MEMSTEER: SELF-EVOLVING MEMORY FOR LONG-HORIZON VISION AND LANGUAGE REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-evolving natural-language memory lets frozen models draw on experience during long-horizon visual and textual reasoning, but its value depends on selecting useful guidance at each step. Given the same memory and the same tools, a frozen solver that keeps every policy in context or picks one for itself falls below tool use without any memory in most of our settings. We propose MemSteer, a framework that couples state-conditioned memory selection with outcome-validated memory evolution across high-resolution images, long documents, and interactive environments. A lightweight controller selects at every decision the natural-language policies that the frozen solver acts on, learning from groups of rollouts whose advantages are discounted by how widely identical choices already scatter rewards. On a slower timescale, failed trajectories are clustered, a language model writes one abstract policy per cluster, and the update is evaluated through a paired comparison on unseen tasks. Extensive experiments on nine benchmarks spanning long-horizon visual search, long-document reasoning, and interactive task execution with four frozen solvers show MemSteer improves average performance over strong tool-use and experience-based baselines.

1 INTRODUCTION

Long-horizon reasoning tasks require an agent to gather evidence over many steps before reaching a solution, such as locating a small sign in a high-resolution image (Wu & Xie, 2024), finding the decisive passage in a long document (Lee et al., 2024), or completing a multi-stage household task (Shridhar et al., 2021). Foundation models are increasingly deployed as tool-using agents (Yao et al., 2023), while many capable models are accessible only as frozen systems whose parameters cannot be updated. Self-evolving agents address this setting by adapting through external natural-language memory, accumulating reflections (Shinn et al., 2023), insights (Zhao et al., 2024), workflows (Wang et al., 2025b), or playbooks (Zhang et al., 2026b) for reuse on subsequent tasks.

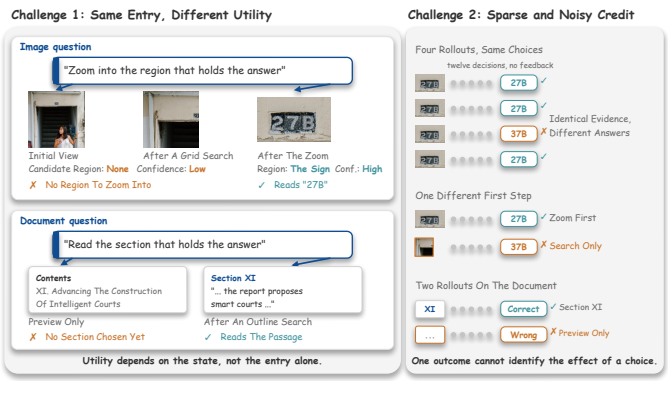


Figure 1: Illustration of the two challenges.

The first challenge (as shown in Figure 1, left) arises from the state-dependent utility of experience. The utility of an experience entry depends not only on its relevance to the task, but also on the evidence available at the current reasoning state. As the trajectory progresses, the task remains unchanged while the accumulated evidence can change substantially, making the same guidance useful at one step but ineffective or even detrimental at another. For example, an instruction to zoom into a candidate region becomes useful only after such a region has been localized, and applying it earlier introduces an unnecessary action without additional evidence. Existing self-evolving agents commonly retrieve memory according to task-level similarity (Zhao et al., 2024; Ouyang et al., 2026) or expose a larger

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.

memory directly to the frozen solver (Zhang et al., 2026b). These mechanisms capture task relevance, but do not explicitly estimate the utility of each entry under the evolving reasoning state. Consequently, the frozen solver must infer step-level applicability from context, without outcome-grounded evidence about which entries have been effective in comparable states. As a result, an otherwise valid experience entry may be invoked where its utility is low or negative.

The second challenge (as shown in Figure 1, right) arises from sparse and noisy outcome supervision. A long-horizon episode produces only a terminal success or failure after many intermediate decisions, and repeated executions show run-to-run variability. The outcome therefore entangles those decisions with that variability, and success or failure cannot be attributed to a single reasoning step or memory entry. Existing self-evolving methods construct or revise memory through language-model judgments over individual trajectories: votes over extracted insights (Zhao et al., 2024), edits to a playbook (Zhang et al., 2026b), rewritten cheatsheets (Suzgun et al., 2025), or self-assessed success labels (Ouyang et al., 2026). Such judgments can propose plausible diagnoses or updates, but cannot establish whether a change improves performance on future episodes. The same ambiguity applies to memory use: one episode cannot tell whether selecting an entry improved the outcome or merely coincided with it. Both memory selection and memory evolution therefore require repeated outcome-based comparisons that separate the effect of a choice from background variability.

To address these challenges, we propose MemSteer, a self-evolving framework that separates state-conditioned memory use from outcome-validated memory evolution while keeping the underlying solver frozen. At each reasoning step, a lightweight controller selects which experience entries should guide the solver's next action and learns these selections from episodic outcomes. Rather than relying solely on semantic relevance, the controller exploits outcome evidence accumulated from comparable reasoning states to estimate the utility of each entry. On a slower timescale, MemSteer abstracts recurring failures into candidate memory entries and tests updates on unseen tasks before retention. Both memory selection and memory evolution are therefore grounded in measured outcomes, while the frozen solver itself remains unchanged. Because the experience entries encode reusable evidence-gathering operations rather than modality-specific content, the same framework applies to high-resolution images, long documents, and interactive environments. Across nine benchmarks and four frozen solvers, MemSteer raises mean accuracy over ReAct-style tool use in the settings with complete test coverage for both methods. Our contributions are:

- We formulate self-evolving memory for frozen agents as two coupled decisions: state-conditioned memory selection and outcome-grounded memory evolution, and identify sparse, variable episodic feedback as the common source of credit-assignment uncertainty.
- We propose MemSteer, a self-evolving framework that jointly optimizes these two decisions while keeping the underlying solver frozen, and applies across high-resolution images, long documents, and interactive environments without modality-specific redesign.
- We introduce a set-scoring controller trained with noise-aware group-relative advantages to estimate step-level utility under outcome variability, and a gate that turns recurring failures into candidate entries and keeps only updates validated on unseen tasks.
- Experiments across nine benchmarks and four frozen solvers show that MemSteer improves average performance over tool-use and experience-based baselines in long-horizon visual search, long-document reasoning, and interactive tasks.

2 Related Work

Long-Horizon Reasoning. Large language models address multi-step problems by generating intermediate reasoning (Wei et al., 2022; Kojima et al., 2022), aggregating multiple reasoning trajectories (Wang et al., 2023), and interleaving reasoning with actions in an environment (Yao et al., 2023; Shridhar et al., 2021). Tasks with evidence distributed across a large input space require reasoning to alternate between evidence acquisition and inference. For high-resolution images, agents acquire evidence through guided visual search (Wu & Xie, 2024), tree-structured exploration (Shen et al., 2025), adaptive focus sequences (Zhang et al., 2025a), or reinforcement-learned search policies (Zheng et al., 2026), while TreeBench evaluates the traceability of the resulting evidence-gathering process (Wang et al., 2026). For long documents, agents retrieve relevant passages (Lewis et al., 2020), iteratively navigate compressed page representations (Lee et al., 2024), or traverse hierar-

chical summaries (Chen et al., 2023). Outcome-based learning of such sequential behaviors commonly employs group-relative or leave-one-out baselines (Shao et al., 2024; Ahmadian et al., 2024) together with conservative policy updates (Schulman et al., 2017; Peng et al., 2019). Existing approaches therefore provide increasingly capable mechanisms for multi-step evidence acquisition, but their action spaces and reasoning procedures are typically specialized to a particular modality or environment, limiting the direct reuse of such behaviors across heterogeneous evidence sources.

Self-Evolving Agents. Prior work commonly externalizes experience as natural-language memory, enabling adaptation without updating the underlying solver. Within an episode, verbal feedback can revise subsequent reasoning or behavior (Shinn et al., 2023; Madaan et al., 2023). Across episodes, persistent experience has been encoded as insights (Zhao et al., 2024), workflows (Wang et al., 2025b), skill libraries (Wang et al., 2024), cheatsheets (Suzgun et al., 2025), evolving playbooks (Zhang et al., 2026b), reasoning memories (Ouyang et al., 2026), or procedural memories (Fang et al., 2026), together with systems for organizing and retrieving long-term context (Packer et al., 2023; Park et al., 2023; Chhikara et al., 2025; Xu et al., 2025). Related work on frozen-model adaptation further optimizes prompts and programs (Yang et al., 2024; Khattab et al., 2024; Yuksekgonul et al., 2024), model routing (Ong et al., 2025), reflection generation (Yao et al., 2024), and retrieval of prior cases (Zhou et al., 2025). Most closely related, MemSkill learns to select from an evolving collection of memory skills that govern the extraction, consolidation, and pruning of interaction-derived memory (Zhang et al., 2026a). MemSteer instead learns which experience entries should guide the frozen solver at each reasoning state, while validating memory updates through paired outcome comparisons on unseen tasks. This couples policy selection with validated memory updates for a fixed solver (as shown in Table 1).

Table 1: Comparison with agent memory systems.

Method	Train-free	Revise	Learned	Per-step	Unseen test
ExpeL	✓	✓	×	×	×
AWM	✓	×	×	×	×
ACE	✓	✓	×	×	×
ReasoningBank	✓	×	×	×	×
MemSkill	×	✓	✓	×	×
MemSteer	×	✓	✓	✓	✓

3 METHODOLOGY

Problem Formulation. We consider tasks $x \sim \mathcal{D}$ solved by a frozen tool-using solver F through interaction with an environment. At decision step t, the state s_t summarizes the task instruction, the evidence accumulated so far, and the interaction history. The agent maintains an external natural-language Policy Memory $\mathcal{M} = \{m_1, \ldots, m_{|\mathcal{M}|}\}$ together with a controller π_θ. The controller selects either a subset of memory entries $A_t \subseteq \mathcal{M}$ or STOP. For a continuing decision, the solver conditions on s_t and A_t to produce a tool action u_t, after which the environment transitions to the next state:

$$A_t \sim \pi_\theta(\cdot \mid s_t, \mathcal{M}), \qquad u_t \sim p_F(\cdot \mid s_t, A_t), \qquad s_{t+1} \sim P(\cdot \mid s_t, u_t). \tag{1}$$

An episode produces a trajectory τ of length $H(\tau) \leq T$, where T denotes the decision budget. At termination, the task yields a binary outcome $Y(\tau) \in \{0, 1\}$. The learning objective is

$$\max_{\theta,\ \mathcal{M} \in \mathcal{L}} J(\theta, \mathcal{M}) = \mathbb{E}_{x \sim \mathcal{D}} \mathbb{E}_{\tau \sim p(\tau \mid x, \theta, \mathcal{M})} \big[Y(\tau) \big], \tag{2}$$

where F remains fixed and $\mathcal{L}$ denotes the space of natural-language memories. The two optimization variables capture complementary decisions: θ determines how the current memory is used at each reasoning state, whereas $\mathcal{M}$ determines the experience available to subsequent episodes.

Overview. Figure 2 illustrates the workflow of MemSteer, which consists of policy-guided inference, controller optimization, and policy evolution. A lightweight controller selects state-conditioned memory guidance for the frozen solver, learns these selections from grouped rollouts with noise-aware outcome credit, and alternates with memory updates that are accepted only after validation on unseen tasks. The solver remains fixed throughout.

