INFORMATION-CONSTRAINED MEMORY EVOLUTION FOR SELF-IMPROVING LANGUAGE AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

External memory helps language agents improve through experience without retraining their acting models, but seemingly useful lessons can harm decisions and repeated feedback can encourage overfitting. We aim to learn memory edits that improve task performance while controlling how much information the editor receives from reused feedback tasks. We propose Information-Constrained Memory Evolution (ICME), which evaluates each edit through its effect on the entire memory bank, keeping the acting model and retriever fixed. A limited-bit feedback channel trains the editor from evaluations of the same tasks before and after each edit, preserving the expected learning signal; after training, the editor updates fresh memory banks without evaluator feedback. In matched experiments with Qwen3-14B, ICME improves success rates over Auto-Dreamer by 3.9 percentage points on unseen ALFWorld tasks and 5.9 points on ScienceWorld. By linking feedback capacity to a bound on the final memory bank's expected overestimation of performance, this work provides a principled basis for learning from memory edits under repeated feedback reuse.

1 INTRODUCTION

Language agents increasingly tackle tasks that unfold through interaction with tools and environments, from web navigation (Yao et al., 2023) to interactive scientific reasoning (Wang et al., 2022). External memory offers a practical route to continued improvement by carrying useful experience across tasks without repeatedly retraining the acting model (Zhao et al., 2024). The challenge is to make accumulated experience reliably improve future decisions as memory grows and the agent encounters new tasks.

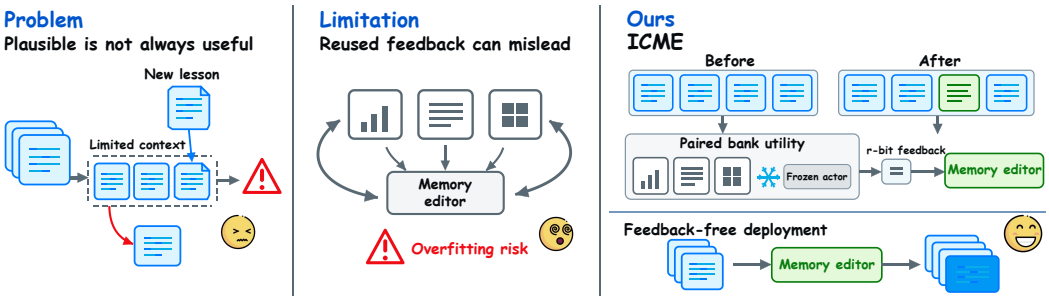

Figure 1: **From plausible lessons to useful memory.** Left: a new lesson can push useful memories out of the limited retrieval context. Middle: repeatedly reusing the same feedback tasks risks overfitting. Right: ICME compares whole-bank utility before and after an edit, then trains the memory editor through an r-bit feedback channel. The learned editor updates fresh memory banks without evaluator feedback.

Recent work has improved how experience becomes reusable guidance. ReasoningBank distills strategies from successful and failed attempts (Ouyang et al., 2026), while ACE maintains evolving playbooks through incremental curation (Zhang et al., 2026c). A lesson's value depends on how it interacts with the existing memory bank. Under a limited context budget, new entries can displace

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.

useful old ones, so even a plausible lesson may degrade downstream decisions. Memory improvement therefore requires understanding how stored knowledge changes the behavior of the agent that uses it.

Outcome-driven learning addresses this issue by optimizing memory behavior against task performance: Memory-R1 learns memory-management policies (Yan et al., 2026), while Auto-Dreamer trains consolidation with downstream and counterfactual rewards (Ye et al., 2026). Yet utility optimization introduces a further challenge when feedback tasks are limited and repeatedly reused. Repeated adaptation to the same feedback tasks can make measured gains overstate performance on new tasks (Dwork et al., 2015). This risk persists with a frozen acting model because its external memory remains adaptive. The unresolved issue is how to learn useful memory changes while controlling dependence on the feedback used to learn them.

