scenario. Additional details on data construction, skill annotation, optimization, and hyperparameters are provided in Appendix A.4.

4.2 Performance of Evolution in Task Streams

Table 1 compares BORDER with existing parameter-based self-evolution methods on three benchmarks. Results are grouped by backbone, with each method evaluated under Isolated and Interleaved task streams and averaged over three random seeds. Cum. Acc. measures cumulative accuracy throughout the stream, while Hold. Acc. evaluates the evolved policy on held-out tasks. Avg. summarizes the corresponding scores across benchmarks, and Vanilla provides the non-evolving reference. **BORDER ranks first or second in both average cumulative and holdout accuracy across all evaluated backbones and streaming scenarios.** On Qwen3-4B-Instruct, it leads both average metrics in both scenarios; on Qwen2.5-7B-Instruct, it leads both in Isolated and maintains the highest average cumulative accuracy in Interleaved.

The cumulative performance advantage extends to heterogeneous Interleaved streams across both backbones. On Qwen3-4B-Instruct, BORDER exceeds the strongest baseline in average cumulative and holdout accuracy by 3.87 and 2.09 percentage points, respectively. On Qwen2.5-7B-Instruct, it achieves the highest cumulative accuracy on BFCL and BrowseComp+, and the second-highest on Tau2, yielding an average cumulative advantage of 0.80 points over the strongest baseline. The holdout results are less uniform: its average holdout accuracy in this setting is 10.41%, compared with 11.46% for RLSD and SEED. Together, these results support the effectiveness of BORDER in learning from heterogeneous task streams, while showing that its consistent cumulative advantage does not imply uniformly superior held-out performance.

4.3 Component Analysis for Streaming Self-Evolution

Table 2 presents an incremental component analysis using Qwen3-4B-Instruct under Interleaved task streams. Unlike the main experiments, this study uses 32 training tasks per benchmark, forming a stream of 96 tasks in total and using half the main-experiment training data. Results are averaged over three random seeds. Starting from experience accumulation alone (Accum. Only), we first add interference suppression (Supp.) and then experience transfer to obtain BORDER (Full). **Adding interference suppression improves average holdout performance.** Average holdout accuracy increases from 16.67% to 22.92%, with gains of 9.38 percentage points on both BFCL and Tau2.

Table 1: **Comparison of BORDER with existing methods** across two backbones, three benchmarks, and two streaming scenarios. For each backbone and numerical column, scores are ranked separately within each Mode (Isolated or Interleaved), excluding Vanilla. The best and second-best results are marked with darker shading and **boldface**, and lighter shading and underlining, respectively. Ties share the same marking; second-best denotes the second-highest distinct value.

Methods	Mode	BFCL		BrowseComp+		Tau2		Avg	
		Cum. Acc.	Hold. Acc.	Cum. Acc.	Hold. Acc.	Cum. Acc.	Hold. Acc.	Cum. Acc.	Hold. Acc.
Qwen3-4B-Instruct									
Vanilla	–	13.28	14.58	3.13	3.13	11.20	9.38	9.20	9.03
OPSD	Isolated	2.86	0.0	0.26	3.13	9.14	6.30	4.09	3.14
	Interleaved	1.56	0.0	0.0	0.0	6.30	6.30	2.61	2.08
GRPO	Isolated	**23.96**	21.86	8.59	21.86	17.54	25.00	16.70	22.91
	Interleaved	**24.22**	**25.00**	12.76	15.63	21.47	31.25	19.48	23.96
RLSD	Isolated	18.22	9.38	9.38	**25.00**	19.80	31.25	15.80	21.88
	Interleaved	19.53	18.75	13.28	**31.25**	17.80	28.13	16.87	26.04
SEED	Isolated	14.06	21.88	12.50	0.0	**28.13**	28.13	18.23	16.67
	Interleaved	8.85	12.50	**23.44**	0.0	13.02	12.50	15.10	8.33
BORDER (Ours)	Isolated	17.97	**25.00**	**21.18**	9.38	**28.13**	**37.50**	**22.43**	**23.96**
	Interleaved	21.77	**25.00**	19.53	18.75	**28.72**	**40.63**	**23.35**	**28.13**
Qwen2.5-7B-Instruct									
Vanilla	–	0.0	0.0	4.17	3.13	5.99	12.50	3.39	5.21
OPSD	Isolated	0.52	0.0	8.33	12.50	6.51	6.25	5.12	6.25
	Interleaved	0.0	0.0	1.56	0.0	5.22	6.25	2.26	2.08
GRPO	Isolated	5.99	**9.38**	8.81	15.60	7.07	9.38	7.29	11.45
	Interleaved	7.03	6.25	10.16	18.75	**8.85**	6.25	8.68	10.41
RLSD	Isolated	5.21	**9.38**	8.33	12.50	6.01	6.25	6.52	9.38
	Interleaved	5.99	6.25	7.29	21.88	7.83	6.25	7.04	**11.46**
SEED	Isolated	4.42	6.25	0.26	0.0	7.83	9.38	4.17	5.21
	Interleaved	6.51	**9.38**	9.64	**25.00**	3.65	0.0	6.60	**11.46**
BORDER (Ours)	Isolated	**6.51**	**9.38**	**11.46**	**28.13**	**9.38**	**12.50**	**9.12**	**16.67**
	Interleaved	**8.33**	6.25	**11.46**	15.63	8.64	**9.38**	**9.48**	10.41