Policy-Guided Inference. Each memory entry m pairs four natural-language fields, namely applicability, guidance, success condition, and recovery, with a routing family $\phi(m)$. The families SEARCH, FOCUS, and ENHANCE explore unobserved evidence, inspect a selected region in detail, and refine evidence already under consideration. Families are defined by their effect on evidence

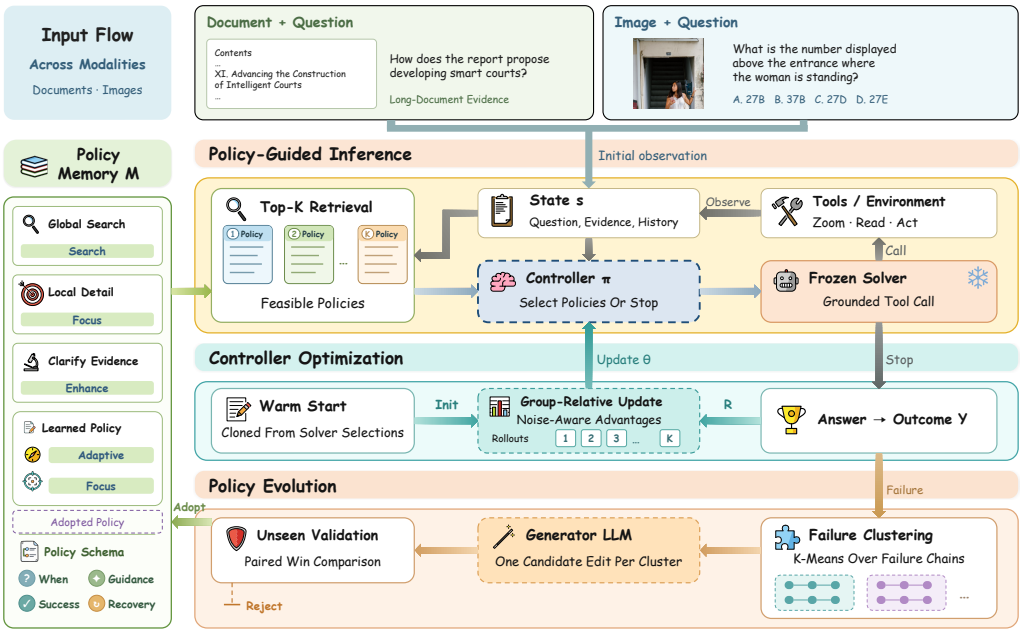


Figure 2: Overview of MemSteer, which selects guidance for a frozen solver, learns from grouped rollouts, and validates memory updates on unseen tasks.

rather than by modality-specific tool identity, and each environment maps them to its available tools. The ADAPTIVE family represents composite multi-step guidance that may invoke tools across families, whereas STOP is a separate termination decision. The seed memory $\mathcal{M}_0$ contains one entry for each routing family. After each tool call, the solver assesses the accumulated evidence, what remains unresolved, and its confidence. The state s_t combines this assessment with the task instruction and the action history, represented by a text embedding $e(s_t)$ and progress features z_t (Appendix A.2). The k entries most similar to $e(s_t)$ whose routing family has an available tool form the candidate set $\mathcal{C}_t$, and the controller scores each candidate conditioned on the current state:

$$\ell_m = f_\theta\big([h_s,\, h_m,\, h_s \odot h_m,\, |h_s - h_m|]\big), \quad \text{where} \quad \begin{aligned} h_s &= \mathrm{GELU}\big(W_s[\mathrm{LN}(e(s_t)), z_t]\big), \\ h_m &= \mathrm{GELU}\big(W_m\,\mathrm{LN}(e(m))\big). \end{aligned} \tag{3}$$

A separate head produces the termination logit $\ell_{\text{STOP}} = g_\theta([h_s, z_t])$. Because every entry is scored through the same embedding-based function, entries added later need no architectural change, and the controller has 0.86M trainable parameters. With $p_{\text{STOP}} = \sigma(\ell_{\text{STOP}})$ and $p_m = \sigma(\ell_m)$, the controller samples a Bernoulli termination decision, then independently samples candidate entries with probabilities p_m when continuing. If more than n entries are selected, it retains those with the highest probabilities, and if none is selected, it keeps the highest-scoring candidate. The selected entries guide one environment action of the frozen solver.

Noise-Aware Controller Optimization. We initialize the controller by behavior cloning (Pomerleau, 1991) memory selections elicited from the frozen solver on training states, which transfers the solver's prior over memory use to π_θ without a language-model selection at every later decision. Controller learning uses the regularized return

$$\widetilde{R}(\tau) = Y(\tau) - \lambda_c\, C(\tau), \tag{4}$$

where $C(\tau)$ is the mean number of entries selected over continuing decisions, so that unnecessary guidance is discouraged. The task objective and all reported evaluations depend on $Y(\tau)$ alone. Each round samples up to G rollouts per task, and return variation within a group reflects both controller selections and solver variability. We estimate the latter from the *signature* of a rollout, the sequence of subsets it selected. Let v_{noise} and v_{mix} be the mean within-group return variance among groups with a single signature and among groups with several. The noise attenuation factor is

$$\alpha_{\text{noise}} = \max\left(0,\, 1 - \frac{v_{\text{noise}}}{v_{\text{mix}} + \epsilon}\right), \tag{5}$$

where $\epsilon > 0$ ensures numerical stability. Single-signature groups are used only to estimate v_{noise} and contribute no policy gradient. For rollout i in a multi-signature group g, we compute the leave-one-out advantage (Ahmadian et al., 2024)

$$A_i = \alpha_{\text{noise}} \left(\widetilde{R}_i - \frac{1}{|g| - 1} \sum_{j \in g \setminus \{i\}} \widetilde{R}_j \right), \tag{6}$$

shared across the decisions of that rollout. For decision t in rollout i, let $\widehat{A}_t = A_i$ denote the rollout advantage assigned to that decision. Larger dispersion under identical signatures raises v_{noise} and reduces the credit assigned to differences in controller selections. Inspired by advantage-weighted regression (Peng et al., 2019), each decision receives $w_t = \min(\exp(\widehat{A}_t/\beta), w_{\max})$, with weights normalized to unit mean over decisions with nonzero advantage. The controller minimizes

$$\mathcal{L}(\theta) = -\mathbb{E}_t\big[(w_t - 1) \log \pi_\theta(A_t \mid s_t, \mathcal{M})\big] - c_H\, \mathbb{E}_t\big[\mathcal{H}(\pi_\theta(\cdot \mid s_t, \mathcal{M}))\big] + c_{\text{KL}}\, \mathbb{E}_t\big[\rho_t - 1 - \log \rho_t\big], \tag{7}$$

where $\rho_t = \pi_\theta(A_t \mid s_t, \mathcal{M})/\pi_{\text{old}}(A_t \mid s_t, \mathcal{M})$. Centering the normalized weight at one removes behavior cloning, which would reinforce sampled decisions regardless of their relative returns.

Outcome-Validated Policy Evolution. Memory edits are proposed only from *action-sensitive* tasks whose rollout groups vary in both selection signature and outcome, reducing the risk of attributing solver-induced variation to memory choices. Failed trajectories from these tasks are recorded as failure chains of state assessments, selected entries, and tool actions, and are subsequently embedded and clustered with K-means (Lloyd, 1982). For each cluster, the generator $\mathcal{G}$ proposes a single memory edit (an addition, a revision, or a deletion) conditioned on the clustered failures, the current memory, and outcome statistics of existing entries. Edits containing sample-specific content are discarded. The resulting candidate memory $\mathcal{M}'$ is evaluated against the current memory $\mathcal{M}$ on an unseen validation split using identical fixed controller parameters, thereby separating the effect of the memory edit from subsequent controller adaptation. Let b and c denote the numbers of validation tasks solved exclusively by $\mathcal{M}'$ and $\mathcal{M}$, respectively. In runs using the strict validation rule, the memory update can be formulated as

$$\mathcal{M}_r = \begin{cases} \mathcal{M}', & b > c, \text{ or } b = c \text{ and } |\mathcal{M}'| < |\mathcal{M}|, \\ \mathcal{M}, & \text{otherwise.} \end{cases} \tag{8}$$

The strict rule retains a candidate with a positive paired win–loss balance, with exact ties favoring the smaller memory. The validation comparisons in Appendix D.1 also include a one-sided harm screen that permits candidates with small or statistically uncertain losses. Training proceeds for R rounds, with policy evolution applied after every round except the last, and the controller checkpoint most accurate on the validation split is paired with the most recently accepted memory. The complete training procedure is summarized in Algorithm 1.

4 Experiments

Benchmarks and Metrics. We evaluate on nine benchmarks spanning long-horizon visual search, long-document reasoning, and interactive task execution. The image benchmarks are TreeBench (TreeB.) (Wang et al., 2026), HR-Bench 4K (HR-4K) (Wang et al., 2025a), MME-RealWorld-Lite (MME-RW) (Zhang et al., 2025c), and V*Bench (Wu & Xie, 2024). The document benchmarks are LongBench v2 (LongB.) (Bai et al., 2025b), ∞Bench En.MC (∞B.) (Zhang et al., 2024), QuALITY-hard (QuAL.) (Pang et al., 2022), and AmongUs-X (Among.) (Among-us-X, 2026), where the solver identifies the impostor of an Among Us game between language-model agents (Golechha & Garriga-Alonso, 2025) from its public record. The third environment is ALFWorld (Shridhar et al., 2021), with 134 unseen games across six task types. We report test accuracy, or success rate on ALFWorld. Runs with API failures use evaluated episodes as the denominator. Paired item-level comparisons use exact McNemar tests (McNemar, 1947). Details in Appendix B.1.

Baselines. Across settings, we compare with 14 baseline methods. Direct answer, zero-shot CoT (Kojima et al., 2022), and CoT-SC (Wang et al., 2023) answer without tools. On documents, full-document reading, RAG (Lewis et al., 2020), and ReadAgent (Lee et al., 2024) provide reference

results under different access conditions. ReAct-style tool use (Yao et al., 2023) gives the solver tools without memory. ACE (Zhang et al., 2026b), ExpeL (Zhao et al., 2024), AWM (Wang et al., 2025b), and MemSkill (Zhang et al., 2026a) build experience-based guidance. Three seed-memory controls use $\mathcal{M}_0$ without learned selection: all policies in context, random selection, and solver self-selection. ReAct-style tool use and the seed-memory controls share MemSteer's tools, episode limits, and answer prompt apart from the additional document search and budget specified below. ExpeL, AWM, and MemSkill act with ReAct-style tools. Details are in Appendix B.2.

Implementation Details. The frozen solvers are Qwen3-VL-32B-Instruct (Bai et al., 2025a) and GLM-4.6V (V Team et al., 2025) for image tasks and Qwen3.5-27B (Team, 2026) and Mistral-Large-2512 (Liu et al., 2026) for document tasks. Mistral-Large-2512 is also used for ALFWorld. Solver decoding uses temperature 0 except for CoT-SC, which votes over three samples drawn at temperature 0.7. The generator is the frozen solver itself, except with Qwen3.5-27B, whose policies are written by DeepSeek-V4-Pro (DeepSeek-AI et al., 2026). States and policies are embedded with Qwen3-Embedding-0.6B (Zhang et al., 2025b), and the controller has 256 hidden units trained with Adam (Kingma & Ba, 2014). We use $k = 8$, $n = 3$, $\lambda_c = 0.01$, $T = 12$ by default (50 on ALFWorld), $R = 3$, $\beta = 0.5$, $w_{\max} = 10$, $c_H = 0.02$, and $c_{\mathrm{KL}} = 0.1$. Training uses fixed groups of four rollouts or the $G = 2 \rightarrow 6$ configuration, depending on the solver (Table 6). Each environment maps the routing families to its own tools. More details are in Appendix A.

4.1 Main Results

Long-Horizon Visual Search Tasks. Table 2 shows that MemSteer achieves the highest average accuracy in the main comparison with both frozen solvers, reaching 70.2% with Qwen3-VL and 68.3% with GLM-4.6V, respectively 1.0 and 2.0 points above the strongest self-evolving baselines. Across the eight solver–benchmark combinations, MemSteer ranks first or second in five, including strong HR-Bench 4K accuracy with both solvers and the best V*Bench result with GLM-4.6V. Relative to ReAct-style tool use, the clearest consistent gains occur on MME-RW, with improvements of 1.9 and 2.7 points for Qwen3-VL and GLM-4.6V, respectively. Near-parity on TreeBench and V*Bench is consistent with greater value from selective evidence refinement on MME-RW.