We study memory editing that improves downstream decisions while explicitly accounting for the training feedback it consumes. We introduce *Information-Constrained Memory Evolution* (ICME), which treats memory improvement and feedback exposure as coupled design concerns (Figure 1). With the acting model and retriever frozen, ICME learns from an edit's change in whole-bank task utility, capturing interactions between new content and existing memories. A bounded feedback channel preserves the expected policy-gradient signal while limiting information about the reused feedback tasks. The editor learns from the states its own edits produce and, once trained, continues editing fresh memory banks without evaluator feedback.

Our contributions are threefold:

- **We reveal a gap between plausible and useful memory.** In our ALFWorld preliminary study, 23.8% of highly rated lessons reduce utility, and harmful edits are more frequent under strong retrieval competition (Section 3).
- **We propose information-constrained memory evolution.** ICME learns reusable editing behavior from downstream consequences. Our analysis connects feedback capacity to a bound on the final bank's expected optimism under feedback reuse (Section 4.4).
- **ICME improves task performance and transfers its editing capability.** With a frozen Qwen3-14B actor, ICME exceeds Auto-Dreamer by 3.9 and 5.9 percentage points in ALFWorld unseen and ScienceWorld success rates, respectively, in our matched comparisons (Table 1). Starting from empty memory, learned editors also improve mean query utility over fixed editors in all nine transfer conditions (Table 8).

2 Related Work

Experience-based memory. Early agents reuse verbal reflections and cross-task lessons without updating model weights (Shinn et al., 2023; Zhao et al., 2024). AutoGuide conditions guidelines on context, while ReasoningBank and ACE develop success/failure distillation and incremental playbook curation (Fu et al., 2024; Ouyang et al., 2026; Zhang et al., 2026c). The downstream value of these memory updates depends on both their content and their effect on retrieval. Information-Constrained Memory Evolution (ICME) uses measured utility, including retrieval displacement, to supervise edits.

Learning memory decisions. Outcome-driven reinforcement learning replaces heuristic operations with learned policies in Memory-R1 and Mem-α (Yan et al., 2026; Wang et al., 2025a). Credit assignment extends to consolidation-level counterfactual utility in Auto-Dreamer and attribution-based process rewards in AttriMem (Ye et al., 2026; Li et al., 2026); MemEvolve and MemSkill further evolve memory architectures and reusable editing procedures (Zhang et al., 2026a;b). To isolate an edit's effect within the existing bank, ICME assigns credit through paired whole-bank evaluation of single-entry edits, with the actor and retriever frozen.

Feedback efficiency and generalization. MemRL learns retrieval utilities, and RoMeRL improves feedback coverage through reduced-order utility states (Zhang et al., 2026d; Yang et al., 2026). These methods address sparse utility updates; repeatedly querying a fixed task set additionally raises adaptive-overfitting concerns. Adaptive-data-analysis theory relates selection bias to released information (Russo & Zou, 2016; Xu & Raginsky, 2017). Combining this perspective

with unbiased stochastic quantization (Alistarh et al., 2017), ICME releases one r-bit gain code per round, preserves the conditional mean policy gradient, and bounds expected optimism of the final bank. The guarantee quantifies the final bank's dependence on the reused feedback set.

3 Preliminary Study

Does a plausible lesson make useful memory? On the design split D_P, a frozen writer blindly appends one lesson after each episode in six ALFWorld streams, yielding 192 writes. The actor and retriever are also frozen. We measure each write's mean success change Δ_t on 12 paired probes, alongside semantic plausibility, bank occupancy ω_t, and retrieval displacement χ_t averaged over repeated runs. These diagnostics never reach the writer; the training-feedback set D_F is unused. Appendix B.8 gives the protocol and analysis details.