Table 2: Incremental ablation study (%) of BORDER using Qwen3-4B-Instruct in the Interleaved scenario, averaged over three random seeds. Starting with experience accumulation (Accum. Only), we add experience interference suppression (Supp.) and then experience transfer to obtain BORDER (Full). Within each column, the best results are bold with darker shading, and the second-best results have lighter shading. Tied results receive the same highlighting.

Settings	BFCL		BrowseComp+		Tau2		Avg	
	Cum. Acc.	Hold. Acc.	Cum. Acc.	Hold. Acc.	Cum. Acc.	Hold. Acc.	Cum. Acc.	Hold. Acc.
Accum. Only	10.61	18.75	17.78	0.0	33.78	31.25	21.33	16.67
Accum. + Supp.	11.62	**28.13**	17.22	0.0	**41.89**	**40.63**	24.5	**22.92**
BORDER (Full)	**20.71**	25.00	**26.67**	**9.38**	36.49	18.75	**26.67**	17.71

These results support the usefulness of selectively applying hindsight supervision in this reduced-data setting.

Adding experience transfer further improves cumulative performance, although its effects vary across domains. Relative to Accum. + Supp., the full method increases cumulative accuracy on BFCL and BrowseComp+ by 9.09 and 9.45 percentage points, respectively, while cumulative accuracy on Tau2 decreases by 5.40 points. Compared with Accum. Only, the full method nevertheless achieves higher cumulative accuracy on all three benchmarks. The holdout results reveal a different pattern: adding transfer improves BrowseComp+ holdout accuracy from 0.00% to 9.38%, but reduces Tau2 holdout accuracy from 40.63% to 18.75%. Consequently, the full method attains the highest average cumulative accuracy, whereas Accum. + Supp. achieves the highest average holdout accuracy. This indicates that cross-task experience reuse can improve performance throughout the stream without uniformly improving held-out performance across domains.

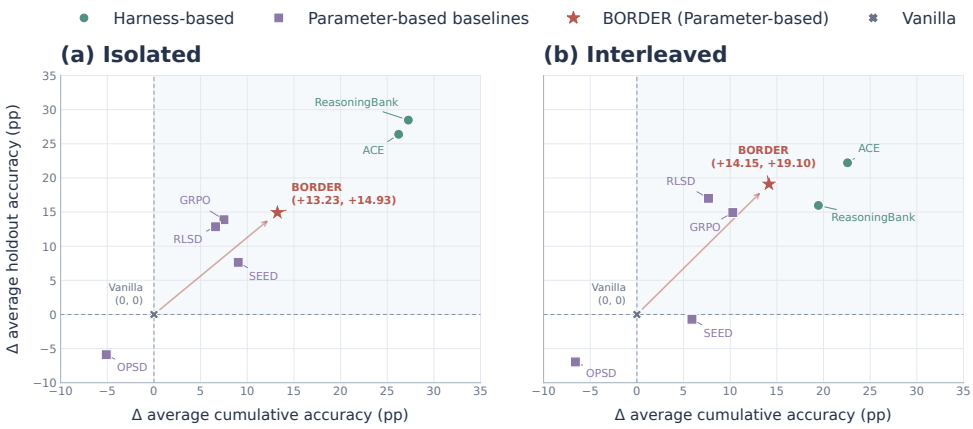


Figure 3: **Comparison with harness-based methods on Qwen3-4B-Instruct.** Each point shows gains over the same Vanilla policy in average cumulative accuracy (horizontal axis) and average holdout accuracy (vertical axis), measured in percentage points, under (a) Isolated and (b) Interleaved task streams. Circles denote harness-based methods, squares denote parameter-based baselines, and the star highlights BORDER, which also uses parameter-based evolution.

4.4 Performance Comparison with Harness-Based Methods

We examine whether consolidating experience into policy parameters can yield gains comparable to harness-based adaptation, and how their relative performance changes when task domains are interleaved. To this end, we compare ACE and ReasoningBank with parameter-based methods using the same Qwen3-4B-Instruct backbone. Figure 3 plots gains over the same Vanilla policy in average cumulative accuracy on the horizontal axis and average holdout accuracy on the vertical axis, measured in percentage points. The two panels correspond to Isolated and Interleaved streams. Vanilla lies at the origin, and points farther toward the upper right indicate larger gains in both performance throughout the stream and performance on held-out tasks.