Table 2: Image benchmark accuracy (%). **Bold** and underline mark the best and second-best values.

	Qwen3-VL-32B-Instruct					GLM-4.6V				
Method	TreeB.	HR-4K	MME-RW	V*	Avg.	TreeB.	HR-4K	MME-RW	V*	Avg.
Direct answer	49.5	80.2	56.2	84.3	67.5	46.3	72.8	46.0	71.9	59.3
Zero-shot CoT	**54.0**	82.0	53.2	90.9	70.1	49.1	80.9	**54.8**	84.3	67.3
CoT-SC	51.6	81.9	55.8	89.3	69.7	49.1	81.4	54.5	83.5	67.1
ReAct-style tools	51.9	80.2	54.2	89.3	68.9	**51.9**	82.7	48.5	86.8	67.5
ACE	52.3	74.8	57.8	83.5	67.1	47.7	70.9	50.5	79.3	62.1
ExpeL	50.1	80.1	54.9	85.9	67.8	49.1	82.0	46.7	83.4	65.3
AWM	50.5	81.3	**58.1**	86.7	69.2	51.2	82.3	46.6	85.1	66.3
MemSkill	51.6	81.2	55.2	88.4	69.1	50.2	81.2	48.8	85.1	66.3
All policies	51.2	82.2	55.0	86.8	68.8	**51.9**	82.6	47.5	86.0	67.0
Random selection	50.2	80.1	51.9	**91.7**	68.5	47.7	79.1	47.9	82.6	64.3
Self-selection	50.9	81.6	53.5	86.0	68.0	48.9	80.0	46.0	81.0	64.0
MemSteer (ours)	52.3	**82.5**	56.1	90.0	**70.2**	51.6	**82.8**	51.2	**87.6**	**68.3**

Long-Document Reasoning Tasks. Table 3 shows that our proposed MemSteer achieves the highest average accuracy among tool-mediated methods with both frozen solvers, reaching 71.6% with Qwen3.5-27B and 58.1% with Mistral-Large-2512, respectively 9.8 and 3.1 points above the strongest self-evolving baselines. For Qwen3.5-27B, the largest gains over ReAct-style tool use occur on ∞Bench (+29.8) and Among (+16.0), while Mistral-Large-2512 improves on the three benchmarks with complete ReAct-style comparisons by 6.3–7.3 points. The larger Qwen gains coincide with additional document-search access on three benchmarks; on QuALITY, where tool access is matched, MemSteer still improves by 4.7 points. This persistence under matched tools indicates that the benefit is not explained by tool availability alone and is consistent with more effective state-conditioned evidence selection.

Interactive Task Execution Tasks. Table 4 shows that MemSteer achieves the highest overall success rate on 134 unseen ALFWorld games with Mistral-Large-2512, reaching 8.2%, 0.7 points above

Table 3: Document benchmark accuracy (%). Grey rows show methods with different document-access protocols and are excluded from ranking.

	Qwen3.5-27B					Mistral-Large-2512				
Method	LongB.	∞B.	Among.	QuAL.	Avg.	LongB.	∞B.	Among.	QuAL.	Avg.
Direct answer	33.1	49.1	38.7	39.7	40.1	28.9	41.5	60.7	39.0	42.5
Zero-shot CoT	42.7	50.9	53.0	46.3	48.2	31.9	60.4	31.0	45.0	42.1
CoT-SC	44.4	49.7	50.7	46.0	47.7	**40.8**	**64.8**	50.7	43.7	50.0
ACE	38.8	48.4	62.3	42.3	48.0	33.8	49.1	63.3	46.7	48.2
ReAct-style tools	38.1	51.6	63.0	69.0	55.4	32.7	52.2	67.0	55.3	51.8
ExpeL	43.1	53.4	75.0	70.6	60.5	35.0	54.7	70.0	58.0	54.4
AWM	41.7	57.2	73.0	66.7	59.7	33.1	56.6	69.0	56.3	53.8
MemSkill	43.5	59.1	74.7	70.0	61.8	33.8	56.0	71.0	59.0	55.0
All policies	33.5	52.2	58.7	67.0	52.8	30.4	54.7	71.0	57.1	53.3
Random selection	33.6	47.8	51.3	51.3	46.0	30.1	40.7	64.1	46.2	45.3
Self-selection	37.7	51.6	59.7	66.3	53.8	31.5	50.0	70.4	60.6	53.1
Full document	49.3	90.3	52.7	80.3	68.2	39.4	82.4	72.0	75.3	67.3
RAG (top-8)	49.2	81.8	66.3	83.0	70.1	37.3	79.3	36.0	75.3	57.0
ReadAgent	52.6	92.9	64.3	83.7	73.4	47.1	86.2	56.3	70.0	64.9
MemSteer (ours)	**52.5**	**81.4**	**79.0**	**73.7**	**71.6**	37.3	58.5	**74.3**	**62.3**	**58.1**

Table 4: ALFWorld success rate (%) on 134 unseen games with Mistral-Large-2512.

Method	Pick (24)	Examine (18)	Clean (31)	Heat (23)	Cool (21)	Pick 2 (17)	Overall
ReAct-style tools	**12.5**	0.0	**9.7**	**8.7**	4.8	**5.9**	7.5
All policies	8.3	0.0	6.5	4.3	4.8	0.0	4.5
Random selection	0.0	0.0	3.2	0.0	4.8	0.0	1.5
Self-selection	4.2	0.0	6.5	4.3	4.8	0.0	3.7
ExpeL	8.3	**5.6**	6.5	**8.7**	4.8	**5.9**	6.7
AWM	8.3	0.0	6.5	4.3	4.8	**5.9**	5.2
MemSkill	8.3	**5.6**	3.2	4.3	**9.5**	**5.9**	6.0
MemSteer (ours)	**12.5**	**5.6**	6.5	**8.7**	**9.5**	**5.9**	**8.2**

ReAct-style tool use and 1.5 points above the strongest self-evolving baseline. The improvement is concentrated on Examine and Cool, where MemSteer gains 5.6 and 4.7 points over ReAct-style tools, while matching it on Pick, Heat, and Pick-2 and remaining lower on Clean. This pattern indicates that state-conditioned policy selection can improve particular long-horizon interaction sequences without uniformly improving environment control.

4.2 Framework Analysis

State-Conditioned Policy Selection. Figure 3 shows that operation utility varies substantially with the reasoning state. On image tasks, the within-task advantage of stopping shifts from -12.6 points at low solver confidence to $+5.4$ points at high confidence, while evidence-acquisition operations exhibit distinct confidence-dependent trends; similar variation appears on document tasks. The learned selections reflect these differences: on image tasks, stopping accounts for only 16–19% of decisions at low confidence but rises to 66–72% at high confidence. These show that a fixed operation preference is insufficient and support state-conditioned policy selection across decisions.

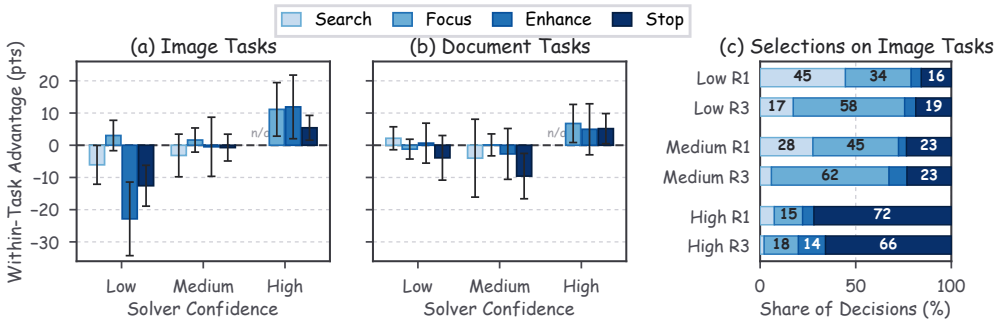


Figure 3: Operation value by confidence (12 runs, 95% intervals, n/a for fewer than 30 cases).

Table 5: Component ablations. Δ is relative to the full system's average accuracy.

Variant	Mem.	Evol.	Sel.	Train.	Test accuracy (%)					
Images, Qwen3-VL-32B					TreeB.	HR-4K	MME-RW	V*	Avg.	Δ (%)
MemSteer (full)	✓	✓	✓	✓	**52.3**	**82.5**	56.1	**90.0**	**70.2**	–
w/o controller training	✓	✓	✓	×	50.7	79.4	**56.3**	86.6	68.2	−2.8
w/o selector, solver selects	✓	×	×	×	50.9	81.6	53.5	86.0	68.0	−3.1
w/o selector, all in context	✓	×	×	×	51.2	82.2	55.0	86.8	68.8	−2.0
w/o memory	×	×	×	×	51.9	80.2	54.2	89.3	68.9	−1.9
Documents, Qwen3.5-27B					LongB.	∞B.	Among.	QuAL.	Avg.	Δ (%)
MemSteer (full)	✓	✓	✓	✓	**52.5**	**81.4**	**79.0**	**73.7**	**71.6**	–
w/o controller training	✓	✓	✓	×	44.8	**81.4**	**79.0**	69.7	68.7	−4.1
w/o selector, solver selects	✓	×	×	×	37.7	51.6	59.7	66.3	53.8	−24.9
w/o selector, all in context	✓	×	×	×	33.5	52.2	58.7	67.0	52.8	−26.3
w/o memory	×	×	×	×	38.1	51.6	63.0	69.0	55.4	−22.6

Ablation Studies. Table 5 shows that policy memory is most useful with learned selection. On images, the full system averages 70.2% versus 68.0 to 68.9% for the ablations, while Figure 4 shows that seed memory without learned selection often trails tool use alone. On documents, the warm start trails the full system mainly on LongBench and QuALITY, while the larger gaps without selection also reflect extra search access. The image results with matched tools are consistent with the benefit of selecting guidance for the current reasoning state.

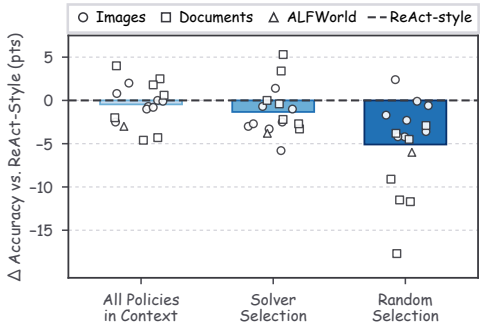


Figure 4: Seed-memory variants across 16 reported benchmark settings.

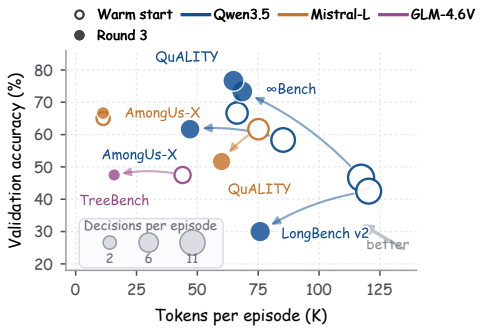


Figure 5: Controller training across seven settings without document search.

Efficient and Extensible Tool Use. Figure 5 shows that controller training cuts tokens per episode by 34% across seven settings while mean validation accuracy rises. On HR-Bench 4K, fewer tool calls coincide with higher test accuracy (as shown in Figure 6), consistent with learning when further inspection is unnecessary. With document search, accuracy rises and token use falls on all three tested benchmarks (as shown in Table 7). Together, these results support state-conditioned decisions about when to gather more evidence and when to stop, even as the available tools expand.

