Evolving Agents

Leonardo Ranaldi
ILCC, School of Informatics, University of Edinburgh, United Kingdom
OMNIA Lab & University of Rome Tor Vergata, Italy
Idiap Research Institute, Switzerland
{first_name.last_name}@ed.ac.uk

Abstract

AI agents powered by LLMs plan, reason and use tools well, but remain brittle to adapt and evolve because they fail to abstract the mechanisms underlying problem-solving. Reasoning, memory, and decision-making remain fragmented, leaving the systematic reuse of experience for adaptation at scale out of reach. In this paper, we introduce EvA (**Ev**olving **A**gents), a framework to improve evolution in agentic reasoning that leverages *quasi-symbolic abstractions*—semi-structured dynamic representations that furnish constructs to distil and reuse reasoning mechanisms. They are learned and refined through experience, and when instantiated, deliver meta-states for organising reasoning. EvA orchestrates this mechanism via a *Perceptor* modelling observation and action, an *Actor* conditioning its policy on these abstractions, and a *Controller* overseeing structure to explore paths or initiate rollbacks. As experience accumulates, EvA refines both its abstractions and the modules that construct and instantiate them by learning past trajectories into reusable mechanisms. Our initial analysis shows that EvA improves accuracy and adapts under mid-episode distribution shifts that cause agents to plateau on complex reasoning and interactive planning tasks. These results position EvA as a step towards adaptive reasoning, memory, and meta-control.

1 Introduction

AI agents powered by LLMs plan and invoke tools well, effectively decomposing goals into actionable sub-tasks across a growing range of benchmarks (Yao et al., 2023b; Du et al., 2025; Qu et al., 2025). The step-wise alternation between reasoning and execution makes them compelling, setting a practical foundation for solving complex processes and enabling a rapid evolution of the activities they undertake, exposing them to a growing experience formed by a combination of scenarios and interactions (Wu et al., 2025; Sapkota et al., 2026).

Despite serving as frontline proxies, current agents fail to take advantage of the collected experience and systematically struggle to scale up these mechanisms into reusable competence (Ouyang et al., 2026), resulting in repetition of similar mistakes (Zhu et al., 2025) and losing experience gained from related solutions (Zhang et al., 2025).

While recent efforts structure problem solutions into abstract space representations (Ranaldi et al., 2025; Ranaldi and Pucci, 2025; Qu et al., 2026), introduce reasoning memories to distil experiences (Feng et al., 2025; Ouyang et al., 2026; Yu et al., 2026; Xia et al., 2026), or architectures with meta-control mechanisms (Zhang et al., 2026; Dupoux et al., 2026; He et al., 2026); they still overlook a unified model to merge and orchestrate reasoning, memory, and control into a common space.

We propose EvA (**Ev**olving **A**gents), a foundational framework to improve agentic reasoning evolution by leveraging *quasi-symbolic abstractions*– semi-structured representations that are learned and refined through experience, integrated in memory, and when used, furnish dynamic meta-states that scaffold the solving process on demand. EvA orchestrates reasoning and evolution processes through three operating modules (see Figure 1). A *Perceptor* dynamically distils and refines abstractions from observations and actions; an *Actor* grounds its policy and conditions its actions on the abstractions; and a *Controller* oversees the abstraction as a meta-state to explore alternative paths or trigger rollbacks. Grounding these modules in a unified representational bedrock allows a continuous feedback loop between perception, execution, and high-level reasoning. This approach introduces a promising paradigm in which agents can reason and act and, at the same time, evolve through experience—a process that arises as experience persistently reshapes the abstractions and the modules used across subsequent interactions to consolidate past trajectories into reusable mechanisms.

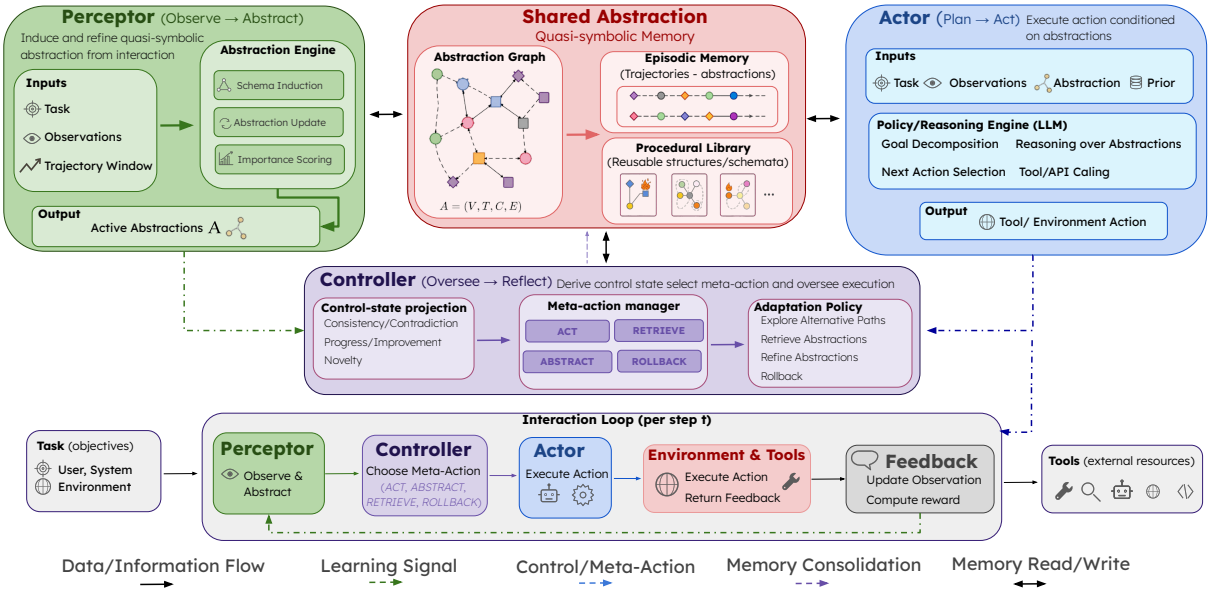

Figure 1: Overview of the EvA framework. The architecture orchestrates agentic reasoning and self-evolution via three modules (*Perceptor*, *Actor*, and *Controller*) grounded in a shared quasi-symbolic abstraction memory.

To empirically validate our foundational framework, we systematically evaluate EvA across diverse reasoning and interactive planning tasks (ALFWorld, ScienceWorld, and WebShop) and knowledge-intensive QA (Natural Questions, TriviaQA, HotPotQA, and 2WikiMultiHopQA). These tasks stress the same mechanism from different angles: interactive environments require coherent state tracking across actions, shift settings require revision after a failing trajectory, and retrieval benchmarks require past evidence to return as usable structure. The experiments show consistent improvements over static architectures. EvA reduces error rates compared to standard baselines and yields a 3.7× relative convergence ratio against Direct. Furthermore, EvA provides more robust reasoning trajectories under challenging conditions, recovering from mid-stream distribution shifts that stall common baseline agents. Ablation studies attribute these gains to the joint effect of the abstraction substrate, structural memory retrieval, and two-timescale adaptation. Additional backbone and frozen-weight evaluations show that the benefits persist across model families and across settings with fixed parameters.

Our contributions are the following:

- **EvA Architecture:** We introduce a unified *Perceptor–Actor–Controller* architecture where modules operate together with a shared memory substrate. Capabilities such as verifiable self-correction, structural memory retrieval, and tool unification emerge from this design instead of requiring separate subsystems.
- **A Unified Representational Hypothesis:** We hypothesise that a quasi-symbolic abstraction, extended from isolated problem-solving to continuous interaction trajectories, can serve as a structured summary of the execution trace, a shared interface for memory retrieval, and the source representation from which the Controller derives its compact meta-state. This common schema reduces heterogeneity across interfaces among reasoning, memory, and meta-control.
- **Modular Adaptation Strategy:** We formulate a training strategy that orchestrates the Controller and the abstraction, while updating the Perceptor and Actor. We study how this approach enforces an adaptive regime that bypasses the instability inherent in continuous meta-control updates.
- **Empirical Validation:** We provide the first empirical evidence across different backbones and diverse task families, detailing consistent and substantial gains in error reduction, distribution-shift resilience, and robustness.

These contributions make EvA a foundation for studying adaptive agentic reasoning through structured experience and open a new avenue where evolution in agents requires a shared representational layer where perception, policy, memory, and control can interact across episodes.

2 EvA: Evolving Agents

To make agents able to evolve and adapt through experience, we introduce EvA (**Ev**olving **A**gents), a foundational framework that extends reasoning by distilling experience into competencies via *quasi-symbolic abstractions* (§2.1). EvA operates through *Perceptor*, *Actor*, and *Controller* (§2.2), that reason (§2.3) and learn from experience (§2.4).