Harness-based methods outperform all evaluated parameter-based methods on both metrics in Isolated, but both ACE and ReasoningBank obtain lower scores in Interleaved. ACE nevertheless retains the highest scores on both metrics in Interleaved. **BORDER leads the evaluated parameter-based methods on both metrics in both scenarios and narrows the gap to harness-based evolution under Interleaved streams.** Its Interleaved gains over Vanilla reach +14.15 percentage points in cumulative accuracy and +19.10 points in holdout accuracy. The holdout gap to ACE narrows from 11.46 points in Isolated to 3.12 points in Interleaved. Relative to ReasoningBank in Interleaved, BORDER trails by 5.30 points in cumulative accuracy but leads by 3.13 points in holdout accuracy. These results show that the relative advantage of the two evolution approaches depends on both the stream scenario and the evaluation metric.

5 Conclusion

In this work, we studied continual self-evolution under heterogeneous task streams through three complementary requirements: experience accumulation, selective transfer, and interference suppression. We proposed BORDER, a boundary-aware framework that organizes experience into local and global skills for contextual reuse and selective internalization into policy parameters. By coupling experience-guided reinforcement learning with filtered on-policy self-distillation, BORDER regulates the influence of accumulated experience without requiring task identities. Experiments across two backbone models, two streaming scenarios, and three diverse benchmarks demonstrate consistent improvements in average cumulative accuracy over the evaluated parameter-based baselines, supporting the effectiveness of learning from task streams and consolidating experience into the policy. Component analyses further reveal the distinct effects of experience filtering and cross-task reuse. Our findings highlight the importance of controlling when and how accumulated experience shapes future learning in heterogeneous task streams.

AI USE STATEMENT

Generative AI tools supported manuscript editing and language refinement, the discovery of relevant literature, and programming assistance. The authors independently examined and verified all literature identified through these tools. Programming support was limited to code editing, completion, and debugging, and all AI-assisted code underwent manual review, testing, and validation. The research ideas, methodological approach, experimental design, software architecture, implementation logic, and interpretation of findings were developed by the authors. Generative AI was not employed to create synthetic datasets or formulate theoretical models or conceptual frameworks. The authors checked all AI-assisted material and retain full responsibility for the final work, including its text, claims, code, and other artifacts.

ETHICS STATEMENT

BORDER aims to improve the ability of LLM agents to learn continually from task streams through selective experience reuse and policy optimization. Such adaptation may reinforce biases, propagate misleading guidance across tasks, or internalize undesirable behaviors from interaction experience. Retained trajectories and skills may also contain sensitive information, creating privacy risks when stored or reused. Our filtering mechanisms target unreliable hindsight supervision and should not be interpreted as guarantees of safety, fairness, or privacy. We therefore recommend minimizing the retention of sensitive data, controlling access to stored experience, and evaluating evolved policies for harmful behavior and performance degradation before deployment. Applications involving consequential decisions should additionally include continued monitoring and human oversight. Responsible use requires assessing both immediate task performance and the broader effects of accumulated experience on subsequent behavior.

REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our findings, detailed implementation instructions for BORDER are provided in in Section 3 , Appendix A.4. Additionally, we plan to release our source code in the future to further support reproducibility. These measures are intended to facilitate the verification and replication of our results by other researchers in the field.

REFERENCES

Victor Barrès, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-bench: Evaluating conversational agents in a dual-control environment. *CoRR*, abs/2506.07982, 2025. doi: 10.48550/ARXIV.2506.07982. URL https://doi.org/10.48550/arXiv.2506.07982.

Zijian Chen, Xueguang Ma, Shengyao Zhuang, Ping Nie, Kai Zou, Sahel Sharifymoghaddam, Andrew Liu, Joshua Green, Kshama Patel, Ruoxi Meng, Mingyi Su, Yanxi Li, Haoran Hong, Xinyu Shi, Xuye Liu, Hosna Oyarhoseini, Nandan Thakur, Crystina Zhang, Luyu Gao, Wenhu Chen, and Jimmy Lin. BrowseComp-plus: A fair and disentangled evaluation benchmark for deep search agents. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 22349–22370, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1023. URL https://aclanthology.org/2026.acl-long.1023/.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready AI agents with scalable long-term memory. In Inês Lynce, Nello Murano, Mauro Vallati, Serena Villata, Federico Chesani, Michela Milano, Andrea Omicini, and Mehdi Dastani (eds.), *ECAI 2025 - 28th European Conference on Artificial Intelligence, 25-30 October 2025, Bologna, Italy - Including 14th Conference on Prestigious Applications of Intelligent Systems (PAIS 2025)*, volume 413 of *Frontiers in Artificial Intelligence and Applications*, pp. 2993–3000. IOS Press, 2025. doi: 10.3233/FAIA251160. URL https://doi.org/10.3233/FAIA251160.

Jinyuan Fang, Yanwen Peng, Xi Zhang, Yingxu Wang, Xinhao Yi, Guibin Zhang, Yi Xu, Bin Wu, Siwei Liu, Zihao Li, Zhaochun Ren, Nikos Aletras, Xi Wang, Han Zhou, and Zaiqiao Meng. A

comprehensive survey of self-evolving AI agents: A new paradigm bridging foundation models and lifelong agentic systems. *CoRR*, abs/2508.07407, 2025. doi: 10.48550/ARXIV.2508.07407. URL https://doi.org/10.48550/arXiv.2508.07407.