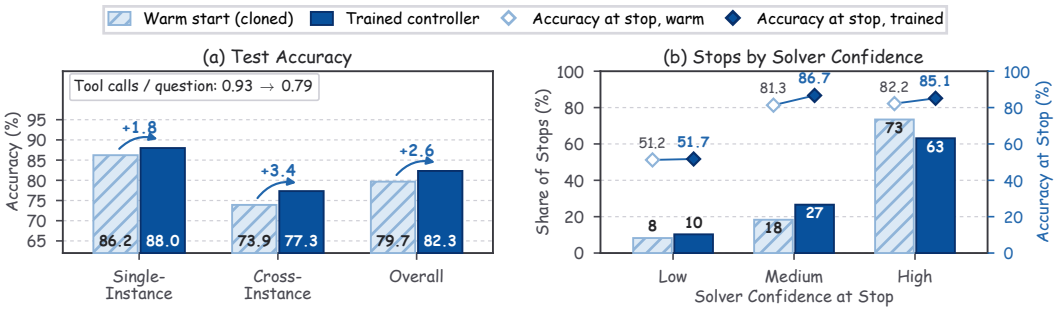


Figure 6: Controller training on 605 HR-Bench 4K tests (Qwen3-VL-32B). (a) Accuracy and tool calls. (b) Stop share and accuracy by confidence.

Outcome-Grounded Optimization. Table 6 shows that training groups with different choices and outcomes range from 17.0% to 80.9% across solvers. The sampling schemes also differ, so this

Table 6: Rollout outcomes by solver. Mistral-L uses fixed four except on AmongUs-X ($2 \rightarrow 6$).

Solver	Rollouts	Contrastive	Outcome	α_{noise}	Cand.	b<c
Qwen3-VL	G=4	17.0	1.4	0.80 / 0.47	8	4
GLM-4.6V	$2 \rightarrow 6$	50.9	0.3	0.96 / 0.81	7	1
Mistral-L	$4 / 2 \rightarrow 6$	36.6	2.5	0.80 / 0.00	8	3
Qwen3.5	$2 \rightarrow 6$	80.9	0.0	1.00 / 1.00	8	4

Table 7: Document search with Qwen3.5-27B.

	Acc. (%)		Tokens / ep.	
Benchmark	w/o	with	w/o	with
LongBench	30.8	**52.5**	117K	**78K**
∞Bench	54.7	**81.4**	64K	**30K**
AmongUs-X	60.7	**79.0**	59K	**35K**

range does not isolate a sampling effect. Twelve of 31 candidate memory updates lower validation accuracy. These results support outcome comparisons for both policy selection and memory updates. Further gate records and adopted entries are in Appendices D.1 and D.6.

Hyperparameter Sensitivity. Table 8 shows that the default configuration achieves the best or tied-best reported accuracy across all four hyperparameter sweeps for both solver–benchmark pairs. Qwen3.5-27B on AmongUs-X is more sensitive, with accuracy dropping from 79.0% to 73.0% when the policy cost is removed, whereas GLM-4.6V on TreeBench varies within a narrower 48.8–51.6% range. The default remains optimal or tied across candidate size, selection budget, rollout count, and policy cost, indicating a stable operating point despite task-dependent sensitivity.

Table 8: Hyperparameter sensitivity in test accuracy (%). Defaults are shaded.

	Candidates k			Policies / decision n			Rollouts / task G			Policy cost λ_c		
Value	4	8	12	1	2	3	2	4	6	0	0.01	0.05
Qwen3.5 / AmongUs-X	76.0	79.0	73.7	75.0	76.3	79.0	75.7	75.7	79.0	73.0	79.0	74.0
GLM-4.6V / TreeBench	50.5	51.6	48.8	51.2	49.8	51.6	48.8	49.1	51.6	50.2	51.6	51.6

4.3 Case Studies

Figure 7 presents a visual and a document case. In the HR-Bench 4K example, the warm start uses a grid and then six zooms, progressively moving away from the road sign. Its final crop contains only sky, and it answers incorrectly. The trained controller zooms directly to the sign, reads the destination and its arrow, then stops and answers correctly after two decisions. The contrast shows a difference in the evidence acquired rather than a benefit from making more tool calls. On LongBench v2, failures from repeated search and premature stopping yield policies that the controller later selects on related tasks. Further visual trajectories are in Appendix D.4.

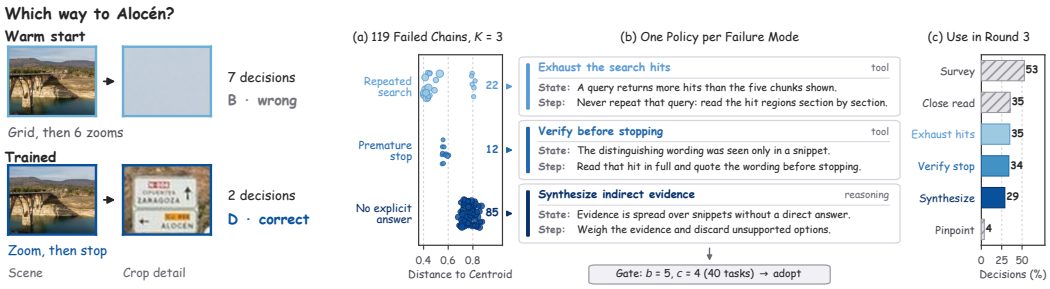


Figure 7: Case studies on HR-Bench 4K (left) and LongBench v2 (right). In the document case, marker size shows chain length and grey hatching marks seed policies.

5 Conclusion

This paper proposes MemSteer, which learns when a frozen solver should use natural-language policies and validates memory updates on unseen tasks. At each decision, a lightweight controller selects policies for the current reasoning state and learns from outcome comparisons across rollouts. On a slower timescale, the framework abstracts recurring failures into candidate policies and retains updates that pass paired validation. Experiments on nine image, document, and interactive benchmarks with four solvers show stronger average performance than tool-use and experience-based baselines.

AI USE STATEMENT

We used generative AI tools to assist with code edits, manuscript polishing, figure preparation, and mathematical notation checks. We reviewed the AI-assisted work. These tools did not independently determine the methods or technical conclusions. We take responsibility for the final content, including AI-assisted text and figures.

REPRODUCIBILITY STATEMENT

The training algorithm, implementation details, and seed policies are described in Appendices A.1 to A.3, while Appendix A.5 lists the hyperparameters. Benchmark splits, preprocessing, metrics, and baseline adaptations are in Appendices B.1 and B.2, and the full prompts appear in Appendix C. Statistical procedures are specified alongside the benchmark metrics in Appendix B.1 and the memory-update validation records in Appendix D.1. Hyperparameter sensitivity results are reported in Appendix D.2.

REFERENCES

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to Basics: Revisiting REINFORCE Style Optimization for Learning from Human Feedback in LLMs. *arXiv preprint arXiv:2402.14740*, 2024.

Among-us-X. AmongUs-X. Hugging Face dataset, `https://huggingface.co/datasets/Among-us-X/AmongUs-X`, 2026.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL Technical Report. *arXiv preprint arXiv:2511.21631*, 2025a.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench v2: Towards Deeper Understanding and Reasoning on Realistic Long-context Multitasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3639–3664. Association for Computational Linguistics, 2025b.

Howard Chen, Ramakanth Pasunuru, Jason Weston, and Asli Celikyilmaz. Walking Down the Memory Maze: Beyond Context Limit through Interactive Reading. *arXiv preprint arXiv:2310.05029*, 2023.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building Production-Ready AI Agents with Scalable Long-Term Memory. In *ECAI 2025: 28th European Conference on Artificial Intelligence*, volume 413 of *Frontiers in Artificial Intelligence and Applications*, pp. 2993–3000. IOS Press, 2025.

David L. Davies and Donald W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, 1979.

DeepSeek-AI, Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chengyu Hou, Chenhao Xu, Chenze Shao, Chong Ruan, Conner Sun, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Donghao Li, Dongjie Ji, Erhang Li, Fang Wei, Fangyun Lin, Fangzhou Yuan, Feiyu Xia, Fucong Dai, Guangbo Hao, Guanting Chen, Guoai Cao, Guolai Meng, Guowei Li, Han Yu, Han Zhang, Hanwei Xu, Hao Li, Haofen Liang, Haoling Zhang, Haoming Luo, Haoran Wei, Haotian Yuan, Haowei Zhang, Haowen Luo, Haoyu Chen, Haozhe Ji, Hengqing Zhang, Honghui Ding, Hongxuan Tang, Huanqi Cao, Huazuo Gao, Hui Qu, Hui Zeng, J Yang, JQ Zhu, Jia Luo, Jia Song, Jia Yu, Jialiang Huang, Jialu Cai, Jian Liang, Jiangting Zhou, Jiasheng Ye, Jiashi Li, Jiaxin Xu, Jiewen Hu, Jieyu Yang, Jin Chen, Jin Yan, Jingchang Chen, Jingli Zhou, Jingting Xiang, Jingyang Yuan, Jingyuan Cheng, Jingzi Zhou, Jinhua Zhu, Jiping Yu, Joseph Sun, Jun Ran, Junguang Jiang, Junjie Qiu, Junlong Li, Junmin Zheng, Junxiao Song, Kai Dong, Kaige Gao, Kang Guan, Kexing Zhou, Kezhao Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Wang, Leyi Xia, Li Zhang, Liang Zhao, Lihua Guo, Lingxiao Luo, Linwang

Ma, Linyan Zhu, Litong Wang, Liyu Cai, Liyue Zhang, Longhao Chen, MS Di, MY Xu, Max Mei, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Mingxu Zhou, Minmin Han, Ning Wang, Panpan Huang, Panpan Wang, Peixin Cong, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qingyang Li, Qinyu Chen, Qiushi Du, Qiwei Jiang, Rui Tian, Ruifan Xu, Ruijie Lu, Ruiling Xu, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runqian Chen, Runqiu Yin, Runxin Xu, Ruomeng Shen, Ruoyu Zhang, Ruyi Chen, SH Liu, Shanghao Lu, Shangmian Sun, Shangyan Zhou, Shanhuang Chen, Shaofei Cai, Shaoheng Nie, Shaoqing Wu, Shaoyuan Chen, Shengding Hu, Shengyu Liu, Shiqiang Hu, Shirong Ma, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, Shuying Yu, Songyang Zhou, Tao Ni, Tao Yun, Tian Jin, Tian Pei, Tian Ye, Tianle Lin, Tianran Ji, Tianyi Cui, Tianyuan Yue, Tingting Yu, Tun Wang, W Zhang, WL Xiao, Wangding Zeng, Wei An, Weilin Zhao, Wen Liu, Wenfeng Liang, Wenjie Pang, Wenjing Luo, Wenjing Yao, Wenjun Gao, Wenkai Yang, Wenlve Huang, Wenqing Hou, Wentao Zhang, Wenting Ma, Xi Gao, Xiang He, Xiangwen Wang, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaokang Zhang, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingchen Liu, Xingkai Yu, Xingyou Li, Xinyu Yang, Xinyu Zhang, Xu Chen, Xuanyu Wang, Xuecheng Su, Xueyin Chen, Xuheng Lin, Xuwei Fu, YC Yan, YQ Wang, YW Ma, Yanfeng Luo, Yang Zhang, Yanhong Xu, Yanru Ma, Yanwen Huang, Yao Li, Yao Li, Yao Xu, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Qian, Yi Shao, Yi Yu, Yichao Zhang, Yifan Ding, Yifan Shi, Yijia Wu, Yiliang Xiong, Yiling Ma, Ying He, Ying Tang, Ying Zhou, Yingjia Luo, Yinmin Zhong, Yishi Piao, Yisong Wang, Yixiang Zhang, Yixiao Chen, Yixuan Tan, Yixuan Wei, Yiyang Ma, Yiyuan Liu, Yonglun Yang, Yongqiang Guo, Yongtong Wu, Yu Wu, YuKun Li, Yuan Cheng, Yuan Ou, Yuanfan Xu, Yuanhao Li, Yuduan Wang, Yuehan Yang, Yuer Xu, Yuhan Wu, Yuhao Meng, Yuheng Zou, Yukun Zha, Yunfan Xiong, Yupeng Chen, Yuping Lin, Yuqian Cao, Yuqian Wang, Yushun Zhang, Yuting Yan, Yutong Lin, Yuxian Gu, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuxuan Zhou, Yuyang Zhou, Yuzhen Huang, ZF Wu, Zehao Wang, Zehua Zhao, Zehui Ren, Zekai Zhang, Zhangli Sha, Zhe Fu, Zhe Ju, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zheren Gao, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhixian Huang, Zhixuan Chen, Zhiyu Wu, Zhizhou Ren, Zhongyu Wu, Zhuoshu Li, Zhuping Zhang, Zian Xu, Zihao Wang, Zihua Qu, Zihui Gu, Zijia Zhu, Zilin Li, Zipeng Zhang, Ziwei Xie, Ziyi Gao, Ziyi Wan, Zizheng Pan, and Zongqing Yao. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026. URL https://arxiv.org/abs/2606.19348.