2.1 Quasi-Symbolic Abstractions Mechanism

Integrating symbolic elements into natural language reasoning is crucial to disentangle the underlying logical mechanics from surface-level semantics. While natural-language demonstrations are practical for organising problem-solving processes (Ranaldi and Freitas, 2024), they entangle logic with concrete knowledge, causing brittle and stateless behaviours. We aim to disentangle the process from content via an intermediate representation, distilling and reusing it. We extend quasi-symbolic reasoning (Ranaldi et al., 2025) to continuous agentic scenarios, defining it as a task-agnostic typed representation that isolates the task-invariant properties from its foundation, mapping the reasoning into an abstraction of transitions and outcomes:

$$\mathcal{A} = (\mathcal{V}, \mathcal{T}, \mathcal{C}, \mathcal{E}), \tag{1}$$

where $\mathcal{V}$ denotes variables, entities, tools, or states; $\mathcal{T}$ encodes the state-action transitions; $\mathcal{C}$ express constraints, preconditions; and $\mathcal{E}$ captures outcome signals (progress, success). To make continuous agentic evolution, EvA induces this structure from experience. At step t, the system observes an interaction:

$$\tau_t^{(k)} = \big((o_i, a_i, f_{i+1}, o_{i+1})\big)_{i=t-k}^{t-1}, \tag{2}$$

where o_i is the observation before action a_i, f_{i+1} is the tool, verifier, or environment feedback produced after that action, and o_{i+1} is the observation. Given the task goal g and current observation o_t, the Perceptor maps this trajectory into a quasi-symbolic abstraction:

$$\mathcal{A}_t = \mathcal{A}_{\theta_P}\big(g, o_t, \tau_t^{(k)}\big), \tag{3}$$

yielding $\mathcal{A}_t{=}(\mathcal{V}_t, \mathcal{T}_t, \mathcal{C}_t, \mathcal{E}_t)$, preserving $\mathcal{A}$ interface while reflecting $\tau_t^{(k)}$. This abstraction functions as the experience. EvA distils them into quasi-symbolic structures that serve as proxies, enabling cross context transfer and driving its evolution, surpassing the limitations of raw trajectories that entangle surface features with reasoning.

2.2 Framework Modules

EvA orchestrates reasoning and evolution via three interacting modules—the *Perceptor*, *Actor*, and *Controller*–unifying how task-solving is abstracted, executed, and distilled for systematic reuse.

Perceptor (World Modelling). The Perceptor functions as the abstraction engine, instantiating $\mathcal{A}_{\theta_P}$ on $(g, o_t, \tau_t^{(k)})$ (§2.1), and thereby mapping variables, transitions, and constraints that mark the current interaction. It also computes a sequence-confidence proxy $c_t \in \mathcal{E}_t$, the geometric-mean token probability of the generated abstraction:

$$c_t = \Big(\prod_{j=1}^{N_t} p_{\theta_P}\big(x_{t,j} \mid x_{t,<j}, g, o_t, \tau_t^{(k)}\big)\Big)^{1/N_t} \in (0, 1], \tag{4}$$

where $x_{t,1:N_t}$ denotes the tokens of $(\mathcal{V}_t, \mathcal{T}_t, \mathcal{C}_t)$ together with the fields of $\mathcal{E}_t$ generated prior to the proxy. The value c_t is appended to $\mathcal{E}_t$ once these fields are produced and is thereby excluded from its own estimate; the Controller reads it as a confidence signal (Appendix A for stable log-domain form).

Controller (Meta-Control). The *Controller* governs the reasoning cycle by deriving a compact meta-state from the active abstraction $\mathcal{A}_t$ and the memory $\mathcal{M}_t$:

$$s_t^m = \mathrm{Proj}(\mathcal{A}_t, \mathcal{M}_t) = [c_t, \delta_t, \nu_t, \ell_t], \tag{5}$$

which combines the sequence-confidence proxy $c_t \in \mathcal{E}_t$, a contradiction ratio δ_t over the active constraints $\mathcal{C}_t$, a novelty score ν_t against memory, and the normalised progress signal $\ell_t \in [0, 1]$ recorded in $\mathcal{E}_t$ (Appendix A for the component definitions).

Based on s_t^m, it parameterises a meta-policy $a_t^m \sim \Pi_\phi(\cdot \mid s_t^m)$ over four meta-actions:

- ACT: proceeds with the Actor's policy under $\mathcal{A}_t$.
- ABSTRACT: induces a revised abstraction over an extended trajectory.
- RETRIEVE: matches priors from memory.
- ROLLBACK: restores the most recent internally consistent abstraction and reasoning context. Rollback does not automatically reverse actions already executed in the environment; the Actor instead replans from the current observation.

The *Controller* regulates the interactions between perception, abstraction, memory, and action without replacing the *Actor*'s policy; it decides when the agent should continue, reuse experience, or recover from a bad reasoning path.

Actor (Policy Execution). The Actor executes the goal-directed policy through direct interaction with the environment. It conditions its actions on the quasi-symbolic structures routed by the Controller. When prior experience is retrieved and instantiated, the Actor conditions on both the active interpretation and the recovered structural prior:

$$a_t \sim \pi_{\theta_A}(\cdot \mid g, o_t, \mathcal{A}_t, \widetilde{\mathcal{A}}_t). \tag{6}$$

This allows past experience to condition the current policy. The active and grounded abstractions are serialised in the Actor context before action generation, including the generation of tool calls.

Memory. The memory $\mathcal{M}$ stores abstractions that preserve the agent's interpretation of past scenarios (states, actions, constraints, and outcomes). It has two roles: *(i) episodic*, linking abstractions to their source trajectories, and *(ii) procedural*, retaining the operative reasoning state (e.g., completed steps, tools, and active preconditions). Hence, memory grounds reasoning in the current context while allowing past experience to shape future behaviour.

The *Controller* evaluates the active abstraction $\mathcal{A}_t$. If $\mathcal{A}_t$ is insufficient due to low confidence, missing constraints, or contradictions, it issues RETRIEVE. Retrieved experiences provide stateful guidance for refining interpretations and delineating viable reasoning paths.

An abstraction $\mathcal{A}_t$ is inserted after a verified success or an instructive failure:

$$\mathcal{M} \leftarrow \mathcal{M} \cup \{\mathcal{A}_t\}. \tag{7}$$

Each stored abstraction retains a link to its source trajectory and an outcome label $y_t \in \{\text{SUCCESS}, \text{FAILURE}\}$.

Here, $r_j = n_j^+ + n_j^-$ denotes the number of attributed grounded uses of the stored abstraction $\mathcal{A}_j$. While $r_j < m$, the abstraction is protected from pruning. Once $r_j \geq m$, it is retained when $q_\eta(\mathcal{A}_j) \geq \lambda$ and pruned otherwise:

$$\mathcal{M} \leftarrow \begin{cases} \mathcal{M}, & q_\eta(\mathcal{A}_j) \geq \lambda, \\ \mathcal{M} \setminus \{\mathcal{A}_j\}, & \text{otherwise.} \end{cases} \tag{8}$$

The dynamic utility score is

$$q_\eta(\mathcal{A}_j) = \frac{n_j^+}{n_j^+ + n_j^-} \cdot \gamma^{\Delta t_j}, \tag{9}$$

where n_j^+ counts attributed grounded uses followed by successful completion, including defensive uses of failure-derived abstractions as avoidance constraints. Conversely, n_j^- counts uses followed by an attributed ROLLBACK or unsuccessful episode termination. Thus, y_j records the abstraction provenance, whilst n_j^+ and n_j^- capture the outcomes of its subsequent reuse, allowing a failure-derived abstraction to acquire $n_j^+ > 0$.

Only retrieved abstractions that survive Bind and contribute to $\widetilde{\mathcal{A}}_t$ receive a utility update. The score q_η is defined when $r_j > 0$, whilst pruning is applied only when $r_j \geq m$. The term Δt_j denotes the number of episodes since the last positively credited use, or since insertion if none has occurred. Hence, recency discounts previously useful but stale abstractions, whilst entries without evidence of positive reuse remain protected until $r_j \geq m$ and are then pruned. The hyperparameters $\eta = (\gamma, m, \lambda)$ control recency, minimum reuse evidence, and pruning. At capacity, the lowest-utility eligible entry is removed, with ties resolved by the largest Δt_j.

When memory is queried, EVA selects the prior abstractions compatible with the active one:

$$\mathcal{R}_t = \underset{K_R,\, \mathcal{A}_j \in \mathcal{M}_t}{\mathrm{ArgTopK}} \, \mathrm{sim}_\psi(\mathcal{A}_t, \mathcal{A}_j), \tag{10}$$

where $\mathrm{ArgTopK}$ returns the K_R stored abstractions gaining the highest similarity scores and sim_ψ compares their quasi-symbolic variables, states, transitions, constraints, and non-terminal outcome signals. Terminal success and failure labels are retained as metadata and are not used for structural matching.

The retrieved set $\mathcal{R}_t$ is then instantiated:

$$\widetilde{\mathcal{A}}_t = \mathrm{Bind}(\mathcal{A}_t, \mathcal{R}_t, o_t). \tag{11}$$

The Bind operator maps roles, states, and constraints from the retrieved abstractions to concrete entities in o_t, retaining only compatible mappings. Outcome labels guide instantiation: successful priors contribute reusable transitions, whilst failure-derived priors contribute constraints on transitions that previously led to failure. EVA converts a transition into an avoidance constraint only when the failure can be localised through environmental feedback, a verifier violation, or the action immediately preceding an attributed rollback; otherwise, the trajectory is retained as evidence of episodic failure without becoming a procedural prohibition. The resulting $\widetilde{\mathcal{A}}_t$ augments the active abstraction with a grounded procedural state to guide the Actor.

2.3 Reasoning Cycle

At inference time, EVA operates via a recurrent abstraction–memory–action cycle. Given a trajectory $\tau_t^{(k)}$, the *Perceptor* induces the active abstraction $\mathcal{A}_t$, representing the state space through variables, transitions, constraints, and outcome signals. The *Controller* derives the compact meta-state s_t^m from $\mathcal{A}_t$ and $\mathcal{M}_t$, and uses it to select the next meta-action:

$$a_t^m \sim \Pi_\phi(\cdot \mid s_t^m). \tag{12}$$

If $a_t^m = \textsc{Act}$, the *Actor* grounds its policy within the abstraction to produce the next action:

$$a_t \sim \pi_{\theta_A}(\cdot \mid g, o_t, \mathcal{A}_t). \tag{13}$$

If $a_t^m = \textsc{Retrieve}$, memory returns structurally compatible abstractions, which are grounded in the present context and supplied to the Actor:

$$a_t \sim \pi_{\theta_A}(\cdot \mid g, o_t, \mathcal{A}_t, \widetilde{\mathcal{A}}_t). \tag{14}$$

If $a_t^m = \textsc{Abstract}$, the Perceptor re-induces the abstraction from an extended interaction-history window $\tau_t^{(k')}$ with $k' > k$, without advancing the environment. When $a_t^m = \textsc{Rollback}$, EVA restores the most recent internally consistent abstraction and reasoning context, then replans from the current environment observation. External state is reverted only when the simulator explicitly supports checkpoint restoration.