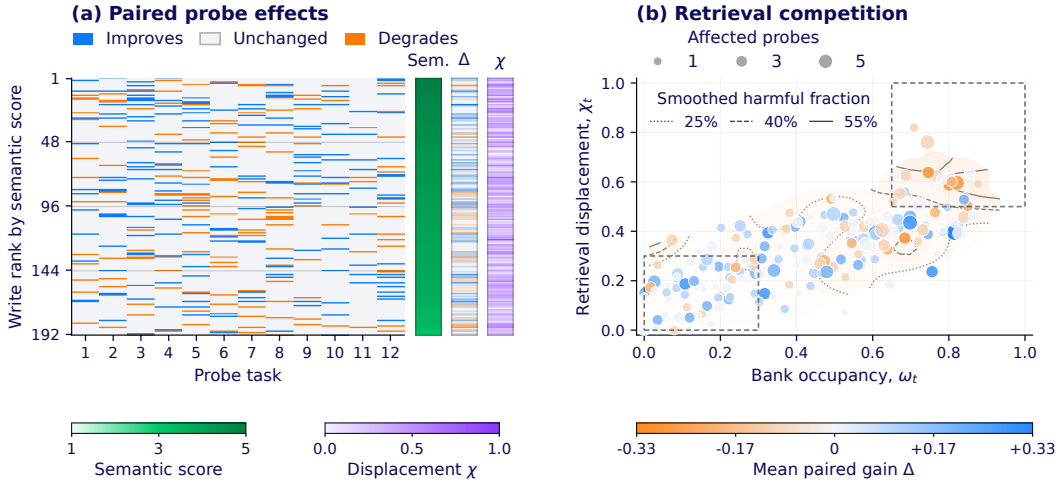

Figure 2: **Plausible lessons can harm task performance.** **(a)** Paired probe effects for 192 writes, ranked by semantic score; side tracks show score, gain Δ_t, and displacement χ_t. **(b)** Gain versus occupancy and displacement. Color denotes gain, size denotes affected-probe count, and contours show smoothed harmful fractions. Appendix B.8.3 details the encoding.

Plausibility is a weak proxy for utility. Even among lessons scoring at least four out of five, 23.8% are harmful, and semantic score correlates only weakly with gain (Spearman $\rho = 0.11$; Figure 2a). Probe effects are often sparse and can offset one another, producing small net effects.

Harm accompanies retrieval competition. The harmful fraction rises from 18.3% in the low-competition region to 59.1% in populated, strongly perturbed banks (Figure 2b; region definitions in Appendix C.1). Together, these findings motivate learning from downstream utility, retaining exposure to harmful edits, and preserving weak feedback under a restricted information budget.

4 Method

Overview. *Information-Constrained Memory Evolution* (ICME) learns memory edits that improve a frozen actor's task performance. Each round, an isolated evaluator measures an edit's utility gain and sends an r-bit reward to train the editor (Figure 3). Stochastic encoding preserves zero gains and the expected policy gradient while limiting feedback capacity. Both training and deployment apply every valid edit, including harmful ones.

Setting. The actor π_θ, tools, tokenizer, and retriever are frozen; only editor parameters ϕ are trained. A fixed task sequence $x_{1:T}$ supplies visible experience; a separate feedback set $D_F = (z_i)_{i=1}^n$, reused across rounds, supplies rewards. Design choices are fixed on disjoint data D_P; audit and test tasks stay sealed until training ends. Each stream starts with empty memory, retaining the