Runnan Fang, Yuan Liang, Xiaobin Wang, Jialong Wu, Shuofei Qiao, Pengjun Xie, Fei Huang, Huajun Chen, and Ningyu Zhang. Memp: Exploring Agent Procedural Memory. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 17490–17502. Association for Computational Linguistics, 2026.

Satvik Golechha and Adrià Garriga-Alonso. Among Us: A Sandbox for Measuring and Detecting Agentic Deception. In *Advances in Neural Information Processing Systems*, 2025.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, et al. DSPy: Compiling Declarative Language Model Calls into Self-Improving Pipelines. In *International Conference on Learning Representations*, 2024.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large Language Models are Zero-Shot Reasoners. In *Advances in Neural Information Processing Systems*, 2022.

Kuang-Huei Lee, Xinyun Chen, Hiroki Furuta, John Canny, and Ian Fischer. A Human-Inspired Reading Agent with Gist Memory of Very Long Contexts. In *International Conference on Machine Learning*, 2024.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems*, 2020.

Alexander H. Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, Alexandre Sablayrolles, Amélie Héliou, Amos You, Andy Ehrenberg, Andy Lo, Anton Eliseev, Antonia Calvi, Avinash Sooriyarachchi, Baptiste Bout, Baptiste Rozière, Baudouin De Monicault, Clémence Lanfranchi, Corentin Barreau, Cyprien Courtot, Daniele Grattarola, Darius Dabert, Diego de las Casas, Elliot Chane-Sane, Faruk Ahmed, Gabrielle Berrada, Gaëtan Ecrepont, Gauthier Guinet, Georgii Novikov, Guillaume Kunsch, Guillaume Lample, Guillaume Martin, Gunshi Gupta, Jan Ludziejewski, Jason Rute, Joachim Studnia, Jonas Amar, Joséphine Delas, Josselin Somerville Roberts, Karmesh Yadav, Khyathi Chandu, Kush Jain, Laurence Aitchison, Laurent Fainsin, Léonard Blier, Lingxiao Zhao, Louis Martin, Lucile Saulnier, Luyu Gao, Maarten Buyl, Margaret Jennings, Marie Pellat, Mark Prins, Mathieu Poirée, Mathilde Guillaumin, Matthieu Dinot, Matthieu Futeral, Maxime Darrin, Maximilian Augustin, Mia Chiquier, Michel Schimpf, Nathan Grinsztajn, Neha Gupta, Nikhil Raghuraman, Olivier Bousquet, Olivier Duchenne, Patricia Wang, Patrick von Platen, Paul Jacob, Paul Wambergue, Paula Kurylowicz, Pavankumar Reddy Muddireddy, Philomène Chagniot, Pierre Stock, Pravesh Agrawal, Quentin Torroba, Romain Sauvestre, Roman Soletskyi, Rupert Menneer, Sagar Vaze, Samuel Barry, Sanchit Gandhi, Siddhant Waghjale, Siddharth Gandhi, Soham Ghosh, Srijan Mishra, Sumukh Aithal, Szymon Antoniak, Teven Le Scao, Théo Cachet, Theo Simon Sorg, Thibaut Lavril, Thiziri Nait Saada, Thomas Chabal, Thomas Foubert, Thomas Robert, Thomas Wang, Tim Lawson, Tom Bewley, Tom Bewley, Tom Edwards, Umar Jamil, Umberto Tomasini, Valeriia Nemychnikova, Van Phung, Vincent Maladière, Virgile Richard, Wassim Bouaziz, Wen-Ding Li, William Marshall, Xinghui Li, Xinyu Yang, Yassine El Ouahidi, Yihan Wang, Yunhao Tang, and Zaccharie Ramzi. Ministral 3, 2026. URL https://arxiv.org/abs/2601.08584.

Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2): 129–137, 1982.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-Refine: Iterative Refinement with Self-Feedback. In *Advances in Neural Information Processing Systems*, 2023.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. RouteLLM: Learning to Route LLMs with Preference Data. In *International Conference on Learning Representations*, 2025.

Siru Ouyang, Jun Yan, I-Hung Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long T. Le, Samira Daruki, Xiangru Tang, et al. ReasoningBank: Scaling Agent Self-Evolving with Reasoning Memory. In *International Conference on Learning Representations*, 2026.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as Operating Systems. *arXiv preprint arXiv:2310.08560*, 2023.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, et al. QuALITY: Question Answering with Long Input Texts, Yes! In *Conference of the North American Chapter of the Association for Computational Linguistics*, 2022.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative Agents: Interactive Simulacra of Human Behavior. In *ACM Symposium on User Interface Software and Technology*, 2023.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-Weighted Regression: Simple and Scalable Off-Policy Reinforcement Learning. *arXiv preprint arXiv:1910.00177*, 2019.

Dean A Pomerleau. Efficient training of artificial neural networks for autonomous navigation. *Neural computation*, 3(1):88–97, 1991.

Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, et al. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300*, 2024.

Haozhan Shen, Kangjia Zhao, Tiancheng Zhao, Ruochen Xu, Zilun Zhang, Mingwei Zhu, and Jianwei Yin. ZoomEye: Enhancing Multimodal LLMs with Human-Like Zooming Capabilities through Tree-Based Image Exploration. In *Conference on Empirical Methods in Natural Language Processing*, 2025.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language Agents with Verbal Reinforcement Learning. In *Advances in Neural Information Processing Systems*, 2023.

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

Mirac Suzgun, Mert Yuksekgonul, Federico Bianchi, Dan Jurafsky, and James Zou. Dynamic Cheatsheet: Test-Time Learning with Adaptive Memory. *arXiv preprint arXiv:2504.07952*, 2025.

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

V Team, Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, et al. GLM-4.5V and GLM-4.1V-Thinking: Towards Versatile Multimodal Reasoning with Scalable Reinforcement Learning. *arXiv preprint arXiv:2507.01006*, 2025.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An Open-Ended Embodied Agent with Large Language Models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL `https://openreview.net/forum?id=ehfRiF0R3a`.

Haochen Wang, Xiangtai Li, Zilong Huang, Anran Wang, Jiacong Wang, Tao Zhang, Jiani Zheng, Sule Bai, Zijian Kang, Jiashi Feng, Zhuochen Wang, and Zhaoxiang Zhang. Traceable Evidence Enhanced Visual Grounded Reasoning: Evaluation and Method. In *International Conference on Learning Representations*, 2026.

Wenbin Wang, Liang Ding, Minyan Zeng, Xiabin Zhou, Li Shen, Yong Luo, Wei Yu, and Dacheng Tao. Divide, Conquer and Combine: A Training-Free Framework for High-Resolution Image Perception in Multimodal Large Language Models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 7907–7915, 2025a.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *International Conference on Learning Representations*, 2023.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent Workflow Memory. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 63897–63911. PMLR, 2025b.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*, 2022.

Penghao Wu and Saining Xie. V*: Guided Visual Search as a Core Mechanism in Multimodal LLMs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-MEM: Agentic Memory for LLM Agents. In *Advances in Neural Information Processing Systems*, 2025.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. Large Language Models as Optimizers. In *International Conference on Learning Representations*, 2024.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing Reasoning and Acting in Language Models. In *International Conference on Learning Representations*, 2023.

Weiran Yao, Shelby Heinecke, Juan Carlos Niebles, Zhiwei Liu, Yihao Feng, Le Xue, Rithesh Murthy, Zeyuan Chen, Jianguo Zhang, Devansh Arpit, et al. Retroformer: Retrospective Large Language Agents with Policy Gradient Optimization. In *International Conference on Learning Representations*, 2024.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Zhi Huang, Carlos Guestrin, and James Zou. TextGrad: Automatic "Differentiation" via Text. *arXiv preprint arXiv:2406.07496*, 2024.

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. *arXiv preprint arXiv:2602.02474*, 2026a.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, et al. Agentic Context Engineering: Evolving Contexts for Self-Improving Language Models. In *International Conference on Learning Representations*, 2026b.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Khai Hao, Xu Han, Zhen Leng Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. ∞Bench: Extending Long Context Evaluation Beyond 100K Tokens. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15262–15277. Association for Computational Linguistics, 2024.

Xintong Zhang, Zhi Gao, Bofei Zhang, Pengxiang Li, Xiaowen Zhang, Yang Liu, Tao Yuan, Yuwei Wu, Yunde Jia, Song-Chun Zhu, et al. Adaptive Chain-of-Focus Reasoning via Dynamic Visual Search and Zooming for Efficient VLMs. *arXiv preprint arXiv:2505.15436*, 2025a.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025b.

Yi-Fan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu, Shuangqing Zhang, Junfei Wu, Feng Li, Kun Wang, Qingsong Wen, Zhang Zhang, et al. MME-RealWorld: Could Your Multimodal LLM Challenge High-Resolution Real-World Scenarios that are Difficult for Humans? In *International Conference on Learning Representations*, 2025c.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. ExpeL: LLM Agents Are Experiential Learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024.

Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. DeepEyes: Incentivizing "Thinking with Images" via Reinforcement Learning. In *International Conference on Learning Representations*, 2026.

Huichi Zhou, Yihang Chen, Siyuan Guo, Xue Yan, Kin Hei Lee, Zihan Wang, Ka Yiu Lee, Guchun Zhang, Kun Shao, Linyi Yang, et al. Memento: Fine-tuning LLM Agents without Fine-tuning LLMs. *arXiv preprint arXiv:2508.16153*, 2025.

A Implementation Details

The appendix presents implementation and tool interfaces in Appendix A, benchmark protocols and baseline adaptations in Appendix B, prompts in Appendix C, and further results in Appendix D.

A.1 Algorithm

Algorithm 1 summarizes training. Each round updates the controller from grouped task rollouts. Before the final round, clustered failure chains yield candidate memory edits, which the validation gate evaluates with the fixed cloned controller π_{bc}. Testing pairs the final memory with the controller selected by validation accuracy.

Algorithm 1: Training procedure of MemSteer.