At inference time, EVA constructs, validates, retrieves, grounds, and acts on quasi-symbolic abstractions through this recurrent cycle.

2.4 Learning Scheme

EVA separates prior meta-control calibration from interaction-time adaptation via two distinct phases.

Phase I: prior meta-control calibration. Before downstream interaction, the Controller parameters ϕ are calibrated on environments drawn from $\mathcal{D}_{\text{env}}$. During this phase, the Perceptor and Actor undergo a fixed number of temporary inner-loop updates, and the resulting post-adaptation performance supplies the reward used to update the Controller. The Controller may therefore be revised at scheduled outer-loop intervals during this calibration phase.

Phase II: interaction-time adaptation. After calibration, the resulting Controller ϕ^* is frozen. In the full single-agent configuration, only the Perceptor θ_P and Actor θ_A are updated through the fast inner loop during interaction. The projection Proj, structural similarity sim_ψ, utility rule q_η, and the typed $(\mathcal{V}, \mathcal{T}, \mathcal{C}, \mathcal{E})$ schema also remain fixed. Frozen-weight configurations preserve the same abstraction–memory–action cycle but restrict adaptation to the construction, retrieval, and consolidation of quasi-symbolic abstractions.

Experience adaptation. During interaction-time adaptation, GRPO updates the Actor from the task-outcome reward and the Perceptor from a combined reward incorporating structural consistency and novelty. Perceptor updates are gated by $\zeta_t = \mathbb{I}[c_t \geq \kappa_c]$; only sufficiently confident segments contribute, whilst KL regularisation limits drift from the pre-adaptation reference.

Protocol. For each task instance and adaptation round, we sample $G = 8$ rollouts from the same initial context. After environment or task-verifier feedback, one joint update is applied to θ_P and θ_A, while the calibrated Controller ϕ^* remains fixed. One interaction step denotes an executed environment action; one adaptation step denotes a parameter update. Updates occur between episodes, without branching or restoring the active environment. Model adapters and episodic memory persist within each run and are reset before every seed and benchmark.

Prior calibration. The Controller is calibrated to select meta-actions that support Perceptor and Actor adaptation:

$$\phi^* = \arg\max_{\phi} \; \mathbb{E}_{e \sim \mathcal{D}_{\text{env}}} \left[\mathcal{F}\big(e, \phi, \theta_P^*(\phi), \theta_A^*(\phi)\big) \right], \tag{15}$$

where $\mathcal{D}_{\text{env}}$ denotes the calibration environments, θ_P^* and θ_A^* the post-adaptation parameters (with dependence on e left implicit), and $\mathcal{F}$ rewards task success whilst penalising contradictions. We optimise ϕ with a score-function policy-gradient estimator. For each environment, the Perceptor and Actor undergo a fixed number of inner updates, followed by a Controller update from the resulting reward; the inner loop is treated as a non-differentiable black box. After calibration, the Controller remains frozen and responds to confidence drops or consistency violations through ABSTRACT and ROLLBACK.

3 Experiments

We evaluate EVA across different tasks designed to assess long-horizon interaction, resilience to non-

stationarity, and agentic retrieval. We use ALFWorld (Shridhar et al., 2021), WebShop (Yao et al., 2023a), AppWorld (Trivedi et al., 2024), reporting success rates for ALFWorld, average scores for WebShop, alongside Task Goal Completion (TGC) and Scenario Goal Completion (SGC) for AppWorld. To gauge adaptive capacity under distribution shifts, we utilise ScienceWorld (Wang et al., 2022) and ScienceWorld-Shift (SW-Shift), a variant featuring a mid-episode domain change reporting overall success rates. Finally, agentic retrieval is measured via Exact Match on Natural Questions (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), HotpotQA (Yang et al., 2018), and 2Wiki (Ho et al., 2020).

Additional Metrics. The contradiction ratio δ_t may use environment feedback or a consistency check, whereas the reported *Avg.*LER counts only externally verified violations of environment rules, task constraints, or tool preconditions:

$$\mathrm{LER} = \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}\left[v^{(n)} = 1\right], \tag{16}$$

where $v^{(n)}{=}1$ when $\tau^{(n)}$ terminates in such a violation. We report *Avg.*LER as the macro-average across benchmarks with external verification.

Relative convergence compares each method with Direct through the number of environment interactions required to reach 90% of its own final performance estimate, computed over the last 10% of the interaction budget. Values above 1 indicate faster convergence than Direct; for example, $3.7\times$ means that the threshold is reached using approximately 27% of Direct's interactions. The reported ratios are macro-averaged across the interactive benchmarks.

Baselines. We compare against different methods. *(1) Prompt-based agents*: Direct and ReAct (Yao et al., 2023b). *(2) RL-based frameworks*: GRPO (Feng et al., 2025). *(3) Skill-learning and memory-augmented RL*: SkillRL (Xia et al., 2026), which distils and evolves callable skills from interaction trajectories. *(4) Text-space skill optimisation*: GEPA (Agrawal et al., 2026) and SkillOpt (Yang et al., 2026), both of which freeze the agent's weights and iteratively refine an external free-text skill document. For the retrieval setting, we additionally compare against Search-R1 and Search-R2 (Jin et al., 2025; He et al., 2026).

EvA Configurations. To isolate the impact of our architectural choices, we evaluate different configurations that preserve the same quasi-symbolic representational substrate while varying parameter adaptation, experience consolidation, and agent orchestration. **EvA** denotes the full single-agent configuration, adapting both the Perceptor and Actor through the inner loop. **EvA-Doc** is its frozen-weight counterpart, disabling interaction-time parameter updates and relying on abstractions constructed, retrieved, and consolidated through the Controller's ABSTRACT and RETRIEVE actions. **EvA-S** retains the full single-agent adaptation scheme but restricts memory consolidation to verified successes, discarding instructive failures. **EvA-A** adapts only the Actor whilst keeping the Perceptor frozen. Finally, **EvA-M** is a frozen-weight multi-agent extension in which a Planner and specialised Executors coordinate through a shared procedural memory, $\mathcal{M}_{\text{shared}}$.

Implementation. We implement all experiments in the VeRL framework. The Perceptor, Actor, and Controller use the same base checkpoint with module-specific adapters. In the full EvA configuration and its weight-adaptive single-agent variants, the Perceptor and Actor adapters are updated at interaction time with GRPO, whilst the prompt-conditioned Controller adapter is calibrated before interaction through score-function policy-gradient updates and then frozen. EvA-Doc and EvA-M keep all module-specific adapters frozen during interaction. The principal comparison in Table 1 uses Qwen-2.5-7B, with additional backbones in Appendix D. For the text-space baselines, we follow the original protocols, using the same backbone for both the target and the optimiser and calibrating the skill document on the training split before interaction. Full configuration details are provided in Appendices B and C. All methods use the same backbone checkpoint, task splits, and maximum environment-interaction budget.

4 Results & Discussion

4.1 Main Results

EvA against prior methods. Table 1 reports principal comparison on Qwen-2.5-7B (Appendix D for additional backbones). EvA yields the strongest performance across proposed task, achieving 88.7% on ALFWORLD and 78.2% on ScienceWorld. Specifically, on ALFWorld, EvA outperforms ReAct by $+30.0$ points

Method	ALFWorld ↑	SciWorld ↑	SW-Shift ↑	WebShop Sco. ↑	AppW.-TGC ↑	AppW.-SGC ↑	*Avg.*LER ↓	Rel. Conv. ↑
Direct	42.6 (-46.1)	62.7 (-15.5)	28.6 (-32.3)	60.4 (-27.8)	–	–	–	1.0×
ReAct	58.7 (-30.0)	67.6 (-10.6)	38.4 (-22.5)	71.2 (-17.0)	18.9	9.7	25.0%	1.5×
GRPO	74.2 (-14.5)	72.8 (-5.4)	47.5 (-13.4)	80.1 (-8.1)	33.4	21.6	15.2%	1.9×
SkillRL	81.5 (-7.2)	74.3 (-3.9)	50.7 (-10.2)	83.6 (-4.6)	41.5	28.0	12.4%	2.0×
GEPA	82.8 (-5.9)	75.4 (-2.8)	–	84.0 (-4.2)	–	–	11.5%	2.2×
SkillOpt	84.3 (-4.4)	76.0 (-2.2)	–	84.9 (-3.3)	44.6	30.4	10.8%	2.3×
EvA-Doc *(frozen)*	86.5 (-2.2)	77.1 (-1.1)	55.8 (-5.1)	86.7 (-1.5)	47.1 (-4.1)	33.2 (-3.6)	8.4%	2.6×
EvA-S *(success-only)*	86.9 (-1.8)	77.4 (-0.8)	57.2 (-3.7)	86.9 (-1.3)	48.3 (-2.9)	34.1 (-2.7)	8.1%	2.9×
EvA-A *(Actor-only adapt.)*	87.4 (-1.3)	77.6 (-0.6)	58.1 (-2.8)	87.3 (-0.9)	49.8 (-1.4)	35.3 (-1.5)	7.5%	3.1×
EvA	**88.7**	**78.2**	**60.9**	**88.2**	**51.2**	**36.8**	**6.2%**	**3.7×**
EvA-M *(multi-agent)*	89.8 (+1.1)	78.6 (+0.4)	62.4 (+1.5)	88.9 (+0.7)	57.8 (+6.6)	43.5 (+6.7)	5.9%	3.8×

Table 1: Results on Qwen-2.5-7B using metrics and configurations defined in §3. Brackets are relative to EvA.