Huan-ang Gao, Jiayi Geng, Wenyue Hua, Mengkang Hu, Xinzhe Juan, Hongzhang Liu, Shilong Liu, Jiahao Qiu, Xuan Qi, Qihan Ren, Yiran Wu, Hongru Wang, Han Xiao, Yuhang Zhou, Shaokun Zhang, Jiayi Zhang, Jinyu Xiang, Yixiong Fang, Qiwen Zhao, Dongrui Liu, Cheng Qian, Zhenhailong Wang, Minda Hu, Huazheng Wang, Qingyun Wu, Heng Ji, and Mengdi Wang. A survey of self-evolving agents: What, when, how, and where to evolve on the path to artificial super intelligence. *Trans. Mach. Learn. Res.*, 2026, 2026. URL https://openreview.net/forum?id=CTr3bovS5F.

Tianyi Guan, Yiding Wang, Haotong Yang, Siyuan Cao, Shirui Liu, Yi Hu, Jiaqi Li, and Muhan Zhang. Continualskillbench: Can LLM agents truly evolve their capabilities? *CoRR*, abs/2608.03874, 2026a. doi: 10.48550/ARXIV.2608.03874. URL https://doi.org/10.48550/arXiv.2608.03874.

Tianyi Guan, Yiding Wang, Haotong Yang, Siyuan Cao, Shirui Liu, Yi Hu, Jiaqi Li, and Muhan Zhang. Continualskillbench: Can llm agents truly evolve their capabilities?, 2026b. URL https://arxiv.org/abs/2608.03874.

Shengran Hu, Cong Lu, and Jeff Clune. Automated design of agentic systems. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025a. URL https://openreview.net/forum?id=t9U3LW7JVX.

Shengran Hu, Cong Lu, and Jeff Clune. Automated design of agentic systems. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 21344–21377, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/36b7acf6f6010652b3f2a433774a66fe-Paper-Conference.pdf.

Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6028–6039. PMLR, 2020. URL http://proceedings.mlr.press/v119/liang20a.html.

Minhua Lin, Hanqing Lu, Zhan Shi, Bing He, Rui Mao, Zhiwei Zhang, Zongyu Wu, Xianfeng Tang, Hui Liu, Zhenwei Dai, Xiang Zhang, Suhang Wang, Benoît Dumoulin, and Jian Pei. Position: Agentic evolution is the path to evolving llms. *CoRR*, abs/2602.00359, 2026a. doi: 10.48550/ARXIV.2602.00359. URL https://doi.org/10.48550/arXiv.2602.00359.

Minhua Lin, Juncheng Wu, Zijun Wang, Zhan Shi, Yisi Sang, Bing He, Zewen Liu, Tianxin Wei, Zongyu Wu, Zhiwei Zhang, Dakuo Wang, Xiang Zhang, Benoît Dumoulin, Cihang Xie, Yuyin Zhou, Suhang Wang, and Hanqing Lu. Harness updating is not harness benefit: Disentangling evolution capabilities in self-evolving LLM agents. *CoRR*, abs/2605.30621, 2026b. doi: 10.48550/ARXIV.2605.30621. URL https://doi.org/10.48550/arXiv.2605.30621.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 6467–6476, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/f87522788a2be2d171666752f97ddebb-Abstract.html.

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 *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=jL7fwchScm.

Shishir G. Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The berkeley function calling leaderboard (BFCL): from tool use to agentic evaluation of large language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/patil25a.html.

Changle Qu, Sunhao Dai, Xiaochi Wei, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Jun Xu, and Ji-Rong Wen. From exploration to mastery: Enabling llms to master tools via self-driven interactions. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 55675–55697, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/8c22e5e918198702765ecff4b20d0a90-Paper-Conference.pdf.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024. doi: 10.48550/ARXIV.2402.03300. URL https://doi.org/10.48550/arXiv.2402.03300.

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, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9229–9248. PMLR, 2020. URL http://proceedings.mlr.press/v119/sun20b.html.

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. *Trans. Mach. Learn. Res.*, 2024, 2024. URL https://openreview.net/forum?id=ehfRiF0R3a.

Kaiqi Wang, Wenjin Hou, Yuchen Yan, Hongrui Jia, Zhisheng Zhong, Botao Ren, Yifei Chen, Songyang Zhang, Yongliang Shen, Jun Xiao, Yi Yang, Yueting Zhuang, and Hehe Fan. Diving into reliable self-evolving agents: A survey. *Preprints*, September 2026. doi: 10.20944/preprints202609.0913.v1. URL https://doi.org/10.20944/preprints202609.0913.v1.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, Eli Gottlieb, Yiping Lu, Kyunghyun Cho, Jiajun Wu, Li Fei-Fei, Lijuan Wang, Yejin Choi, and Manling Li. RAGEN: understanding self-evolution in LLM agents via multi-turn reinforcement learning. *CoRR*, abs/2504.20073, 2025. doi: 10.48550/ARXIV.2504.20073. URL https://doi.org/10.48550/arXiv.2504.20073.