Input : frozen solver F, generator $\mathcal{G}$, seed memory $\mathcal{M}_0$, splits $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{val}}$, rounds R
Output: selected controller $\pi^\star$ and memory $\mathcal{M}$

$\mathcal{M} \leftarrow \mathcal{M}_0$, authoring guidance $q \leftarrow \emptyset$
/* **Warm start** */
Elicit solver self-selections with $\mathcal{M}$ on $\mathcal{D}_{\text{train}}$ and clone them into π_θ and the gate controller π_{bc}
for $r = 1, \ldots, R$ **do**
  /* **Controller optimization** */
  For every $x \in \mathcal{D}_{\text{train}}$, collect rollouts using fixed $G = 4$ or the $G = 2 \rightarrow 6$ configuration (Table 6)
  Score each rollout with $\widetilde{R}$ (Eq. 4) and estimate α_{noise} from single-signature groups (Eq. 5)
  Compute A_i on multi-signature groups (Eq. 6) and update θ by minimizing Eq. 7
  if $r < R$ **then**
    /* **Memory evolution** */
    Build failure chains from the failed rollouts of action-sensitive tasks
    Cluster the chains with K-means and let $\mathcal{G}$ propose one edit per cluster under guidance q
    Discard edits with sample-specific content or near-duplicates and apply the rest to obtain $\mathcal{M}'$
    Evaluate π_{bc} with $\mathcal{M}$ and with $\mathcal{M}'$ on $\mathcal{D}_{\text{val}}$, then select $\mathcal{M}$ by the validation gate
    Rewrite q from the gate outcome and refit π_{bc} on the retained memory
Select $\pi^\star$ among the warm start and the controllers of rounds $1, \ldots, R$ as the most accurate with $\mathcal{M}$ on $\mathcal{D}_{\text{val}}$
return $\pi^\star$, $\mathcal{M}$

A.2 Method Details

State and Candidates. The progress features z_t are the decision index divided by T, the confidence of the latest assessment (low, medium, high mapped to 0, 0.5, 1), a flag for unresolved evidence, the focus depth (one minus the area of the current artifact relative to the source), a flag for a focused artifact, the fractions of successful and invalid tool calls, and whether the last call succeeded. A policy is feasible when its family has a tool that applies to an available artifact, for example ENHANCE only after a focused artifact exists. The candidate set keeps the k feasible policies with the highest cosine similarity to $e(s_t)$.

Decisions and Episodes. At most $n = 3$ policies are kept per decision. If more candidates fire, the n with the highest probability are kept. If none fires, the highest-scoring candidate is kept. STOP is masked at $t = 0$ unless the first assessment is confident and leaves nothing unresolved. Evaluation thresholds every probability at 0.5 instead of sampling. Besides STOP and the budget T, an episode ends after two consecutive returns to an already seen artifact state or three consecutive invalid tool calls.

Controller Training. The warm start fits the stop head to the solver's stop decisions and the candidate scores to the single entry it names, with label smoothing 0.1. In Eq. 5, set $\alpha_{\text{noise}} = 0$ when $v_{\text{mix}} = 0$; when $v_{\text{mix}} > 0$ but no group contains a single signature, set $\alpha_{\text{noise}} = 1$. Otherwise, use the value from Eq. 5. Groups whose rollouts share one signature or one return carry no advantage and are left out of the update. The weights w_t are normalized to mean one over decisions with $\widehat{A}_t \neq 0$ and set to one elsewhere.

Memory Evolution. A failure chain records the first assessment, the candidates and selected policies at each decision, each tool call and its status, the following assessment, the termination reason, the predicted and correct answers, and whether the solver answers correctly without tools. Chains are restricted to action-sensitive tasks when at least eight such chains exist. For n failed tasks, K is chosen within two of $n/20$ and no larger than $\min(n/8, 16)$ using the silhouette score, with Davies–Bouldin tie breaking (Rousseeuw, 1987; Davies & Bouldin, 1979). The generator receives cluster statistics, the current memory, and at most twelve chains, split between the six nearest and six farthest from the centroid. Candidates containing a sample identifier, file name, numeric bounding box, quoted sample-specific literal, or answer letter are revised up to three times. A candidate is discarded if its embedding has cosine similarity of at least 0.92 to a stored policy. A policy becomes eligible for revision after at least three selections and a failure rate at least 0.10 above the round's error rate. It becomes eligible for deletion after hurting at least three more tasks than it helped. At most one policy is deleted per round, and the warm-start controller is refit on the adopted memory after each gate.

Table 9: Operation families in each environment. A selected policy exposes only the tools of its family.

	Initial view	SEARCH	FOCUS	ENHANCE
Images	image downscaled to $\leq$2048 px	2×2 grid crop with 3% overlap	zoom into a box at scale 1–3	sharpen a focused crop
Documents	first 1,500 characters	outline into 4 consecutive sections with previews	read a span of $\leq$4,000 characters	keyword search in the focused span
ALFWorld	first observation and goal	go to, look	open, close, take, put, examine, inventory	clean, heat, cool, use

A.3 Seed Policy Memory

Each seed policy specifies applicability, guidance, a success condition, and failure recovery. Table 10 lists the applicability and guidance of the twelve entries used for images, documents, and ALFWorld.

A.4 Meta Loop

Up to six authoring rules guide policy specificity and the choice between tool policies and reasoning skills. After each gate, the current rules replace the incumbent if the candidate is not rejected and its validation accuracy change meets or exceeds that of the previous incumbent. A language model rewrites the rules using the gate outcome, generation statistics, and validation selection counts (Appendix C).

A.5 Hyperparameters

B Benchmarks and Baselines

B.1 Benchmarks, Splits, and Metrics

For paired item-level comparisons, the exact two-sided McNemar test (McNemar, 1947) uses discordant counts b and c. On HR-Bench 4K, the four option orders of each source question are kept together for uncertainty analysis.

B.2 Baseline Details

Direct answer uses MemSteer's answer prompt on the initial view without tools and ends in "Answer: X". **Zero-shot CoT** uses the two-stage "Let's think step by step" prompt (Kojima et al., 2022), and **CoT-SC** votes over three samples at temperature 0.7 (Wang et al., 2023). Both see the initial view. **Full document** supplies the entire text. **RAG** retrieves the eight most similar 1,500-character

Table 10: Seed memory $\mathcal{M}_0$. One entry per routing family in each environment.

	Family	Applicability	Guidance
Images	SEARCH	The relevant visual region cannot yet be located from the currently available evidence.	Systematically inspect coarse spatial partitions, retain promising candidates, and avoid revisiting covered regions.
	FOCUS	The relevant region is known, but its labels, symbols, or numerical values are too small to read reliably.	Inspect that region at higher local detail while preserving enough surrounding context for alignment.
	ENHANCE	A focused artifact contains the correct evidence source but remains blurred, low-contrast, or numerically ambiguous.	Improve the clarity of the existing focused artifact before searching a different region.
	STOP	Every fact required for the requested comparison, count, route judgment, or calculation has been explicitly observed.	Stop gathering visual evidence only when remaining uncertainty cannot change the result.
Documents	SEARCH	The section holding the required evidence cannot yet be located from the currently available text.	Survey the document in coarse consecutive sections, retain promising ones, and avoid re-surveying covered spans.
	FOCUS	A promising section is known, but its exact statements have only been seen as truncated previews.	Read that span at full fidelity while keeping enough surrounding context to interpret it correctly.
	ENHANCE	A focused span holds the right material but the decisive value, name, or condition is still buried in it.	Search the focused span for the decisive term before surveying any new section.
	STOP	Every fact required for the requested comparison, count, or judgment has been explicitly read.	Stop gathering evidence only when remaining uncertainty cannot change the result.
ALFWorld	SEARCH	The object or receptacle the task needs has not been located yet.	Move to the receptacles where that kind of object is usually kept, nearest first, and look before moving on.
	FOCUS	The needed object or receptacle is in view but has not been handled yet.	Open closed receptacles, take the object the task names, and put it where the task says.
	ENHANCE	The task object is held but the task asks for it cleaned, heated, cooled, or examined under a light.	Go to the appliance the task implies and apply it to the held object before placing it.
	STOP	Every requirement of the task has been carried out.	Stop acting only when the task is completed.

chunks in document order and uses zero-shot CoT. **ReadAgent** writes gists at model-chosen page breaks of 280–600 words and looks up at most five pages (Lee et al., 2024). **ReAct-style tool use** lets the solver use any available tool or stop under MemSteer's budgets and termination rules. **ACE** builds a sectioned playbook in one training pass with the official generator, reflector, and curator prompts (Zhang et al., 2026b). On images, its answering model sees the initial image with the playbook, while reflection and curation use answer traces and feedback as text. **ExpeL** edits an insight list with ADD, EDIT, UPVOTE, and DOWNVOTE and retrieves two successful demonstrations (Zhao et al., 2024). **AWM** induces up to three workflows from successful trajectories (Wang et al., 2025b). Both use MemSteer's round-one training rollouts. On images, their induction reads the question, tool sequence, and outcome as text, while the solver still receives image artifacts and ReAct-style visual tools. **MemSkill** selects memory-construction skills for interaction spans, writes memories with an executor, and refines the skills from task feedback with a designer (Zhang et al., 2026a). For images, each span contains the solver's textual observations and tool calls, while episode outcomes supervise skill learning. The resulting memory enters the frozen vision-language solver's context alongside image and crop artifacts. MemSkill's controller governs memory construction, while the solver retains ReAct-style visual tools. **All policies in context** exposes every non-stop

Table 11: Implementation hyperparameters.

Episode		**Controller and warm start**	
retrieved candidates k	8	state / policy embedding	1024
policies per decision	$\leq$3	hidden units	256
decision budget $T_{\max}$	12	state features	8
document search budget	16 / 20	parameters	861,698
ALFWorld budget	50	clone epochs / batch	8 / 64
repeated-state / invalid limit	2 / 3	clone learning rate	10^{-3}
answer reasoning tokens	700	label smoothing	0.1
document search reasoning tokens	2,048		
solver temperature	0 (CoT-SC 0.7)		
Policy round		**Memory evolution**	
rounds R / rollouts G	3 / 2–6	max clusters / min support	16 / 8
epochs / batch	4 / 32	target cluster size	20
Adam learning rate	10^{-4}	chains per cluster	12
gradient-norm clip	1.0	duplicate cosine	0.92
temperature β / cap $w_{\max}$	0.5 / 10	generator temperature	0.2
entropy c_H / divergence c_{KL}	0.02 / 0.1	memory cap / deletes	12 / 1
clip ratio (ablation)	0.2	guidance rules	$\leq$6
policy cost λ_c	0.01		

Table 12: Benchmarks and splits. Splits are drawn at random (seed 42) from the pool in the last column. ALFWorld uses its official sets.

	Benchmark	Train	Val	Test	Pool and preprocessing
Images	TreeBench	80	40	285	all 405 questions
	HR-Bench 4K	25	15	160	questions, with the 4 cyclic option orders kept together (640 test rows)
	MME-RealWorld-Lite	100	60	400	multiple-choice questions with options A–E
	V*Bench	40	30	121	all 191 questions
Documents	LongBench v2	45	40	260	4-option questions with context $\leq$600K characters
	∞Bench En.MC	40	30	159	all 229 questions
	QuALITY-hard	100	60	300	hard dev questions, articles $\leq$60K characters, deduplicated
	AmongUs-X	100	60	300	one-impostor games from 64 self-/cross-play experiments, using public speeches, votes, and ejections only, verdict lines removed
	ALFWorld	100	140	134	train games balanced over 6 task types, valid-seen, valid-unseen

policy. **Random selection** draws up to three candidates per decision. **Solver self-selection** asks the solver to choose one (Appendix C).

C PROMPTS

The following excerpts reproduce the prompts with field and tool names expanded in words and output schemas summarized at the end.

Generator. The system prompt is followed by the current authoring guidance. The user message holds the cluster statistics, the current memory, and the sampled failure chains.