(58.7%→88.7%), the skill-based SkillRL by +7.2, and the leading text-space method, SkillOpt, by +4.4. The Average Logical-Error Rate (*Avg.*LER), measuring trajectories that terminate in active-constraint contradictions, drops from the 20–25% range typical of ReAct to just 6.2% under EvA. This indicates that the constraints encoded in $\mathcal{C}_t$ successfully anchor the agent's reasoning, mitigating the errors that are common over extended horizons. In relative convergence, EvA reaches 90% of its final performance estimate with a ratio of 3.7× against Direct, compared with 2.3× for SkillOpt.

Anatomy of the framework. To disentangle the contributions of individual design choices, Table 1 reports EvA configurations (component analysis provided in §4.2). By setting the learnable parameters, we can isolate the value of each mechanism. The frozen-weight EvA-Doc, which relies on the shared quasi-symbolic memory for adaptation, outperforms baselines, achieving 86.5% on ALFWorld. Its +2.2 margin over SkillOpt shows that a typed substrate outperforms free-text skills under similar constraints, whilst its −2.2 deficit relative to full EvA isolates the distinct benefit of inner-loop weight adaptation. These margins suggest that the **representational substrate and the weight adaptation mechanism provide complementary gains**; neither alone is sufficient.

EvA-S, which consolidates only verified success trails, performs marginally on static benchmarks but struggles under distribution shifts (−3.7 on SW-Shift). This confirms that **retaining reusable failure-derived constraints is critical for adapting to novel environments**. Finally, EvA-A, where only the Actor is adapted, bridges much of the gap but falls short of the full model, suggesting that adapting abstraction induction through the Perceptor provides benefits beyond policy optimisation.

These configurations trace a clear gradient of adaptive capacity, which is most apparent on SW-Shift, where recovery rises monotonically from the frozen EvA-Doc through the restricted variants to the full model (55.8%→57.2%→58.1%→60.9%). This progression reflects the underlying mechanisms: a fixed free-text skill remains unchanged at interaction time, a dynamically re-derived quasi-symbolic memory supports representational adaptation, and inner-loop updates provide an additional level of parameter adaptation. Relative convergence follows the same gradient (2.6 × →2.9 × →3.1 × →3.7×), indicating that the more complete configurations also reach their performance threshold with fewer interactions.

Finally, the multi-agent EvA-M outperforms the single-agent mode; the advantage is most pronounced under non-stationarity and on WebShop, where buying tasks decompose into disjoint, service-specific sub-goals. Here, a Planner can initiate targeted rollbacks on affected sub-goals without voiding the entire trajectory. Conversely, on benchmarks lacking natural decomposition, the gains are marginal; hence, EvA-M serves as an extension for decomposable tasks.

4.2 Ablation Study

Figure 2 and Table 2 report the isolated impact of each component. Removing **quasi-symbolic abstractions** produces the largest performance drops (−24.9 on ALFWorld, −24.7 on AppWorld). This shows that the structured representation forms the basis of the system; without it, the Controller lacks a typed meta-state to monitor, and episodic memory loses its structural retrieval interface. Disabling **online adaptation** also degrades outcomes (−20.6, −21.1). The *w/o Online Adaptation* variant disables both inner-loop parameter updates and interaction-time abstraction consolidation, whereas EvA-Doc freezes model parameters but retains non-parametric adaptation through abstraction construction, retrieval, and document consolidation.

Variant	ALFWorld	AppWorld
Full EvA	**88.7**	**51.2**
Component removals:		
w/o Controller	74.0 (-14.7)	35.8 (-15.4)
w/o Memory	76.2 (-12.5)	38.2 (-13.0)
w/o Quasi-Symbolic Abstractions	63.8 (-24.9)	26.5 (-24.7)
w/o Online Adaptation	68.1 (-20.6)	30.1 (-21.1)
w/o Outer Loop	80.9 (-7.8)	44.6 (-6.6)
w/o Confidence Gating	83.1 (-5.6)	46.8 (-4.4)
Frozen-weight deployment:		
EvA-Doc (typed doc, frozen)	86.5 (-2.2)	47.1 (-4.1)
EvA w/ Free-text Skills (frozen)	82.1 (-6.6)	42.3 (-8.9)
Alternative memory formats:		
EvA w/ Raw Trajectories	58.2 (-30.5)	18.4 (-32.8)
Actor Only (no abstraction)	58.7 (-30.0)	0.6 (-50.6)

Table 2: Ablation covering component removals, frozen-weight, and alternative memory formats. ALFWorld success (%) and AppWorld TGC (%).

The contrast between EvA-Doc (86.5) and EvA (88.7) therefore isolates inner-loop weight adaptation, while *w/o Online Adaptation* (68.1) quantifies the joint contribution of both mechanisms.

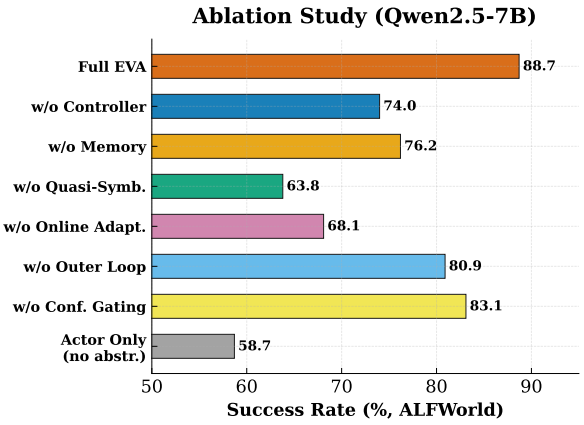

Figure 2: ALFWorld success rate per component removal.

Furthermore, removing the **Controller** causes a large drop (-14.7, -15.4), indicating that meta-level orchestration adds value beyond the abstraction mechanism alone. **Memory** is important for cross-episode generalisation (-12.5, -13.0), while undermining the **outer loop** (-7.8) and **confidence gating** (-5.6) destabilises meta-control and gradient quality, respectively (Figure 8 in Appendix E for a sensitivity analysis of the confidence threshold κ_c). The *w/o Confidence Gating* variant replaces ζ_t with an unweighted update over all segments while keeping the rest of the inner loop intact, and is therefore distinct from the κ_c=0 endpoint in Figure 8.

The Actor-only ablation removes the Controller, abstraction interface, and episodic memory while retaining the same Actor adapter and action format as EvA; it is therefore not identical to the separately implemented ReAct baseline reported in Table 1.

Turning to the frozen-weight variants, EvA-Doc maintains the typed substrate but discards the internal loop, achieving 86.5% on ALFWorld. Replacing the typed document with an otherwise matched free-text skill representation reduces performance to 82.1%. The 4.4-point discrepancy between these two frozen configurations isolates the intrinsic value of typing the shared artefact, which benefits both the Controller's meta-state and structural retrieval via sim_ψ. The subsequent 2.2-point leap from EvA-Doc to the full EvA model provides an estimate of the contribution of inner-loop weight adaptation under this matched configuration. Finally, relying on raw, unprocessed trajectories yields only 58.2%, supporting structural distillation over verbatim retention.

4.3 Analysis

The modular architecture of EvA supports a multi-agent extension (EvA-M). In this paradigm, a Planner agent dissects tasks into sub-goals—expressed as partial abstractions $\mathcal{A}_t^{(g)}$—and dispatches them to specialised executor agents operating over a shared memory, $\mathcal{M}_{\text{shared}}$. The core Perceptor–Actor–Controller organisation is retained: the Planner functions as a Controller with an expanded meta-action space (incorporating Dispatch), whilst each Executor operates as a frozen EvA instance confined to its designated sub-goal.

As shown in Figure 3, deploying EvA-M on Qwen-2.5-7B yields gains on AppWorld ($+6.6$ TGC and $+6.7$ SGC), with smaller gains on ALFWorld ($+1.1$) and WebShop ($+0.7$ average-score points). We argue that this divergence stems from the task topology: AppWorld scenarios need coordinating interdependent tasks across heterogeneous services, enabling the Planner to decompose them into service-specific sub-goals with naturally disjoint constraint subspaces $\mathcal{C}_t^{(i)}$. On the other side, ALFWorld tasks are sufficiently resolved via a single-agent process, making the decomposition overhead superfluous. The marked improvement in Scenario Goal Completion (SGC) is salient, since SGC requires the success of all constituent tasks; EvA-M's Planner can therefore roll back the procedural state associated with a failing sub-goal with-

out discarding the abstractions associated with unaffected sub-goals.

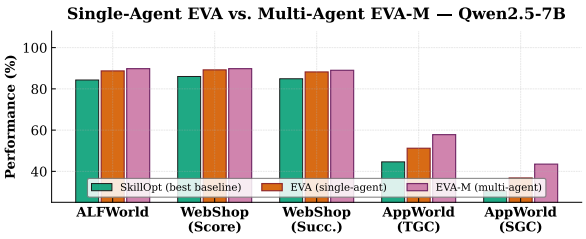

Figure 3: Full single-agent EVA vs. EVA-M.

Method	SINGLE-HOP		MULTI-HOP		AVG.
	NQ	TQA	HotQA	2Wiki	
Search-R1	39.5	56.0	32.6	29.7	39.5
Search-R2	39.9	65.9	39.0	35.8	45.2
SkillOpt	42.8	67.8	42.0	39.6	48.1
EVA	**44.8**	**69.5**	**45.7**	**43.1**	**50.8**

Table 3: Agentic retrieval (Exact Match, %; Qwen2.5-7B). EVA achieves the best average performance, with the largest margins on the multi-hop datasets.