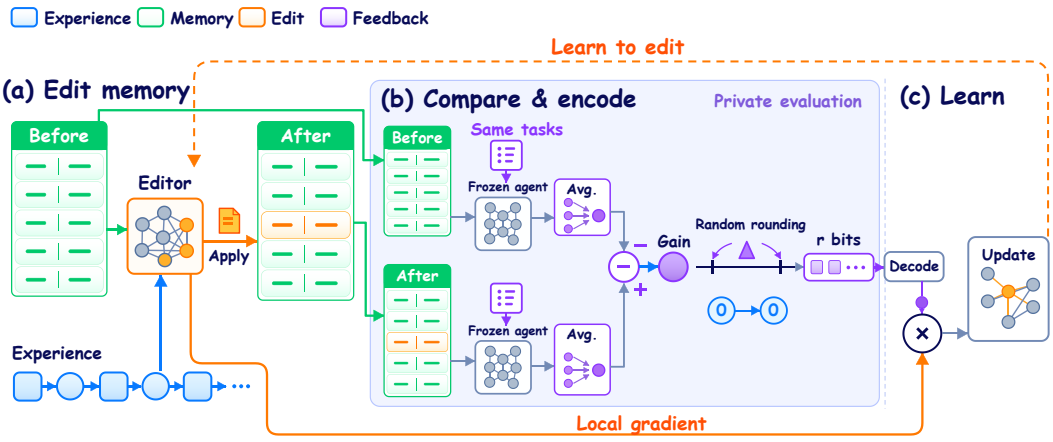

Figure 3: **Overview of Information-Constrained Memory Evolution (ICME). (a)** The editor uses visible experience and the current memory bank to sample a patch, which is applied whenever valid. **(b)** A private evaluator compares the frozen agent's mean utility under the before/after banks on identical feedback tasks and initial states. Zero-preserving stochastic rounding encodes this gain in an r-bit code, the only feedback released to the learner. **(c)** The decoded reward scales the editor's locally computed score gradient to update its parameters. The actor stays frozen throughout; at deployment, the editor is also frozen and memory editing continues without evaluator feedback.

editor parameters and cumulative feedback budget. Let $\mathcal{M}_t^-$ and $\mathcal{M}_t$ denote the banks before and after the round-t edit.

4.1 Editing memory under a shared transition rule

Direct application exposes the editor to the consequences of its own edits, including harmful ones. Each entry stores a condition, a strategy, a stable ID, and a source pointer. For task x_t, frozen cosine retrieval selects from the top k entries within an injected-context limit of L tokens. The actor executes the task, producing experience e_t. Given the fixed serialization $s_t = \mathrm{Serialize}(\mathcal{M}_t^-, e_t)$, the editor samples and applies a patch:

$$p_t \sim q_{\phi_{t-1}}(\cdot \mid s_t), \qquad \mathcal{M}_t = \mathrm{Apply}(\mathcal{M}_t^-, p_t). \tag{1}$$

The patch p_t specifies one operation in $\{\textsc{Add}, \textsc{Replace}, \textsc{Delete}, \textsc{Keep}\}$. The first three modify one entry; KEEP leaves the bank unchanged. Invalid patches, including those exceeding the bank's token capacity C, also leave it unchanged and are not resampled. Successful and failed experiences use the same editor interface. Appendix A.1 specifies prompts, truncation, and edit semantics; Appendix A.2 gives the low-rank parameterization.

4.2 Paired feedback with an exact zero level

Because edits alter both content and retrieval, rewards measure their effect on task utility. Let $u(\mathcal{M}, z) \in [0, 1]$ be the actor's utility with bank $\mathcal{M}$ on task z under deterministic replay. The isolated evaluator compares both banks from identical task snapshots:

$$\widehat{J}_F(\mathcal{M}) = \frac{1}{n}\sum_{i=1}^{n} u(\mathcal{M}, z_i), \qquad \Delta_t = \widehat{J}_F(\mathcal{M}_t) - \widehat{J}_F(\mathcal{M}_t^-) \in [-1, 1]. \tag{2}$$

Thus Δ_t captures both retrieval displacement and execution effects. The evaluator caches each bank's score vector under its exact retrieval-state serialization. Feedback task text, trajectories, scores, and cache contents remain hidden from the editor (Appendix A.3).