> You are the slow-loop generator for an agentic Policy Memory bank. Each supplied failure is a complete causal reasoning chain from one incorrect tool-

> augmented reasoning episode, not an isolated state. Analyze every chain in the cluster and identify the shared, reusable failure mechanism. A chain may carry a flag stating that the frozen model answered that sample correctly without any tool, so the tool chain itself caused the failure.
>
> Generate exactly one abstract natural-language policy that complements the current memory bank. It may be a *tool policy*, guidance for when one operation family should be used, whose family must be search a new region, focus on a candidate region, or enhance the current evidence, and never stop, because the controller decides stopping with its own head. It may instead be a *reasoning skill*, a reusable multi-step problem-solving method, whose family must be the adaptive action.
>
> A reasoning skill may describe an overall method, but one controller selection still authorizes only the next useful atomic tool action. After the real tool artifact is observed, the controller must select a policy again.
>
> Do not copy sample identifiers, option letters, answers, entity names that occur only in one example, exact values, file paths, layout annotations, or bounding boxes. Do not predict a future observation.
>
> *Output: a diagnosis (the shared failure pattern and why the current memory was insufficient) and one policy with its type, applicability, guidance, success condition, failure recovery, and operation family.*

Meta Loop. The prompt below rewrites the authoring guidance after each gate (Appendix A.4).

> You maintain the authoring guidance for a slow-loop generator that writes abstract natural-language policies for a tool-using agent. You never see the tasks. You see how the policies the generator wrote fared on an unseen split, and which of them the agent chose to use.
>
> Rewrite the guidance so the generator's next policies fare better. Keep only guidance that the evidence supports, drop guidance the evidence contradicts, and merge duplicates. Guidance is about how to write and route policies (scope, specificity, when to prefer a tool policy over a reasoning skill, what makes a policy get selected), never about any particular question.
>
> *Output: a list of guidance rules and a one-paragraph rationale.*

Solver. The frozen solver works in three phases under the system prompt below, shown for images. Each request adds the question, the answer choices, the causal working memory, the selected policy, the tool schemas, and the artifacts. The document solver uses the same prompt with text artifacts.

> You are one frozen visual agent operating in an explicit phase: observe, act, or answer. Use only the question, answer choices, causal working memory, selected policy, tool schemas, and image artifacts provided in the current request. You never receive the gold answer, human explanation, layout annotations, or a future observation.
>
> In the observe phase, report only facts grounded in the supplied artifacts and what information remains unresolved. Artifacts may be bounded previews or crops, and a whole-image artifact is downscaled before it reaches you, so small text, small objects, and fine detail are not legible in it. Absence from one artifact never proves absence from the source image, and a detail that has not been resolved at a legible scale is not yet known. Describe what the artifacts do contain even when the target is not visible, and never return empty evidence. Do not select an answer.
>
> In the act phase, translate the selected abstract Policy Memory into exactly one grounded image function call. A selected multi-step reasoning skill authorizes only the next useful atomic visual action and never an internal sequence. Use only an artifact and a function listed in the request. Do not predict the result of the tool.
>
> In the answer phase, reason step by step from the causal evidence. Check each answer choice against what the artifacts actually show and rule choices out explicitly. End with one final line of the form "Answer: X", where X is one allowed option letter.

Observe and Act Instructions. The observe instruction defines the confidence levels that enter the controller state.

> **Observe.** Report the visible evidence (what the new artifacts show, combined with useful prior evidence, never empty), the unresolved information (specific evidence still needed, or nothing), and a confidence level. Confidence is high only when the detail the question turns on has been read in a supplied artifact at a scale where it is legible, medium when the right region is located but its detail is not yet legible, and low otherwise. Optionally list candidate regions, each visually grounded, with a description and a box relative to the named artifact. For documents, confidence is high only when the statement the question turns on has been read verbatim, and medium when the right span is located but only its preview has been seen.
>
> **Act.** Return one function call with a function from the available functions and a source artifact. Include a box and a scale only when the selected function requires them.

Solver Self-Selection. This prompt produces the warm-start targets and the self-selection baseline.

> You choose the next abstract policy for a question-answering agent. Read the current state and every candidate policy. Select the single policy whose applicability best matches the state. Choose the stop policy only when the accumulated evidence is already sufficient to answer.
>
> *Output: one policy from the candidates.*

ExpeL Insight Extraction. Training trajectories are grouped by task and passed in turn to the prompt below, which edits a running insight list. More details are in Appendix B.2.

> You extract reusable insights from an agent's past attempts at a task family. You are given the existing insight list and new experience. Modify the list with these operations only: ADD a new insight, EDIT the content of an existing insight, UPVOTE to agree with an existing insight, DOWNVOTE to disagree with an existing insight. Emphasize prevalent failure patterns and best practices. Do not mention any specific sample, option letter, or answer.
>
> *Input: the existing insights and the new experience. Output: a list of operations, each with its type, the index of the insight, and the text for an addition or edit.*

AWM Workflow Induction. Up to 20 successful training trajectories are passed to the prompt below, and at most three induced workflows are kept.

> You induce reusable workflows from an agent's successful trajectories. A workflow is a textual description plus an ordered series of steps. Induce them at a fine granularity, a sub-routine that recurs across several tasks rather than a whole task instruction. Replace example-specific values with placeholders. Do not mention any specific sample, option letter, or answer.
>
> *Input: the successful trajectories. Output: a list of workflows, each with a description and its steps.*

D Additional Results

D.1 Memory Update Validation

Table 13 reports paired validation outcomes for candidate memory updates. Here N_{val} is the number of validation tasks, b and c count tasks solved exclusively by the candidate and incumbent memories, and p_{harm} is the exact one-sided harm probability for those discordant outcomes. Decisions use either Eq. 8 or a harm screen that rejects a candidate when $c - b \geq \max(3, \lceil 0.075 N_{\mathrm{val}} \rceil)$ and $p_{\mathrm{harm}} < 0.05$. The Mistral-Large-2512 AmongUs-X run retained the equally sized incumbent after two small gains.

Table 13: Paired validation results for candidate memory updates.

Setting	Round	N_{val}	b	c	p_{harm}	Decision
Qwen3-VL / TreeBench	1	40	4	3	0.773	adopt
	2	40	1	3	0.312	retain
Qwen3-VL / HR-Bench 4K	1	58	6	2	0.965	adopt
	2	60	7	2	0.980	adopt
Qwen3-VL / MME-RealWorld	1	54	1	3	0.312	adopt
	2	59	3	4	0.500	adopt
Qwen3-VL / V*	1	30	2	3	0.500	adopt
	2	30	3	3	0.656	adopt
GLM-4.6V / TreeBench	1	40	0	0	1.000	retain
	2	40	1	1	0.750	retain
GLM-4.6V / HR-Bench 4K	1	60	2	1	0.875	adopt
GLM-4.6V / MME-RealWorld	1	60	3	2	0.812	adopt
	2	60	2	4	0.344	retain
GLM-4.6V / V*	1	30	1	1	0.750	retain
	2	30	3	1	0.938	adopt
Mistral-L / LongBench v2	1	39	1	9	0.011	retain
	2	39	6	11	0.166	adopt
Mistral-L / ∞Bench	1	30	3	2	0.812	adopt
	2	30	4	2	0.891	adopt
Mistral-L / AmongUs-X	1	60	6	4	0.828	retain
	2	60	7	4	0.887	retain
Mistral-L / QuALITY	1	58	3	8	0.113	adopt
	2	59	4	4	0.637	adopt
Qwen3.5 / LongBench v2	1	40	5	4	0.746	adopt
	2	40	4	4	0.637	retain
Qwen3.5 / ∞Bench	1	30	0	0	1.000	retain
	2	30	0	1	0.500	retain
Qwen3.5 / AmongUs-X	1	60	4	7	0.274	retain
	2	60	3	6	0.254	retain
Qwen3.5 / QuALITY	1	60	7	7	0.605	retain
	2	60	4	5	0.500	retain

D.2 Hyperparameter Sensitivity

The accuracy values underlying Table 8 appear to two decimal places in Table 14.

Table 14: Detailed hyperparameter sensitivity in test accuracy (%). Defaults are shaded.

Parameter	Value	Qwen / AmongUs-X	GLM / TreeBench
Candidates k	4	76.00	50.53
Candidates k	8	79.00	51.59
Candidates k	12	73.67	48.77
Policies / decision n	1	75.00	51.23
Policies / decision n	2	76.33	49.82
Policies / decision n	3	79.00	51.59
Rollouts / task G	2	75.67	48.77
Rollouts / task G	4	75.67	49.12
Rollouts / task G	6	79.00	51.59
Policy cost λ_c	0	73.00	50.18
Policy cost λ_c	0.01	79.00	51.59
Policy cost λ_c	0.05	74.00	51.58

D.3 Case Study Details

Table 15 lists the three policies generated in the LongBench v2 case study. They address repeated search, premature stopping, and answers that require combining indirect evidence.

Table 15: Policies generated after round 1 in the LongBench v2 case study (Qwen3.5-27B).

Failure mode → policy	Applicability	Guidance
Repeated search → **Exhaust the search hits** (tool policy)	A document search returned a large number of candidate chunks (more than the five shown) and the agent needs to examine all of them to answer the question.	Do not repeat the identical search. Instead, read the source document in consecutive sections that cover the regions where the search hits were reported, or refine the search query to be more specific (for example, by adding a distinguishing term) to reduce the result set to a manageable size. If the document is structured (for example, a table), consider reading the full text column in batches to manually identify and deduplicate matches.
Premature stop → **Verify before stopping** (tool policy)	The agent has gathered evidence from search snippets or truncated previews and is considering stopping to answer a multiple-choice question, but the exact phrasing needed to distinguish between closely worded answer options has not been verified in the full source text.	Before stopping, identify the specific phrases that differentiate the remaining answer candidates. If those phrases have only been seen in truncated search-result snippets, do not stop. Instead, read the most promising hit in full, with its surrounding context, and confirm the exact wording. Only stop when the distinguishing phrases have been read verbatim.
No explicit answer → **Synthesize indirect evidence** (reasoning skill)	The available evidence is indirect or distributed across multiple snippets, and no single explicit statement directly answers the question.	Identify the key entities or conditions in the question. For each answer choice, list the evidence that supports or contradicts it from the gathered text. Infer the correct answer by eliminating choices that are contradicted or unsupported, and selecting the choice best supported by the weight of indirect evidence.

D.4 Trajectory Examples

Figures 8 and 9 compare the warm start and trained controller on four HR-Bench 4K questions. In the first pair, both initially select the same policies, but the trained controller ranks local detail first and reads the decisive evidence before stopping. In the road-sign pair, the warm start repeatedly inspects regions without the target and fails after seven and nine decisions, while the trained controller reads the signs and answers after two and three decisions. Across these examples, policy ranking directs inspection toward the missing visual evidence.

D.5 Contrastive Rollouts

Figure 10 compares four final-round LongBench v2 rollouts with Mistral-Large-2512. The generated policy *read before stopping* applies while candidate sections remain unread. Both rollouts selecting it answer correctly, whereas both selecting close reading stop after the outline and fail. Their different choices and outcomes form the contrastive signal in Table 6, and Eq. 6 credits selections that include the generated policy.

D.6 Adopted Entries and Benchmark Results

Table 16 shows that adopted policies direct the solver toward missing evidence before stopping. The image entries redirect search across regions, while the document entries call for reading promising spans in full. Tables 17 to 20 report the benchmark scores underlying the main comparisons.