Agentic Retrieval. Table 3 assesses EVA in the context of agentic retrieval. EVA achieves the highest overall average (50.8%) and the largest margins on the multi-hop datasets. Here, converting localised retrieval failures into reusable abstractions lets the agent avoid redundant search steps. While SkillOpt, which logs corrective rules within a free-text skill, limits the performance gap for single-hop queries, it falters on multi-hop reasoning, where the agent must track evidence across sequential steps. Furthermore, unlike Search-R2, which applies local corrections to misleading search steps and then discards them, EVA preserves the corrective abstraction in its memory for future reuse.

Recovery under distribution shift. Figure 4 reports learning curves on SW-Shift, where a domain change is injected at step 500. A frozen EVA-Doc seeded with a generic, non-calibrated document (EVA-Doc-Cold) starts from a weaker post-shift state and recovers only partially (∼49.5%), as it cannot re-derive structure consolidated under the original dynamics. The frozen-weight EVA-Doc, whose document is calibrated on the source dynamics, recovers further (∼55.8%) by extending its shared document at interaction time, yet plateaus below the full model. EVA exploits the two-timescale design most fully: the calibrated meta-policy ϕ^* registers the change through a drop in c_t and rising δ_t, triggering a phase of heightened ABSTRACT and ROLLBACK; the Perceptor then re-derives abstractions consistent with the novel dynamics, and EVA recovers to ∼60.9% within ∼150 interaction steps.

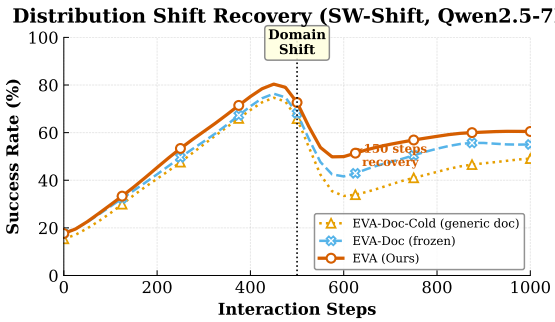

Figure 4: Learning curves on SW-Shift. We injected a domain change at step 500.

5 Related Work

Reasoning with Abstractions. Step-wise reasoning via CoT is vulnerable to content bias and hallucination (Turpin et al., 2023). Neuro-symbolic approaches couple neural perception with symbolic solvers (Freitas et al., 2025), at the cost of a brittle translation boundary. QuaSAR (Ranaldi et al., 2025) proposed quasi-symbolic abstractions for single-problem reasoning, Ranaldi and Pucci (2025) extended the idea to multilingual reasoning. EVA takes this abstraction family further, from single problems to interaction trajectories, and introduces a Controller that governs when to abstract.

Agentic LLMs and Self-Evolution. ReAct (Yao et al., 2023b) interleaves reasoning and acting but lacks persistent memory. ReasoningBank (Ouyang et al., 2026) distils transferable reasoning strategies from past experiences, and SkillRL (Xia et al., 2026) extracts and evolves callable skills from trajectories through recursive reinforcement learning. Autogenesis (Zhang et al., 2026) introduces a protocol for versioned self-evolution. Our contribution is complementary: we focus on the representation over which memories are expressed—the quasi-symbolic abstraction—and unify abstraction, memory, and meta-control within a single objective.

Text-Space Skill Optimisation. To adapt frozen agents without weight updates, a recent line treats the natural-language skill document as the trainable object. Yang et al. (2026) convert scored rollouts into bounded add/delete/replace edits on a single

skill file and accept an edit only when it improves a held-out score, importing learning-rate, validation, and momentum analogues into text space; Agrawal et al. (2026) evolve prompts through reflective feedback. These methods optimise a global free-text artefact before interaction, whilst the target model remains frozen. EvA instead induces and consolidates interaction-specific quasi-symbolic abstractions as the agent operates. In its frozen-weight configurations, adaptation occurs through this shared representational substrate, whereas the full configuration additionally updates the Perceptor and Actor.

Cognitive Architectures and Meta-Control. Dupoux et al. (2026) propose a tripartite architecture (System A, B, M) inspired by cognitive science, separating observation, action, and meta-control on evolutionary and developmental timescales. EvA operationalises this via the Perceptor–Actor–Controller triad, grounding all three systems in a shared quasi-symbolic substrate.

Future Work will extend EvA towards real-world tasks in which adaptive reasoning is central. Many consequential settings require agents to operate over incomplete evidence, evolving constraints, and feedback that may invalidate decisions. Examples include clinical-trial eligibility screening, educational planning, scientific assistance, and administrative workflows where a single mistaken assumption can propagate across several steps.

Future Objective is to study how *quasi-symbolic abstractions* can support agents in these settings by making adaptation explicit. In such tasks, an agent should preserve the path that led to a decision, revise that path when new evidence appears, and expose which constraints, experiences, and recovery steps shaped its behaviour. This direction *connects agentic performance with human oversight*: adaptive reasoning becomes useful only when people can inspect how an agent changed course and where it reused prior. We also plan to examine EvA in collaborative scenarios where agents assist human. The value of an evolving agent lies in its capacity to reduce cognitive load, surface overlooked constraints, and provide structured alternatives while leaving final judgement to the human decision-maker. This would move EvA towards a broader research programme on adaptive agents that learn from experience while remaining accountable to the people.

6 Conclusion

We introduced EvA (**Ev**olving **A**gents), a framework that enables agentic reasoning and evolution by distilling experience into quasi-symbolic abstractions—semi-structured representations that simultaneously serve as a structured summary of the execution trace, an episodic-memory interface, and the source representation from which the Controller derives its meta-state. In the full single-agent configuration, a two-timescale learning scheme separates the slow calibration of meta-control priors from the fast adaptation of perception and policy, enabling robust recovery under distributional shift. Across long-horizon tasks, agentic retrieval benchmarks, and controlled distribution-shift settings, EvA consistently outperforms state-of-the-art baselines, with the largest gains on tasks that require multi-step constraint tracking, structural memory reuse, and adaptation to non-stationarity. We view EvA as a step towards agents that continue to evolve by dynamically restructuring the representations through which they reason and act.

Limitations

Several limitations deserve discussion. First, the execution of the Controller and on-the-fly abstractions generation introduce overhead in the inner loop, and sparse routing mechanisms are a natural direction for reducing latency. Second, the abstraction schema is currently a hand-designed typed schema; scaling to high-setting multimodal inputs will require unsupervised or differentiable structure induction. The signal c_t is a likelihood proxy and retrieval relies on typed predicates such that surface-distinct entities match; confidence estimators and learned predicate are natural refinements. Third, while quasi-symbolic abstractions are auditable, we do not claim they faithfully reflect the model's internal computation: as with CoT, a correct answer does not guarantee that $\mathcal{A}_t$ captures the underlying inference. Fourth, the multi-agent extension (EvA-M) is evaluated in a preliminary setting; a more thorough study of scaling laws with team size and compositional task structure is left for future work.

Acknowledgements

I thank my supervisors, mentors, and collaborators across the University of Edinburgh, the Idiap Research Institute and the University of Rome Tor Vergata for feedback that shaped this research line.

References

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. 2026. Gepa: Reflective prompt evolution can outperform reinforcement learning. *Preprint*, arXiv:2507.19457.

Yiming Du, Wenyu Huang, Danna Zheng, Zhaowei Wang, Sebastien Montella, Mirella Lapata, Kam-Fai Wong, and Jeff Z. Pan. 2025. Rethinking memory in llm based agents: Representations, operations, and emerging topics. *Preprint*, arXiv:2505.00675.

Emmanuel Dupoux, Yann LeCun, and Jitendra Malik. 2026. Why ai systems don't learn and what to do about it: Lessons on autonomous learning from cognitive science. *Preprint*, arXiv:2603.15381.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. 2025. Group-in-group policy optimization for llm agent training. *Preprint*, arXiv:2505.10978.

André Freitas, Marco Valentino, and Danilo Silva de Carvalho. 2025. Neuro-symbolic natural language processing. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, pages 14–15, Suzhou, China. Association for Computational Linguistics.

Bowei He, Minda Hu, Zenan Xu, Hongru Wang, Licheng Zong, Yankai Chen, Chen Ma, Xue Liu, Pluto Zhou, and Irwin King. 2026. Search-r2: Enhancing search-integrated reasoning via actor-refiner collaboration. *Preprint*, arXiv:2602.03647.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *Preprint*, arXiv:2011.01060.

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

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Siru Ouyang, Jun Yan, I-Hung Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long T. Le, Samira Daruki, Xiangru Tang, Vishy Tirumalashetty, George Lee, Mahsan Rofouei, Hangfei Lin, Jiawei Han, Chen-Yu Lee, and Tomas Pfister. 2026. Reasoningbank: Scaling agent self-evolving with reasoning memory. *Preprint*, arXiv:2509.25140.

Changle Qu, Sunhao Dai, Xiaochi Wei, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Jun Xu, and Ji-rong Wen. 2025. Tool learning with large language models: a survey. *Frontiers of Computer Science*, 19(8).

Yuxiao Qu, Anikait Singh, Yoonho Lee, Amrith Setlur, Ruslan Salakhutdinov, Chelsea Finn, and Aviral Kumar. 2026. RLAD: Training LLMs to discover abstractions for solving reasoning problems. In *The Fourteenth International Conference on Learning Representations*.

Leonardo Ranaldi and Andre Freitas. 2024. Aligning large and small language models via chain-of-thought reasoning. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1812–1827, St. Julian's, Malta. Association for Computational Linguistics.

Leonardo Ranaldi and Giulia Pucci. 2025. Multilingual reasoning via self-training. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 11566–11582, Albuquerque, New Mexico. Association for Computational Linguistics.