For integer $r \geq 2$, set $K = 2^{r-1} - 1$ and $h = 1/K$. The grid $\{-1, -1+h, \ldots, 1\}$ has $2^r - 1$ levels, including $-1, 0, 1$, and fits in r bits with one codeword unused. Stochastic rounding (Alistarh et al., 2017) selects between adjacent levels:

$$\begin{aligned} &\xi_t = K(\Delta_t + 1), \quad j_t = \min\{2K - 1, \lfloor \xi_t \rfloor\}, \quad \rho_t = \xi_t - j_t, \\ &\beta_t \sim \mathrm{Bernoulli}(\rho_t), \qquad c_t = j_t + \beta_t, \qquad \widehat{\Delta}_t = h(c_t - K). \end{aligned} \tag{3}$$

Here j_t indexes the lower level and ρ_t is the probability of rounding upward. Only $c_t \in \{0, \dots, 2K\}$ is transmitted; the learner decodes it as $\widehat{\Delta}_t$. Conditional on the exact gain,

$$\mathbb{E}[\widehat{\Delta}_t \mid \Delta_t] = \Delta_t, \qquad \sigma_r^2(\Delta_t) := \mathrm{Var}(\widehat{\Delta}_t \mid \Delta_t) = h^2 \rho_t(1-\rho_t) \le \frac{h^2}{4}. \tag{4}$$

The expectation is over rounding. Zero gain is exact: $\widehat{\Delta}_t = 0$ whenever $\Delta_t = 0$, even if the edit changes the bank. This prevents random updates at zero gain. For $|\Delta_t| \le h$, the variance is $|\Delta_t|(h - |\Delta_t|)$, so small nonzero gains can still have low signal-to-noise ratios. Learning stability therefore also depends on the noise magnitude (Appendix D.1).

4.3 Learning and deployment

Compute the patch's log-probability gradient (score vector) v_t under the parameters that generated p_t, then take one ascent step:

$$v_t = \nabla_\phi \log q_\phi(p_t \mid s_t)|_{\phi=\phi_{t-1}}, \qquad \phi_t = \phi_{t-1} + \eta \widehat{\Delta}_t v_t. \tag{5}$$

The learning rate $\eta > 0$ is fixed. The log-probability includes every sampled token, without length normalization. Negative rewards give a descent direction for the sampled edit's log-probability; zero rewards give no update. Utility before the edit provides an action-independent baseline without a learned value function (Williams, 1992).

For fixed state s containing bank $\mathcal{M}^-$, the objective is the expected next-edit gain:

$$\mathcal{J}_F(\phi; s) = \mathbb{E}_{p \sim q_\phi(\cdot|s)} \left[\widehat{J}_F(\mathrm{Apply}(\mathcal{M}^-, p)) - \widehat{J}_F(\mathcal{M}^-) \right]. \tag{6}$$

The expectation is over sampled patches, with s and D_F fixed. The score-function identity and (4) give

$$\begin{aligned} \mathbb{E}[\widehat{\Delta}_t v_t \mid s_t, D_F, \phi_{t-1}] &= \nabla_\phi \mathcal{J}_F(\phi; s_t)|_{\phi=\phi_{t-1}}, \\ \mathbb{E}_{\beta_t} \|(\widehat{\Delta}_t - \Delta_t) v_t\|_2^2 &= \sigma_r^2(\Delta_t) \|v_t\|_2^2. \end{aligned} \tag{7}$$

The second identity holds the sampled patch fixed. Quantization preserves the conditional mean gradient and adds noise that scales with the score-vector norm. These identities characterize the one-step empirical objective; terminal performance is evaluated over complete streams (Appendix D.2).

After T rounds, freeze ϕ_T. On fresh tasks, start from an empty bank and repeat (1) without feedback-set queries or parameter updates. Training and deployment share validity checks and direct application, so errors can propagate in both; their state distributions can still differ. We evaluate editor transfer over complete streams, separately from transferring the final training bank to a fresh actor (Appendix B.5).

4.4 Feedback capacity and generalization

Each round releases exactly r bits, even at zero gain, so $B_T = Tr \le B_{\max}$; the bit count accumulates across memory resets. Assume $D_F \sim \mathcal{P}^n$ is independent of V, which collects design resources, visible tasks, and learner random tapes. Fix the horizon and output rule before training and release only reward codes. Rounding uses independent private uniform draws, independent of (D_F, V) and excluded from V (Assumption 1). Define population utility $J(\mathcal{M}) = \mathbb{E}_{Z \sim \mathcal{P}} u(\mathcal{M}, Z)$ and binary entropy $h_2(p) = -p \ln p - (1-p)\ln(1-p)$, with $0 \ln 0 = 0$.