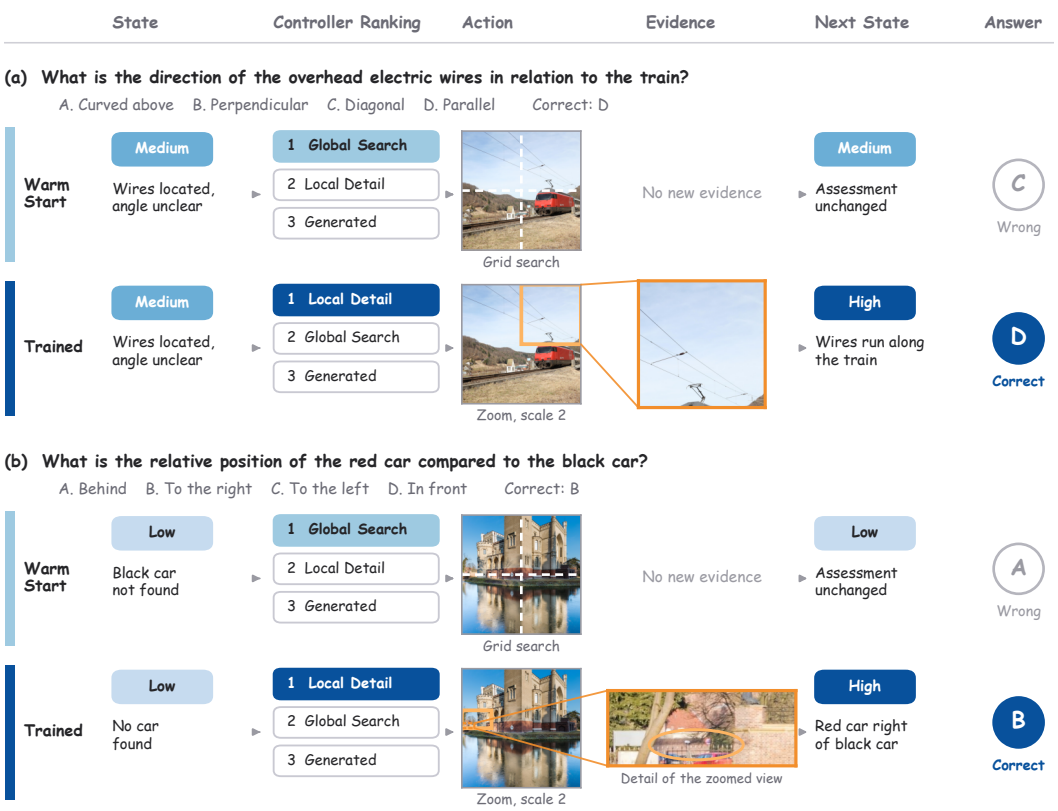


Figure 8: Warm start and trained controller on two HR-Bench 4K questions. Entries follow selection probability. Generated marks a new entry.

Table 16: Generated entries adopted in four settings, abridged. Share: fraction of the non-stop test selections that select the entry.

Setting	Applicability	Guidance	Share
Qwen3-VL-32B, V* tool policy, SEARCH	An initial search or focus leaves the required object or attribute unconfirmed.	Explore a different candidate region or widen the scope, and inspect at least two distinct regions before stopping unless the evidence becomes clear.	33.3%
GLM-4.6V, V* reasoning skill, ADAPTIVE	The target is still missing after a global search, and the agent is about to stop or repeat the same search.	Zoom into the most promising candidate from the global search, and if the target is still absent, consider occlusion or a different search strategy.	27.9%
Mistral-Large-2512, LongBench v2 tool policy, FOCUS	Candidate sections with unresolved details remain unread after an initial survey or partial read.	Before stopping, read the most promising candidate sections in full, and stop only once the details are resolved or no candidate remains.	31.1%
Mistral-Large-2512, ∞Bench reasoning skill, ADAPTIVE	A candidate span holds partial or indirect evidence, but the decisive statement is still unresolved.	Re-read the span in smaller overlapping segments, stay on high-confidence spans before moving on, and switch to keyword search if a tool fails.	26.0%

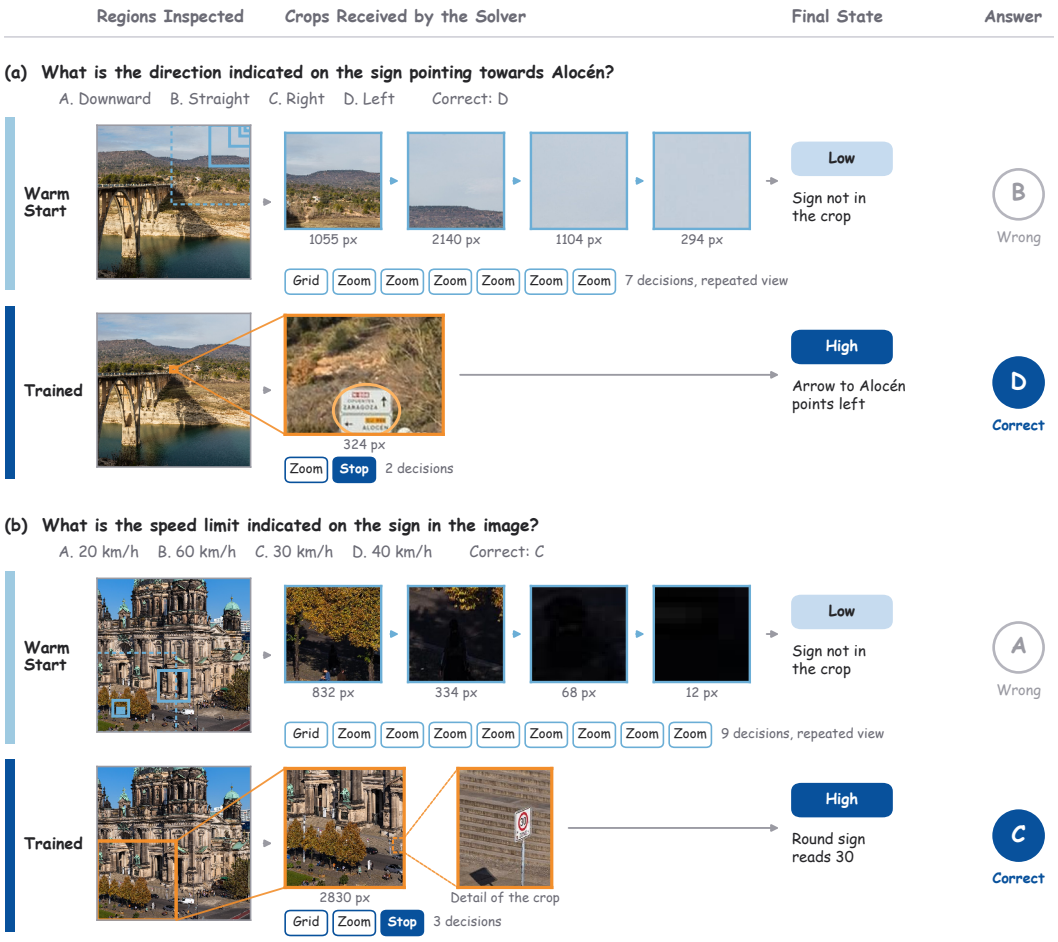


Figure 9: Two HR-Bench 4K visual trajectories. Boxes mark inspected regions, pixel widths show crop size, and chips list tool calls.

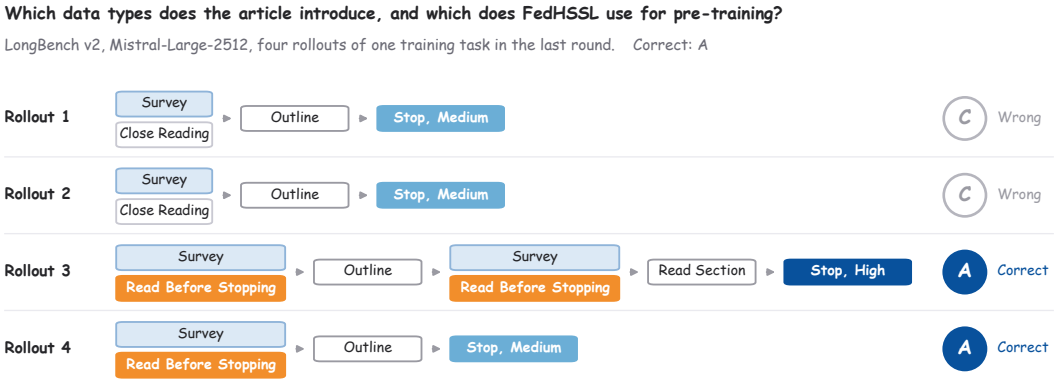


Figure 10: A contrastive rollout group. Entries are listed in the controller's order of probability, and the generated entry is shown in orange.

Table 17: Image benchmark results with Qwen3-VL-32B-Instruct.

Method	TreeBench	HR-4K	MME-RW	V*
Direct answer	49.5	80.2	56.2	84.3
Zero-shot CoT	54.0	82.0	53.2	90.9
CoT-SC (k=3)	51.6	81.9	55.8	89.3
ReAct-style tool use	51.9	80.2	54.2	89.3
ACE	52.3	74.8	57.8	83.5
ExpeL	50.1	80.1	54.9	85.9
AWM	50.5	81.3	58.1	86.7
MemSkill	51.6	81.2	55.2	88.4
All policies in context	51.2	82.2	55.0	86.8
Random selection	50.2	80.1	51.9	91.7
Solver self-selection	50.9	81.6	53.5	86.0
MemSteer (ours)	52.3	82.5	56.1	90.0

Table 18: Image benchmark results with GLM-4.6V.

Method	TreeBench	HR-4K	MME-RW	V*
Direct answer	46.3	72.8	46.0	71.9
Zero-shot CoT	49.1	80.9	54.8	84.3
CoT-SC (k=3)	49.1	81.4	54.5	83.5
ReAct-style tool use	51.9	82.7	48.5	86.8
ACE	47.7	70.9	50.5	79.3
ExpeL	49.1	82.0	46.7	83.4
AWM	51.2	82.3	46.6	85.1
MemSkill	50.2	81.2	48.8	85.1
All policies in context	51.9	82.6	47.5	86.0
Random selection	47.7	79.1	47.9	82.6
Solver self-selection	48.9	80.0	46.0	81.0
MemSteer (ours)	51.6	82.8	51.2	87.6

Table 19: Document benchmark results with Qwen3.5-27B. Grey rows show methods with different document-access protocols and are excluded from ranking.

Method	LongBench v2	∞Bench	AmongUs-X	QuALITY
Direct answer	33.1	49.1	38.7	39.7
Zero-shot CoT	42.7	50.9	53.0	46.3
CoT-SC (k=3)	44.4	49.7	50.7	46.0
ACE	38.8	48.4	62.3	42.3
Full document	49.3	90.3	52.7	80.3
RAG (top-8 chunks)	49.2	81.8	66.3	83.0
ReadAgent	52.6	92.9	64.3	83.7
ReAct-style tool use	38.1	51.6	63.0	69.0
All policies in context	33.5	52.2	58.7	67.0
Random selection	33.6	47.8	51.3	51.3
Solver self-selection	37.7	51.6	59.7	66.3
ExpeL	43.1	53.4	75.0	70.6
AWM	41.7	57.2	73.0	66.7
MemSkill	43.5	59.1	74.7	70.0
MemSteer (ours)	52.5	81.4	79.0	73.7

Table 20: Document benchmark results with Mistral-Large-2512. Grey rows show methods with different document-access protocols and are excluded from ranking.

Method	LongBench v2	∞Bench	AmongUs-X	QuALITY
Direct answer	28.9	41.5	60.7	39.0
Zero-shot CoT	31.9	60.4	31.0	45.0
CoT-SC (k=3)	40.8	64.8	50.7	43.7
ACE	33.8	49.1	63.3	46.7
Full document	39.4	82.4	72.0	75.3
RAG (top-8 chunks)	37.3	79.3	36.0	75.3
ReadAgent	47.1	86.2	56.3	70.0
ReAct-style tool use	32.7	52.2	67.0	55.3
All policies in context	30.4	54.7	71.0	57.1
Random selection	30.1	40.7	64.1	46.2
Solver self-selection	31.5	50.0	70.4	60.6
ExpeL	35.0	54.7	70.0	58.0
AWM	33.1	56.6	69.0	56.3
MemSkill	33.8	56.0	71.0	59.0
MemSteer (ours)	37.3	58.5	74.3	62.3