Leonardo Ranaldi, Marco Valentino, and Andre Freitas. 2025. Improving chain-of-thought reasoning via quasi-symbolic abstractions. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 17222–17240, Vienna, Austria. Association for Computational Linguistics.

Ranjan Sapkota, Konstantinos I. Roumeliotis, and Manoj Karkee. 2026. Ai agents vs. agentic ai: A conceptual taxonomy, applications and challenges. *Information Fusion*, 126:103599.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2021. Alfworld: Aligning text and embodied environments for interactive learning. *Preprint*, arXiv:2010.03768.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. 2024. 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)*, pages 16022–16076, Bangkok, Thailand. Association for Computational Linguistics.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *Preprint*, arXiv:2305.04388.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. 2022. ScienceWorld: Is your agent smarter than a 5th grader? In *Empirical Methods in Natural Language Processing (EMNLP)*.

Junde Wu, Jiayuan Zhu, Yuyuan Liu, Min Xu, and Yueming Jin. 2025. Agentic reasoning: A streamlined framework for enhancing llm reasoning with agentic tools. *Preprint*, arXiv:2502.04644.

Peng Xia, Jianwen Chen, Hanyang Wang, Jiaqi Liu, Kaide Zeng, Yu Wang, Siwei Han, Yiyang Zhou, Xujiang Zhao, Haifeng Chen, Zeyu Zheng, Cihang Xie, and Huaxiu Yao. 2026. Skillrl: Evolving agents via recursive skill-augmented reinforcement learning. *Preprint*, arXiv:2602.08234.

Yifan Yang, Ziyang Gong, Weiquan Huang, Qihao Yang, Ziwei Zhou, Zisu Huang, Yan Li, Xuemei Gao, Qi Dai, Bei Liu, Kai Qiu, Yuqing Yang, Dongdong Chen, Xue Yang, and Chong Luo. 2026. Skillopt: Executive strategy for self-evolving agent skills. *Preprint*, arXiv:2605.23904.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2023a. Webshop: Towards scalable real-world web interaction with grounded language agents. *Preprint*, arXiv:2207.01206.

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

Hongzhuo Yu, Fei Zhu, Guo-Sen Xie, and Ling Shao. 2026. Self-consolidation for self-evolving agents. *Preprint*, arXiv:2602.01966.

Kai Zhang, Xiangchao Chen, Bo Liu, Tianci Xue, Zeyi Liao, Zhihan Liu, Xiyao Wang, Yuting Ning, Zhaorun Chen, Xiaohan Fu, Jian Xie, Yuxuan Sun, Boyu Gou, Qi Qi, Zihang Meng, Jianwei Yang, Ning Zhang, Xian Li, Ashish Shah, and 11 others. 2025. Agent learning via early experience. *Preprint*, arXiv:2510.08558.

Wentao Zhang, Zhe Zhao, Haibin Wen, Yingcheng Wu, Ming Yin, Bo An, and Mengdi Wang. 2026. Autogenesis: A self-evolving agent protocol. *Preprint*, arXiv:2604.15034.

Kunlun Zhu, Zijia Liu, Bingxuan Li, Muxin Tian, Yingxuan Yang, Jiaxun Zhang, Pengrui Han, Qipeng Xie, Fuyang Cui, Weijia Zhang, Xiaoteng Ma, Xiaodong Yu, Gowtham Ramesh, Jialian Wu, Zicheng Liu, Pan Lu, James Zou, and Jiaxuan You. 2025. Where llm agents fail and how they can learn from failures. *Preprint*, arXiv:2509.25370.

A Control-State Components

The Controller reads the control state $s_t^m = [c_t, \delta_t, \nu_t, \ell_t]$ introduced in §2.2; its components are defined as follows. The sequence-confidence proxy c_t (§2.2) is computed in the numerically stable log domain, the equivalent of the geometric mean given in the main text,

$$c_t = \exp\left(\frac{1}{N_t}\sum_{j=1}^{N_t}\log p_{\theta_P}\left(x_{t,j} \mid x_{t,<j}, g, o_t, \tau_t^{(k)}\right)\right). \tag{17}$$

The contradiction ratio measures the fraction of active constraints that the latest feedback violates,

$$\delta_t = \frac{1}{\max(1, |\mathcal{C}_t|)}\sum_{c\in\mathcal{C}_t}\chi(c, f_t), \tag{18}$$

where $\chi(c, f_t) = 1$ when the latest environment or verifier feedback f_t contradicts an active constraint c, and 0 otherwise; when explicit feedback is unavailable, χ is obtained through a consistency check.

The novelty score is $\nu_t = 1$ when $\mathcal{M}_t = \varnothing$, and otherwise

$$\nu_t = 1 - \max_{\mathcal{A}_j\in\mathcal{M}_t}\mathrm{sim}_\psi(\mathcal{A}_t, \mathcal{A}_j). \tag{19}$$

Structural similarity is computed as

$$\mathrm{sim}_\psi(\mathcal{A}, \mathcal{A}') = \frac{\sum_X w_X J(\mathrm{can}(X), \mathrm{can}(X'))}{\sum_X w_X}, \tag{20}$$

where $X \in \{\mathcal{V}, \mathcal{T}, \mathcal{C}, \mathcal{E}^\circ\}$, J is Jaccard similarity, can canonicalises typed predicates, and $\mathcal{E}^\circ$ excludes confidence and terminal outcome labels. The fixed uniform positive weights are denoted by $\psi = (w_\mathcal{V}, w_\mathcal{T}, w_\mathcal{C}, w_\mathcal{E})$, with $J(\varnothing, \varnothing) = 0$; two abstractions are thereby credited for jointly instantiated fields, whilst a shared empty field contributes no spurious similarity.

Finally, $\ell_t \in [0, 1]$ is the environment completion signal: the proportion of verified sub-goals when available, or the normalised environment score; when no intermediate signal exists, it is 0 before terminal feedback and equals the terminal outcome at completion.

B Model Versions

Table 4 reports the exact model checkpoints used in our experiments. The principal comparison, the ablations, the training-dynamics analyses, and the per-subtask breakdowns all use Qwen-2.5-7B; the per-backbone results additionally use Llama-3-8B and Mistral-7B.

Model	Checkpoint / Version
Qwen-2.5-7B	`Qwen/Qwen2.5-7B-Instruct`
Llama-3-8B	`meta-llama/Meta-Llama-3-8B-Instruct`
Mistral-v0.3-7B	`mistralai/Mistral-7B-Instruct-v0.3`

Table 4: Model checkpoints used in the experiments (Huggingface). Access verified on 05/2026.

C Implementation and Hyperparameters

Table 5 reports the main hyperparameters. The inner loop is optimised with GRPO; the outer loop calibrates the Controller parameters ϕ through a score-function policy-gradient estimator over procedurally generated environments, with the post-adaptation reward $\mathcal{F}$ as the meta-objective. The Controller is a prompt-conditioned language-model policy that maps the compact control state to one of four meta-actions. It is calibrated before interaction through score-function policy-gradient updates and then held fixed. Structural retrieval through sim_ψ is computed as a weighted Jaccard overlap over the predicate, and variables, transitions, constraints, and outcome signals each contribute to the match. All experiments are run on 4 NVIDIA H200 GPUs.

The Perceptor reward combines task, structural-consistency, and novelty signals, $R_P = R_{\text{task}} + \alpha R_{\text{cons}} + \beta R_{\text{nov}}$, while Controller calibration uses $\mathcal{F} = R_{\text{task}} - \rho R_{\text{contr}}$, with fixed coefficients α, β, and ρ.

The *w/o Outer Loop* configuration retains the Controller architecture but uses its initial prompt-conditioned policy without score-function meta-calibration. All Perceptor, Actor, memory, and retrieval components remain unchanged.

Frozen-weight configuration (EvA-Doc). The frozen-weight variant shares the architecture and the abstraction grammar above, and differs only in how adaptation is carried out. The backbone weights are fixed, and the GRPO inner loop is disabled and no gradient updates are applied at any point. In place of weight adaptation, the episodic memory of typed abstractions is serialised into a compact, human-readable document prepended to the agent context before each episode. This document is initialised on the training split prior to interaction and subsequently extended at interaction time through the Controller's non-parametric ABSTRACT and RETRIEVE actions, and the shared abstractions can be re-derived when the dynamics change. Because the artefact is typed and self-contained, a single document populated by one

Parameter	Value
Inner loop (GRPO)	
Learning rate	1×10^{-6}
KL coefficient	0.001
Clipping range (ϵ)	0.2
Group size (G)	8
Rollout temperature (train / eval)	0.9 / 0.0
Confidence gating threshold (κ_c)	0.6
Outer loop (meta-calibration)	
Meta-policy estimator	score-function
Calibration environments	50
Inner-adaptation steps (K)	5
Controller	
Implementation	prompt-conditioned policy
Action space	4 meta-actions
Output format	single action token
Episodic memory & retrieval	
Capacity	512 abstractions
Consolidation threshold (λ)	0.6
Recency-decay factor (γ)	0.95
Minimum reuse evidence (m)	3
Retrieved abstractions (K_R)	3
Trajectory window (k)	8
Structural similarity (sim_ψ)	weighted Jac, uniform

Table 5: Hyperparameters for EvA training, meta-calibration, and memory management.

agent can be reused by others without further training, which is the property we exploit when constructing the shared team procedural memory examined in the main text. Other hyperparameters (λ, γ, k, and sim_ψ) are identical to the full model, the comparison between EvA-Doc and EvA isolates the contribution of inner-loop weight adaptation.