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, 2026. URL https://arxiv.org/abs/2607.14777.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for LLM agents. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5,*

2025, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/19909c36f51abc4856b4560aff3d36d6-Abstract-Conference.html.

Dong Yan, Jian Liang, Dapeng Hu, Ran He, Nicholas Jing Yuan, Qi Zhang, and Tieniu Tan. Agentstream: How well do self-evolving llm agents perform under streaming tasks?, 2026. URL https://arxiv.org/abs/2608.00155.

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

Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. Self-distilled RLVR. *CoRR*, abs/2604.03128, 2026a. doi: 10.48550/ARXIV.2604.03128. URL https://doi.org/10.48550/arXiv.2604.03128.

Yutao Yang, Junsong Li, Qianjun Pan, Bihao Zhan, Yuxuan Cai, Lin Du, Jie Zhou, Kai Chen, Qin Chen, Xin Li, Bo Zhang, and Liang He. Autoskill: Experience-driven lifelong learning via skill self-evolution. *CoRR*, abs/2603.01145, 2026b. doi: 10.48550/ARXIV.2603.01145. URL https://doi.org/10.48550/arXiv.2603.01145.

Z.ai. GLM-5.2: Built for long-horizon tasks. https://z.ai/blog/glm-5.2, June 2026. Accessed: 2026-06-22.

Yunpeng Zhai, Shuchang Tao, Cheng Chen, Anni Zou, Ziqian Chen, Qingxu Fu, Shinji Mai, Li Yu, Jiaji Deng, Zouying Cao, Zhaoyang Liu, Bolin Ding, and Jingren Zhou. Agentevolver: Towards efficient self-evolving agent system. *CoRR*, abs/2511.10395, 2025. doi: 10.48550/ARXIV.2511.10395. URL https://doi.org/10.48550/arXiv.2511.10395.

Guibin Zhang, Xun Xu, Yanwei Yue, Zikun Su, Wangchunshu Zhou, Xiaohu You, and Shuicheng Yan. Opd-evolver: Cultivating holistic agent evolver via on-policy distillation. *CoRR*, abs/2606.17628, 2026a. doi: 10.48550/ARXIV.2606.17628. URL https://doi.org/10.48550/arXiv.2606.17628.

Haozhen Zhang, Quanyu Long, Jianzhu Bao, Tao Feng, Weizhi Zhang, Haodong Yue, and Wenya Wang. Memskill: Learning and evolving memory skills for self-evolving agents. *CoRR*, abs/2602.02474, 2026b. doi: 10.48550/ARXIV.2602.02474. URL https://doi.org/10.48550/arXiv.2602.02474.

Jenny Zhang, Shengran Hu, Cong Lu, Robert T. Lange, and Jeff Clune. Darwin godel machine: Open-ended evolution of self-improving agents. *CoRR*, abs/2505.22954, 2025a. doi: 10.48550/ARXIV.2505.22954. URL https://doi.org/10.48550/arXiv.2505.22954.

Jenny Zhang, Shengran Hu, Cong Lu, Robert Tjarko Lange, and Jeff Clune. Darwin gödel machine: Open-ended evolution of self-improving agents. In *The Fourteenth International Conference on Learning Representations*, 2026c. URL https://openreview.net/forum?id=pUpzQZTvGY.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, Bingnan Zheng, Bang Liu, Yuyu Luo, and Chenglin Wu. Aflow: Automating agentic workflow generation. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025b. URL https://openreview.net/forum?id=z5uVAKwmjf.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, Urmish Thakker, James Zou, and Kunle

Olukotun. Agentic context engineering: Evolving contexts for self-improving language models. In *The Fourteenth International Conference on Learning Representations*, 2026d. URL https://openreview.net/forum?id=eC4ygDs02R.

Shengtao Zhang, Jiaqian Wang, Ruiwen Zhou, Junwei Liao, Yuchen Feng, 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. *CoRR*, abs/2601.03192, 2026e. doi: 10.48550/ARXIV.2601.03192. URL https://doi.org/10.48550/arXiv.2601.03192.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: LLM agents are experiential learners. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan (eds.), *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2024, February 20-27, 2024, Vancouver, Canada*, pp. 19632–19642. AAAI Press, 2024. doi: 10.1609/AAAI.V38I17.29936. URL https://doi.org/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. *CoRR*, abs/2601.18734, 2026. doi: 10.48550/ARXIV.2601.18734. URL https://doi.org/10.48550/arXiv.2601.18734.

Boyuan Zheng, Michael Y. Fatemi, Xiaolong Jin, Zora Zhiruo Wang, Apurva Gandhi, Yueqi Song, Yu Gu, Jayanth Srinivasa, Gaowen Liu, Graham Neubig, and Yu Su. Skillweaver: Web agents can self-improve by discovering and honing skills. *CoRR*, abs/2504.07079, 2025. doi: 10.48550/ARXIV.2504.07079. URL https://doi.org/10.48550/arXiv.2504.07079.