Theorem 1 (Finite-feedback control of bank overfitting)**.** *Under these conditions, for bounded utility $u \in [0,1]$, the final ICME bank satisfies*

$$\begin{aligned} \kappa_T &:= (T-1)\ln(2^r - 1) - \sum_{t=1}^{T-1} \mathbb{E} h_2(\rho_t), \\ \left| \mathbb{E}[\widehat{J}_F(\mathcal{M}_T) - J(\mathcal{M}_T)] \right| &\le \min\left\{1, \sqrt{\frac{\kappa_T}{2n}}\right\} \le \min\left\{1, \sqrt{\frac{(T-1)\ln(2^r-1)}{2n}}\right\}. \end{aligned} \tag{8}$$

Here ρ_t is the rounding probability in (3)*; expectations are over the feedback sample and the adaptive protocol.*

Proof sketch. The final edit precedes its reward, so $\mathcal{M}_T$ is determined by $(V, c_{1:T-1})$. Each code has entropy at most $\ln(2^r - 1)$, from which the information chain rule subtracts the entropy of private rounding, $\mathbb{E}h_2(\rho_t)$. Data processing then gives $I(D_F; \mathcal{M}_T \mid V) \leq \kappa_T$. Information inequalities for bounded utility (Russo & Zou, 2016; Xu & Raginsky, 2017) yield (8) (full proof in Appendix D.4.4).

The bound controls the final bank's expected signed optimism; editor transfer is evaluated separately. At $T = 192$, $r = 4$, and $n = 12$, it reduces to the trivial utility-range bound. Appendix D.4 gives the output-sensitive KL refinement, high-probability certificates, sample complexity, and general-interface lower bound.

5 Experiments

The experiments address four questions about performance, feedback allocation, and how memory improves:

- **RQ1: Does ICME improve performance and transfer?** We evaluate task outcomes, web generalization, and transfer of both editors and banks (Section 5.2).
- **RQ2: How should a limited feedback budget be allocated?** We compare feedback encoders, training costs, and final-bank audit utility (Section 5.3).
- **RQ3: What drives edit gains: content or retrieval?** We separate content, retrieval, and interaction effects (Section 5.4).
- **RQ4: Does learning reduce exposure to risky memory states?** We track state visitation and conditional harm during training (Section 5.5).

5.1 Experimental setup

Benchmarks and metrics. Evaluation covers ALFWorld (Shridhar et al., 2021), ScienceWorld (Wang et al., 2022), WebShop (Yao et al., 2022), AppWorld (Trivedi et al., 2024), and WebArena (Zhou et al., 2024), with Mind2Web (Deng et al., 2023) as an auxiliary offline benchmark. Metrics follow official protocols: success/normalized score, AppWorld task/scenario completion, and Mind2Web element accuracy, action F1, and step/task success. Editor-transfer gain and maximum drawdown additionally assess memory evolution.

Comparison protocol. Core comparisons share a frozen actor and task interface, including ReAct (Yao et al., 2023), ReasoningBank (Ouyang et al., 2026), ACE (Zhang et al., 2026c), and Auto-Dreamer (Ye et al., 2026). Appendix B.3 specifies additional baselines and separate regimes for methods that modify retrieval, tools, or actor parameters.

Backbones and implementation. Experiments use Qwen3-4B, Qwen3-14B, and Qwen2.5-32B-Instruct (Yang et al., 2025; Qwen Team, 2025) on $8\times$NVIDIA A100 80GB GPUs. Only the editor's query/value LoRA adapters (Hu et al., 2022) are trainable. Appendix B details data splits, hyperparameters, memory budgets, and statistical reporting across ten paired runs. Performance comparisons use observed means; intervals describe their uncertainty.