Learnable Components. The timescale scheme (§2.4) separates interaction-time adaptation from prior meta-control calibration. The inner timescale comprises the Perceptor and Actor adapters, θ_P and θ_A, which are attached to a shared frozen backbone and updated via GRPO. The outer timescale the prompt-conditioned Controller adapter ϕ, calibrated through score-function policy gradient and then held fixed. The projection forming the control state is deterministic, sim_ψ uses fixed weighted-Jaccard coefficients, and q_η uses the fixed memory-management hyperparameters $\eta = (\gamma, m, \lambda)$. The schema is hand-designed, whilst Bind is implemented via prompted grounding.

D Per-Backbone Results

Table 1 reports the principal comparison on Qwen-2.5-7B. To improve the conclusions, Tables 7 and 8 present the same comparison on Llama-3-8B and Mistral-v0.3-7B under an identical protocol. EvA

Component	Param.	Timescale	Update / Signal
Perceptor adapter	θ_P	inner (fast)	GRPO combined reward + gate ζ_t
Actor adapter	θ_A	inner (fast)	GRPO task reward
Controller adapter	ϕ	outer (slow)	policy gradient
Projection	–	fixed	deterministic
Structural retrieval	ψ	fixed	weighted Jaccard
Utility rule	$\eta = (\gamma, m, \lambda)$	fixed	count and recency
Bind	–	non-parametric	prompted grounding
Schema $(\mathcal{V}, \mathcal{T}, \mathcal{C}, \mathcal{E})$	–	fixed	hand-designed

Table 6: Learnable Components of EvA.

achieves the best performances, the frozen-weight EvA-Doc remains the strongest method that applies no weight updates and trails the full model by a small and consistent margin, and the widest separations again appear on SW-Shift, where a skill consolidated before interaction cannot be re-derived after the mid-stream domain change. The absolute scores on Llama-3-8B are uniformly lower, which is consistent with its weaker base reasoning capacity, yet the relative gains of EvA over the text-space and skill-based baselines are of comparable magnitude, indicating that the contribution of the typed substrate does not depend on the strength of the underlying backbone.

Method	ALF ↑	SciW ↑	Shift ↑	WebShop ↑
Direct	31.0	44.8	18.2	48.3
ReAct	45.8	53.8	24.9	60.1
GRPO	61.3	58.7	35.4	68.4
GEPA	70.4	61.8	39.1	71.9
SkillOpt	71.2	62.8	39.8	72.6
EvA-Doc *(frozen)*	74.1	63.6	48.7	74.5
EvA (ours)	**76.4**	**64.5**	**54.3**	**76.2**

Table 7: Results on Llama-3-8B (success rate, %).

Method	ALF ↑	SciW ↑	Shift ↑	WebShop ↑
Direct	40.1	63.5	30.4	58.7
ReAct	55.3	70.5	39.8	69.8
GRPO	72.6	76.4	49.1	78.6
SkillRL	79.4	78.9	52.3	82.1
SkillOpt	82.1	80.4	51.2	83.4
EvA-Doc *(frozen)*	84.3	81.2	57.6	85.0
EvA (ours)	**86.5**	**82.0**	**62.9**	**86.7**

Table 8: Results on Mistral-v0.3-7B (success rate, %).

E Additional Training Analysis

Controller behaviour over training. Figure 5 tracks the distribution of Controller meta-actions over the interaction. During the early stages of an episode (steps 0–100), the Controller often issues Abstract (∼35%) and Retrieve (∼25%),

reflecting the need to build and recall structural representations before the agent can act effectively. As experience accumulates and the episodic memory supplies reusable patterns, ACT becomes dominant ($\sim$70% by step 400), indicating that the agent has internalised sufficient structure to operate directly from its active abstraction. The ROLLBACK action remains comparatively infrequent ($\sim$5–8%) and rises only when the agent encounters configurations that contradict the current abstraction. This trajectory reflects a gradual shift from world modelling to policy execution: as memory supplies more compatible abstractions, the fixed Controller receives more reliable control states and selects ACT more frequently.

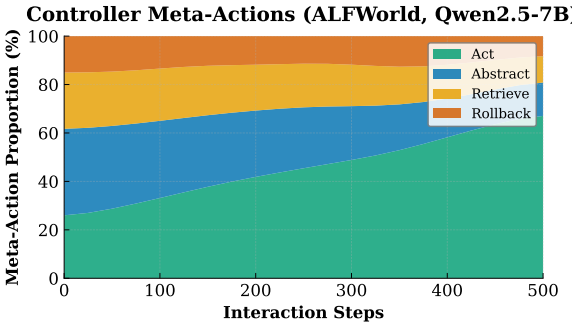

Figure 5: Controller meta-action distribution over interaction steps (ALFWorld).

Training dynamics on ALFWorld. Figure 6 presents the training curves on ALFWORLD for GRPO, SkillRL, the frozen-weight EVA-Doc, and the full model. The same two-phase pattern observed recurs here. The methods improve at a similar rate whilst episodic memory is still sparse, with EVA-Doc tracking the full model closely throughout this phase. A divergence then emerges as the accumulated abstractions compound, and EVA pulls clear of the baselines once the Controller shifts from frequent abstraction construction towards direct ACT decisions supported by the increasingly mature memory. The frozen-weight EVA-Doc continues to improve with document accumulation but plateaus below the full model once its shared document saturates, making visible the headroom that inner-loop weight adaptation contributes.

Episodic memory dynamics. Figure 7 and Table 9 characterise the episodic memory over 50 evaluation episodes. The utility score q_η of successful abstractions grows from $\sim$0.47 to $\sim$0.80, reflecting the accumulation of reliable patterns that

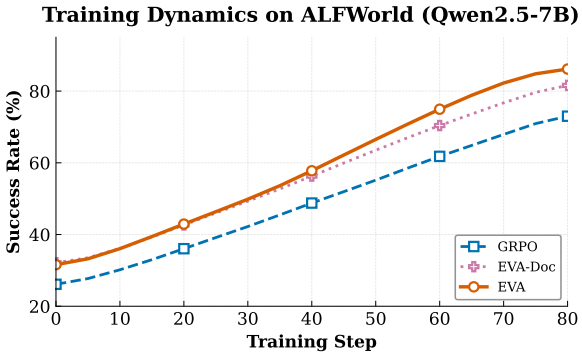

Figure 6: Training dynamics on ALFWORLD (success rate, Qwen-2.5-7B) for GRPO, the frozen-weight EVA-Doc, and full EVA. EVA diverges from the baselines once the episodic memory accumulates reusable abstractions, whilst EVA-Doc plateaus once its shared document saturates.

are retrieved frequently and lead to successful outcomes. The reported averages include reused entries with $0 < r_j < m$, which remain protected from pruning. Failure-derived abstractions that support recovery acquire positive utility through successful defensive reuse. Those that fail to accumulate positive reuse evidence are pruned once eligible, whilst previously useful but stale entries decay through the recency factor $\gamma^{\Delta t_j}$, producing an average pruning of 28.7 obsolete entries. The structural retrieval mechanism achieves a 74.2% hit rate, defined as the proportion of retrievals for which at least one of the top-three matches is task-relevant, against 53.1% for cosine similarity over mean-pooled embeddings. This 21.1-point gap confirms the value of typed predicate matching through sim_ψ. The controlled growth from q_η-based pruning contrasts with append-only skill banks (Xia et al., 2026), which accumulate every procedure without quality filtering, and allows EVA to maintain a compact, high-quality memory form training.

Metric	Value
Abstractions stored	163.4 $\pm$ 14.2
– Successful trajectories	98.1 $\pm$ 9.6 (60.0%)
– Instructive failures	65.3 $\pm$ 10.8 (40.0%)
Elig. abs. pruned ($r_j \geq m,\ q_\eta < \lambda$)	28.7 $\pm$ 6.3
Avg. retrievals per episode	4.8 $\pm$ 1.3
Structural retrieval hit rate (top-3)	74.2%
vs. embedding cosine baseline	53.1%
Avg. $\lvert\mathcal{V}_t\rvert$ (variables)	7.8 $\pm$ 2.3
Avg. $\lvert\mathcal{T}_t\rvert$ (transitions)	5.2 $\pm$ 1.6
Avg. $\lvert\mathcal{C}_t\rvert$ (constraints)	3.8 $\pm$ 1.4

Table 9: Episodic memory statistics at the end of evaluation (ALFWORLD, Qwen-2.5-7B, 50 episodes).

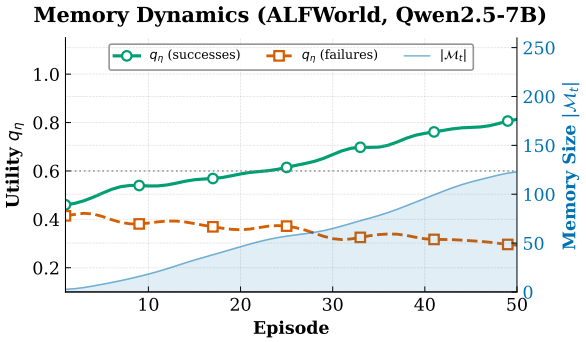

Figure 7: Memory utility and growth over episodes on ALFWORLD. Successful abstractions and useful failure-derived constraints accumulate positive utility, whilst unsupported or stale entries fall below λ and are pruned.

Confidence gating sensitivity. Figure 8 reports the sensitivity of EVA to the confidence gating threshold κ_c that determines which trajectory segments receive gradient updates. Performance peaks at κ_c=0.6 on both ALFWORLD (88.7%) and AppWorld (51.2% TGC). Below this threshold, unreliable abstractions propagate noisy gradients that destabilise the Perceptor; above it, too many segments are suppressed, and the inner loop is starved of training signal. The curve is reasonably flat across $[0.5, 0.7]$, which suggests the threshold is not overly sensitive, although the extremes degrade performance substantially (κ_c=0.0: -10.7 on ALFWorld; κ_c=0.9: -13.8), confirming that confidence gating plays a meaningful role in training stability. These results suggest that the sequence-likelihood proxy provides a useful gating signal, although it is not a calibrated estimate of structural correctness.