Yifei Zhou, Sergey Levine, Jason E. Weston, Xian Li, and Sainbayar Sukhbaatar. Self-challenging language model agents. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/a5a305fac88fb4ae40969cfec5eef48d-Abstract-Conference.html.

APPENDIX CONTENTS

A Experimental Setup **15**

A.1 Datasets 15

A.2 Evaluation Metrics 15

A.3 Baselines 15

A.4 Implementation Details 17

B Limitations and Future Work **18**

C Broader Impact **18**

D Prompt Templates Display **19**

APPENDIX

A EXPERIMENTAL SETUP

A.1 DATASETS

Here, we provide detailed descriptions of the datasets used in our experiments:

- **Berkeley Function Calling Leaderboard (BFCL)** (Patil et al., 2025) evaluates the ability of language models to invoke external functions and tools correctly in realistic interaction scenarios. We use the Base Multi-Turn split introduced in BFCL V3, which contains 200 multi-turn tasks where information required for tool execution is provided through the current user request, previous execution results, or the initial environment state. Solving these tasks requires the agent to maintain conversational context, select appropriate functions, and execute multi-step tool calls while tracking changes in the underlying environment. The native multi-turn evaluator determines task correctness through state-based and response-based checks.
- **BrowseComp-Plus** (Chen et al., 2026) evaluates deep-research agents on complex queries that require iterative retrieval, reasoning, and evidence synthesis. It contains 830 queries derived from BrowseComp and replaces live web search with a fixed corpus of approximately 100K web documents, including human-verified supporting evidence and challenging negative documents. This controlled retrieval environment enables reproducible evaluation while preserving the difficulty of identifying and integrating information distributed across multiple sources. We evaluate agents by the correctness of their final answers using the benchmark's native evaluation protocol.
- **Tau2** (Barrès et al., 2025) evaluates conversational agents in dual-control environments where both the agent and the user can take actions that modify a shared dynamic system. Following AgentStream, we use its Telecom domain, in which an agent must diagnose and resolve technical-support problems while coordinating with a simulated user who can directly manipulate the state of a virtual device through user-side tools. The domain covers issues such as cellular service, mobile data, and MMS, and requires agents to combine backend tool use with effective instructions that guide user actions. Task completion is determined programmatically from the resulting environment state and task-specific assertions.

A.2 EVALUATION METRICS

A.3 BASELINES

We compare BORDER with a non-evolving reference and three families of self-evolution methods: harness-based approaches that update external context or memory, outcome-based reinforcement

learning, and approaches that incorporate self-distillation or skill-based supervision. During evaluation, Vanilla and the parameter-based methods condition on the standard task prompt and the interaction history provided by the environment. Harness-based methods additionally condition on accumulated context or retrieved memories.

Non-evolving reference.

- *Vanilla.* We directly evaluate the instruction-tuned backbone without additional parameter optimization or experience accumulation across tasks. At each interaction step, the model conditions only on the task specification and the observation–action history accumulated within the current task. This baseline provides a reference for agent performance without streaming self-evolution.

Harness-based evolution.

- *ACE* (Zhang et al., 2026d). Agentic Context Engineering (ACE) maintains an evolving natural-language playbook that captures reusable strategies, operational guidance, and lessons from previous interactions. Its workflow consists of three roles: a Generator that solves tasks using the current playbook, a Reflector that analyzes completed trajectories and feedback to extract lessons from successes and errors, and a Curator that incorporates these lessons into the playbook. The Curator produces localized updates to individual entries, while refinement and deduplication control redundancy as experience accumulates. The updated playbook is supplied as additional context for subsequent tasks. Throughout this process, the backbone parameters remain fixed, and adaptation is implemented through changes to the external context.
- *ReasoningBank* (Ouyang et al., 2026). ReasoningBank maintains an external memory of reusable reasoning strategies extracted from previous interaction trajectories. For an incoming task, it retrieves relevant past experiences through embedding-based similarity search and incorporates their associated memory items into the agent's context. After the interaction, the completed trajectory is analyzed to construct new memory items. In the original formulation, an LLM judge assesses whether the trajectory succeeded, and separate extraction prompts derive effective strategies from successful attempts and lessons about potential pitfalls from failed attempts. Each memory item contains a title, a brief description, and the corresponding reasoning guidance. Newly extracted items are appended to the memory bank for use in subsequent tasks. The backbone parameters remain fixed, so accumulated experience influences behavior through memory retrieval and context augmentation.

Outcome-based reinforcement learning.

- *GRPO* (Shao et al., 2024). Group Relative Policy Optimization (GRPO) is a critic-free reinforcement learning algorithm that estimates relative advantages from multiple sampled trajectories for the same task. Outcome rewards are normalized within each rollout group, and the resulting trajectory-level advantages are used to optimize the policy with a clipped importance-ratio objective. In our implementation, supervision is derived exclusively from terminal environment rewards, without additional process labels, teacher predictions, or skill-conditioned signals.

Self-distillation and skill-distillation methods.