5.2 RQ1: Does ICME improve performance and transfer?

Task performance. ICME achieves the highest means within the matched frozen-actor blocks (Tables 1–2). With Qwen3-14B, ALFWorld unseen and ScienceWorld SR exceed Auto-Dreamer by 3.9 and 5.9 percentage points. On AppWorld, ICME exceeds ACE on both partitions across all backbones, including a 13.5-point challenge-TGC gain with Qwen2.5-32B. The learned editor also improves every reported metric over its fixed counterpart. Rankings cover matched frozen-actor comparisons. Expanded-system results also reflect changes to retrieval, orchestration, or tools.

Web generalization. The ScienceWorld-trained editor improves WebArena SR over ReasoningBank in every website group and backbone (Table 6). On Mind2Web, gains span all three generalization splits (Table 7), although cross-domain task success remains only 1.2–2.5%. Better offline action prediction therefore still leaves substantial room for completing whole tasks.

Table 1: **Interactive task performance.** SR is in percent; score and reward are in $[0, 1]$. Values are mean $\pm$ 95% CI over ten paired runs. Bold and underline mark the best and second-best matched results. Comparison regimes are detailed in Appendix B.7.

		ALFWorld		ScienceWorld		WebShop		WebArena
Model	Method	Seen	Unseen	Score	SR	Reward	SR	SR
Qwen3-4B	ReAct (Yao et al., 2023)	52.1±1.6	45.7±1.7	.442±.014	30.8±1.5	.432±.012	27.4±1.4	26.4±1.3
	ReasoningBank (Ouyang et al., 2026)	59.8±1.5	52.4±1.6	.501±.013	36.2±1.5	<u>.501±.011</u>	<u>33.7±1.4</u>	29.2±1.2
	Auto-Dreamer (Ye et al., 2026)	<u>61.4±1.5</u>	<u>54.1±1.6</u>	<u>.518±.012</u>	<u>37.9±1.4</u>	—	—	31.6±1.2
	ICME (fixed editor)	57.2±1.6	50.0±1.6	.486±.013	34.6±1.5	.482±.011	31.5±1.4	<u>32.4±1.2</u>
	ICME	**65.6±1.4**	**58.4±1.5**	**.556±.012**	**42.1±1.4**	**.548±.010**	**38.8±1.3**	**35.0±1.1**
Qwen3-14B	ReAct (Yao et al., 2023)	61.5±1.5	54.9±1.5	.532±.012	39.6±1.4	.516±.011	35.9±1.3	34.2±1.1
	ExpeL (Zhao et al., 2024)	65.3±1.4	58.1±1.5	—	—	.554±.010	39.8±1.3	—
	AutoGuide (Fu et al., 2024)	66.8±1.4	59.6±1.4	—	—	.569±.010	40.7±1.2	—
	AWM (Wang et al., 2025b)	—	—	—	—	—	—	37.3±1.1
	Synapse (Zheng et al., 2024)	—	—	—	—	—	—	36.0±1.1
	ReasoningBank (Ouyang et al., 2026)	70.6±1.3	64.0±1.4	.602±.011	46.7±1.3	<u>.611±.010</u>	<u>45.6±1.2</u>	41.2±1.0
	Auto-Dreamer (Ye et al., 2026)	<u>72.1±1.3</u>	<u>65.8±1.3</u>	<u>.624±.011</u>	<u>48.9±1.3</u>	—	—	<u>42.0±1.0</u>
	ICME (fixed editor)	67.9±1.4	60.7±1.4	.583±.011	44.0±1.3	.585±.010	42.8±1.2	39.1±1.0
	ICME	**76.4±1.2**	**69.7±1.3**	**.671±.010**	**54.8±1.2**	**.653±.009**	**50.9±1.1**	**46.4±0.9**
	MemRL (Zhang et al., 2026d)	73.8±1.3	66.9±1.3	—	—	—	—	—
	Mem^{2}Evolve (Cheng et al., 2026)	74.6±1.2	67.7±1.3	—	—	.635±.009	48.2±1.1	—
Qwen2.5-32B-Instruct	ReAct (Yao et al., 2023)	66.4±1.4	59.7±1.4	.586±.011	44.3±1.3	.574±.010	41.6±1.2	39.0±1.0
	ReasoningBank (Ouyang et al., 2026)	74.0±1.3	67.3±1.3	.651±.010	51.2±1.2	<u>.634±.009</u>	<u>48.9±1.1</u>	44.3±0.9
	Auto-Dreamer (Ye et al., 2026)	<u>75.5±1.2</u>	<u>68.8±1.3</u>	<u>.668±.010</u>	<u>53.0±1.2</u>	—	—	<u>45.2±0.9</u>
	ICME (fixed editor)	71.5±1.3	64.1±1.3	.628±.010	48.6±1.2	.611±.009	46.7±1.1	42.3±0.9
	ICME	**79.2±1.1**	**72.8±1.2**	**.704±.009**	**57.5±1.1**	**.679±.008**	**54.1±1.0**	**49.8±0.8**