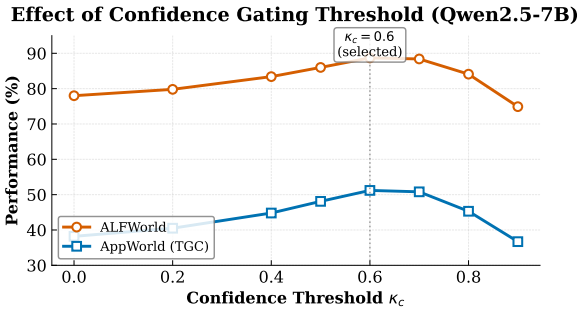

Figure 8: Performance as a function of the confidence gating threshold κ_c.

Cross-model transfer. Quasi-symbolic abstractions encode reasoning patterns at a semantic level independent of the inducing model's token-level biases. Figure 9 evaluates whether the episodic memory induced by one backbone (Llama-3-8B) can be reused by a different backbone (Qwen-2.5-7B) without fine-tuning. Across all three benchmarks, the transfer condition approaches the self-induced upper bound, with a residual gap of at most 2.4 points on SCIENCEWORLD, 3.2 on WebShop, and 4.5 on SW-Shift. The narrowness is notable on SW-Shift, where the transferred abstractions must support adaptation to an environment that the inducing model never experienced. The result provides initial evidence that the transferred abstractions remain useful across model families, although the comparison does not isolate memory provenance from every aspect of the receiver configuration. This portability underlies the frozen-weight EVA-Doc configuration, in which the serialised document is populated by one agent and consumed by others, and provides the empirical foundation for the shared team procedural memory discussed in the main text.

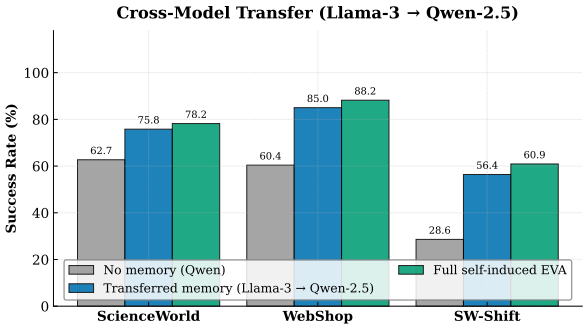

Figure 9: Cross-model transfer across benchmarks. An episodic memory induced by Llama-3-8B is applied to Qwen-2.5-7B without tuning.

Abstraction quality during meta-control calibration. Figure 10 tracks abstraction quality during the prior meta-control calibration phase. The vertical line at step ∼80 marks the first scheduled outer-loop update of the Controller. After calibration is completed, the resulting Controller is frozen throughout interaction-time adaptation and evaluation. The structural retrieval hit rate (left) and the contradiction rate in the induced abstractions (right) evolve as follows. At initialisation, the hit rate is low (∼38%), because the earliest abstractions are induced from a few trajectories and lack the structural diversity required for reliable matching. Following the first scheduled outer-loop update, the contradiction rate drops sharply from ∼14% to ∼10% within a single interval, whilst the hit rate accelerates. By step 300, the memory

reaches a 74% hit rate and a ∼6% contradiction rate, matching the aggregate statistics in Table 9. The crossing pattern illustrates how q_η-based pruning and outer-loop calibration transform an initially noisy set of candidate abstractions into a reliable procedural store.

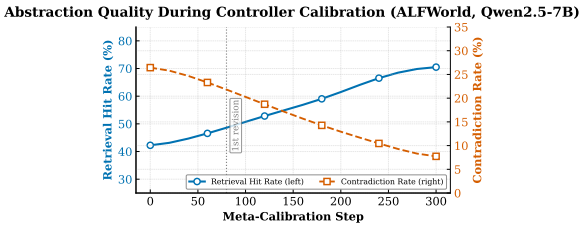

Figure 10: Abstraction quality during prior Controller calibration on ALFWORLD.

F Prompt

Perceptor. The Perceptor maps the interaction context into $\mathcal{A}_t = (\mathcal{V}_t, \mathcal{T}_t, \mathcal{C}_t, \mathcal{E}_t)$.

Perceptor System Prompt

Summarise the current interaction, keeping symbolic elements and natural language side by side.
GOAL: {task_goal}
TRAJECTORY: {trajectory_window}
OBSERVATION: {observation}
Return exactly four labelled fields.
$\mathcal{V}$: relevant entities, tools, roles, or states.
$\mathcal{T}$: observed transitions and candidate next transitions supported by the current goal, observation, and active constraints.
$\mathcal{C}$: preconditions, constraints, and actions to avoid.
$\mathcal{E}$: progress and outcome signals.
Formalise only the relevant predicates and variables, keeping the symbolic form minimal and combined with natural language. Include only information supported by the interaction.
Return: V:, T:, C:, E:.

Controller. The Controller reads $s_t^m = [c_t, \delta_t, \nu_t, \ell_t]$ and selects one meta-action.

Controller System Prompt

SYSTEM: Select one meta-action from the current control state.
CONTROL STATE: confidence c_t={conf}; contradiction ratio δ_t={delta}; novelty ν_t={novelty}; progress ℓ_t={progress}. All values lie in $[0, 1]$.
ACT: proceed under the current abstraction.
ABSTRACT: re-induce the abstraction over an extended trajectory window.
RETRIEVE: query memory for structurally compatible experience.
ROLLBACK: restore the last consistent internal abstraction and reasoning context.
Answer with one token: ACT, ABSTRACT, RETRIEVE, or ROLLBACK.

Binding (Prior Instantiation). The Binding module grounds retrieved abstractions in the current observation.

Binding System Prompt

Ground the retrieved abstractions in the current observation, keeping symbolic predicates and natural language together.
ACTIVE ABSTRACTION: {active_abstraction}
RETRIEVED ABSTRACTIONS: {retrieved_abstractions}
OUTCOME LABELS: {outcome_labels}
OBSERVATION: {observation}
Map compatible roles, states, and constraints onto entities in the observation, keeping symbolic predicates and natural language together. Retain reusable transitions from successful abstractions. Convert only verifier-attributed or rollback-attributed failing transitions into constraints on actions to avoid; do not treat every transition in an unsuccessful trajectory as a failure cause. Discard incompatible mappings. Return the grounded abstraction $\widetilde{\mathcal{A}}_t$, or NONE when no compatible mapping is available.

Binding validation. The Binding output is parsed against the same typed schema used by the Perceptor. Mappings that reference absent entities, violate active constraints, or fail schema validation are discarded. If no mapping survives validation, Bind returns NONE and the Actor receives only the active abstraction $\mathcal{A}_t$.

Actor (Policy Execution). The Actor generates the next environment action from the active and grounded abstractions.

Actor System Prompt

Generate one goal-directed environment action.
GOAL: {task_goal}
OBSERVATION: {observation}
ACTIVE ABSTRACTION: {active_abstraction}
GROUNDED ABSTRACTION: {grounded_abstraction}
Follow the active preconditions and constraints. Do not execute transitions marked as actions to avoid. Output exactly one environment action or tool call.

EvA on WebShop Task

1. Task and Product Evidence

Product image provided with the dataset record.

Search goal: Find a product matching the query *"grocery cookies"*.
Selected product: Walkers Shortbread Saltire Shortbread Rounds Tin, Scottish Cookies, Biscuits, Gift Box, 120 g.
Category: Grocery & Gourmet Food → Breads & Bakery → Cookies → Shortbread.
Price: $18.95.
Product identifier: B00EY177C0.
Available options: None. The item does not require a size, flavour, colour, or quantity-option selection.

2. Interaction and Task Resolution

Initial observation. The agent receives the product-search goal and observes the WebShop search interface.
Perceptor. The Perceptor identifies the central requirement as a product-category match: the target must be a grocery item classified as cookies. It also records that no explicit price limit or product option has been specified.
Controller decision. The Controller allows the Actor to search for the requested category. After a candidate is retrieved, it requires inspection of the product page before purchase, preventing selection based on the search-result title alone.
Actor trajectory:

1. `search[grocery cookies]`
2. Inspect the returned candidates.
3. `click[Walkers Shortbread Saltire Shortbread Rounds Tin]`
4. Verify that the product belongs to the cookies and shortbread categories, costs $18.95, and requires no option selection.
5. `click[Buy Now]`

Resolution. The selected item satisfies the reconstructed goal because it is a grocery product explicitly categorised as cookies and shortbread. No unresolved option or conflicting product attribute remains before purchase.

3. Abstraction Stored in Memory

Abstraction type: Success-derived product-selection procedure.
Goal pattern: Find and purchase a product that matches a requested commercial category.
Reusable procedure:

1. Formulate a search query from the requested category.
2. inspect the returned product candidates;
3. open the most relevant product page;
4. verify the title, category, price, and available options;
5. purchase only after all explicit requirements have been checked.

Constraints:

- The selected item must match the requested product category.
- Category membership must be verified on the product page.
- Any required product option must be selected before purchase.
- A product must not be purchased when an explicit requirement remains unverified.

Outcome: The product is selected successfully and the abstraction is stored with a SUCCESS provenance label.
Possible reuse. For a later request such as *"find grocery crackers"*, EvA can retrieve this abstraction, replace the target category and product candidate, and preserve the general search–inspect–verify–purchase procedure.

Table 10: EvA on a WebShop task and the abstraction it stores. The catalogue query serves as the goal; the stored memory keeps a reusable search–verify–purchase procedure without product-specific wording.