- *OPSD* (Zhao et al., 2026). On-Policy Self-Distillation (OPSD) constructs student and teacher views from the same underlying policy while conditioning them on different information. The student operates under the standard task context and generates on-policy trajectories, whereas the teacher is additionally given privileged information available only during training. The teacher then evaluates the student-sampled tokens and provides dense token-level supervision through distribution matching. Gradients are not propagated through the teacher branch, and the privileged context is removed at inference time.
- *RLSD* (Yang et al., 2026a). RLSD incorporates a privileged self-teacher into reinforcement learning to refine token-level credit assignment. Instead of introducing a separate distribution-matching objective, it uses the log-probability discrepancy between teacher and student to construct a bounded coefficient that modulates the magnitude of each token's GRPO update. The environment-derived advantage continues to determine the update direction, while the teacher

Table 3: Base training hyperparameters for BORDER.

Hyperparameter	Value
Tasks per benchmark (train / holdout)	64 / 32
Stream mode	Interleaved, single pass
Tasks per update	10
Rollout group size N	6
Learning rate	1×10^{-6} (constant)
PPO clip parameter ϵ_{clip}	0.2
KL coefficient β_{KL}	0.01 (low-variance KL)
Sharpness of OPD gate β_{opd}	5.0
OPD loss coefficient λ_{opd}	0.01
Maximum prompt / response length	38,912 / 512 tokens
Maximum interaction steps	40

Table 4: Experience Related hyperparameters of BORDER.

Hyperparameter	Value
Distillation scope	Successful trajectories only
OPD normalization	Response tokens
Sibling-resampled groups per step	≤ 4
Reference rendering cap	6,000 characters
Resampling advantage baseline	Source group
Pool capacity C	48
Admission threshold η	0.6
Text encoder / retrieval threshold	`all-MiniLM-L6-v2` / 0.3
Window W	10 steps
Pool-resampled groups per step	≤ 3

controls how strongly individual tokens are optimized. The influence of self-distillation is gradually reduced over training, causing optimization to approach the standard GRPO objective at later stages.

- *SEED* (Wu et al., 2026). SEED combines outcome-based reinforcement learning with on-policy distillation from self-generated hindsight skills. An initial supervised fine-tuning stage equips the policy to extract natural-language skills from completed interaction trajectories. During reinforcement learning, the current policy serves as both the rollout actor and the trajectory analyzer, deriving hindsight guidance from successful and failed attempts. The sampled action tokens are then re-scored under ordinary and skill-augmented contexts, with the latter providing a privileged self-teacher whose outputs are detached from gradient computation. The resulting log-probability differences determine token-level weights for an auxiliary distillation objective, which is jointly optimized with outcome-based GRPO. After each update, the updated policy is used for both interaction and skill extraction, allowing hindsight supervision to evolve alongside the policy. At inference time, the agent uses only the standard task context, without hindsight skills or an additional analyzer.

For all post-training baselines, we use the same backbone models, environment interfaces, and evaluation protocol as BORDER. Where applicable, we additionally match the task batch size, rollout budget, rollout group size, and number of policy updates. Consequently, the main differences among methods arise from their optimization objectives and from whether retrieved skills or other privileged information are available during training or evaluation.

A.4 Implementation Details

External analyzer at the SFT stage. We initialize the policy's trajectory-analysis capability through SFT on externally annotated trajectory–skill pairs. The corresponding backbone first col-

lects rollouts by interacting with the environment. Each completed episode is then converted into an annotation input containing the task instruction, the full observation–action sequence, and the terminal outcome. We use GLM-5.2 (Z.ai, 2026) to generate an episode summary and a natural-language hindsight skill with the analysis template P2 in Appendix **??**. Annotation uses a temperature of 0.0 and a maximum output length of 4,096 tokens. The resulting trajectory–skill pairs provide supervision for teaching the backbone to extract behavioral guidance from completed interactions. The external model is used only for offline annotation; it neither collects trajectories nor provides online supervision during RL.

Actor and analyzer prompts at the RL stage. During RL, the current policy checkpoint serves as both the actor and the trajectory analyzer. Appendix **??** provides the complete templates. The actor prompt (P1) includes the task description and context, recent interaction history, current observation, and available action specifications. It requests reasoning within `<think>...</think>` tags, followed by exactly one JSON-formatted action within `<action>...</action>` tags. For a verified successful episode, the analysis prompt (P2) returns an `episode_summary` and an `episode_skill` describing the decision rule and action ordering underlying the successful behavior. The admission prompt (P3) assesses the candidate skill's transferability and rewrites it into an imperative description of at most 60 words, removing instance-specific details while retaining its procedure. The three experience-injection blocks serve distinct purposes: P4a supplies a successful same-task trajectory for sibling-guided resampling, P4b supplies a retrieved global skill for pool-guided resampling, and P4c supplies the trajectory's own episode-level skill for teacher rescoring in local OPD. Additional references used during resampling are removed when evaluating the policy-gradient objective, and the student branch in OPD likewise uses the ordinary interaction context.

Training hyperparameters. Tables 3 and 4 summarize the base training configuration and experience-related settings, respectively. The reported interleaved configuration uses 64 training tasks and 32 held-out tasks per benchmark, with a single pass over the training stream. Each update processes 10 incoming tasks and collects a main-pass group of six trajectories per task. Outcome-based policy updates use both successful and failed trajectories, whereas local OPD is restricted to successful trajectories with a positive mean log-likelihood gain under the skill-augmented context. The OPD loss is normalized by the total number of valid response tokens in the training batch, including tokens excluded from distillation by the filters. For experience-guided resampling, reward centering uses the originating task's main-pass group, and the normalization scale is derived from main-pass rewards rather than recomputed within the resampled group. We allow at most four sibling-guided and three pool-guided resampling groups per update, with successful reference trajectories rendered within a budget of 6,000 characters. The global pool holds up to 48 skills. Retrieval uses the frozen `all-MiniLM-L6-v2` encoder and a cosine-similarity threshold of 0.3.

Computing details. Training is conducted on two NVIDIA H200 GPUs.

B Limitations and Future Work

Our evaluation covers two backbone models, three benchmarks, and the Isolated and Interleaved scenarios, leaving generalization to larger models and longer, open-ended task streams to future study. Moreover, BORDER relies on environment feedback to identify successful trajectories and learn from transferred experience; noisy or delayed feedback may weaken both learning paths. Skill abstraction and likelihood-based gating provide practical criteria for experience selection, but do not guarantee transferability or prevent all interference. Hindsight analysis, guided interactions, and continual parameter updates also incur computational overhead. Future work will investigate broader streaming settings, experience selection under uncertain feedback, and more efficient hierarchical learning with explicit interaction and computation budgets.

C Broader Impact

BORDER explores how LLM agents can turn interaction experience into persistent capabilities while regulating its reuse across tasks. Such mechanisms could support assistants that adapt to

changing workflows and carry useful knowledge into subsequent interactions. However, continual learning may also preserve biased or misleading experience, propagate harmful guidance across tasks, or retain sensitive information in skills and policy parameters. The proposed filters assess learning signals and do not establish safety or privacy guarantees. Responsible deployment should therefore include careful handling of retained experience, monitoring of evolving behavior, and human oversight in consequential applications.

D PROMPT TEMPLATES DISPLAY

We provide the prompt templates used for policy interaction (P1), hindsight skill extraction (P2), skill admission and de-instantiation (P3), and experience injection (P4). Placeholders are shown in blue monospace within curly braces. For P4, reference solutions and general skills guide experience-conditioned rollouts, while episode-level skills provide hindsight context for teacher re-scoring during distillation.

P1 Policy Interaction

{benchmark_intro}

Your task is: {task_description}
Task context: {task_context}

Prior to this step, you have already taken {step_count} step(s). Below are the most recent {history_length} observations and the corresponding actions you took:
{action_history}

You are now at step {current_step} and your current observation is: {current_observation}

The actions you may take (name, description, arguments) are:
{available_actions}

Now it's your turn to take an action. You should first reason step-by-step about the current situation. This reasoning process MUST be enclosed within <think> </think> tags.

Once you've finished your reasoning, choose exactly one action and present it within <action> </action> tags as a JSON object of the form

{"name": "<action_name>", "arguments": {...}}.

P2 Hindsight Skill Extraction

Analyze the following agent episode and return ONLY valid JSON.

You need to complete two fields:
1. Write a concise episode_summary.
2. Write one episode_skill that extracts the successful trajectory into workflow: the core decision rule and action ordering that made this trajectory work.

Important constraints:
- Write episode_skill as one short policy-facing skill, not as retrospective explanation of the trajectory.
- Return only these top-level fields: episode_summary, episode_skill.

Return format:

```
{
  "episode_summary": "string",
  "episode_skill": "string"
}
```

Episode context:
- Task description: {task_description}
- episode_success: success
- Interaction trajectory: {trajectory}

P3 Skill Admission and De-instantiation

You are auditing a ”skill” extracted from an agent’s successful episode. A skill is useful when an agent facing a different task of the same kind could follow it; it is useless when it only restates what happened in that one episode or depends on details specific to it.

Candidate skill:
[{skill}]

Rate the candidate’s transferability from 0 to 10. Then rewrite it so it applies to any task of the same kind: remove names, ids, values, item lists and other instance details; keep the decision rule and the order of actions; at most 60 words; imperative voice.

Reply with ONLY a JSON object:

```
{
  "score": <integer 0-10>,
  "generalized_skill": "<text>"
}
```

P4a Experience Injection: Reference Solution

Reference Solution

A successful attempt at this exact task, listed as observation to action steps. Use it to decide the current action:
[{reference_solution}]

P4b Experience Injection: General Skill

General Skill

Refer to this general skill, distilled from experience across many episodes, when deciding what action to take:
[{global_skill}]

P4c Experience Injection: Episode-Level Skill

Episode-Level Skill

Refer to this episode-level skill when deciding what action to take in the current episode:
[{episode_skill}]