Table 2: **AppWorld task and scenario completion (%).** Values are mean $\pm$ 95% CI over ten paired runs. Bold and underline mark the best and second-best frozen-actor results within each model; SAGE updates the actor and is excluded from this ranking. Protocol details are in Appendix B.7.

		Test-normal (168)		Test-challenge (417)	
Model	Method	TGC ↑	SGC ↑	TGC ↑	SGC ↑
Qwen3-4B	ReAct (Yao et al., 2023)	51.8±2.1	35.4±2.4	31.6±1.8	17.9±1.9
	ACE (Zhang et al., 2026c)	<u>58.6±2.0</u>	<u>43.2±2.3</u>	<u>39.4±1.9</u>	<u>23.5±2.0</u>
	ICME (fixed editor)	56.2±2.0	40.7±2.2	36.8±1.8	21.2±1.9
	ICME	**64.9±1.8**	**49.6±2.1**	**46.7±1.8**	**29.8±1.9**
Qwen3-14B	ReAct (Yao et al., 2023)	59.4±1.9	42.1±2.2	39.7±1.8	23.0±1.9
	DC-CU (Suzgun et al., 2025)	63.8±1.8	49.3±2.1	46.1±1.8	29.0±1.9
	ACE (Zhang et al., 2026c)	66.7±1.8	51.8±2.0	50.4±1.7	31.7±1.8
	MemEvolve (Zhang et al., 2026a)	<u>68.0±1.7</u>	<u>53.2±2.0</u>	<u>51.6±1.7</u>	<u>32.8±1.8</u>
	ICME (fixed editor)	64.5±1.8	48.2±2.0	45.9±1.7	28.6±1.8
	ICME	**72.8±1.6**	**58.9±1.9**	**57.6±1.6**	**38.3±1.7**
Qwen2.5-32B-Instruct	ReAct (Yao et al., 2023)	64.8±1.8	46.6±2.1	44.5±1.7	27.4±1.8
	ACE (Zhang et al., 2026c)	<u>70.3±1.7</u>	<u>55.7±1.9</u>	<u>54.0±1.6</u>	<u>34.8±1.7</u>
	ICME (fixed editor)	68.1±1.7	52.3±1.9	50.8±1.6	31.8±1.7
	ICME	**77.1±1.5**	**65.0±1.8**	**67.5±1.5**	**50.6±1.6**
	SAGE (Wang et al., 2026)	74.0±1.6	61.2±1.8	56.2±1.5	38.0±1.7

Editor and bank transfer. Starting from empty banks, learned editors improve J_{32} over fixed editors by .048–.085, with higher AUC and lower maximum drawdown in all nine conditions, including three cross-domain transfers (Table 8a). Separately, with writing disabled, learned banks improve query utility by .063–.084 over token-matched fixed-editor banks (Table 8b). Thus, benefits persist both when rebuilding memory from fresh experience and when reusing stored content.

Takeaway (RQ1). A memory system has two reusable assets: stored experience and the policy that curates future experience. Empty-bank transfer tests the second without inheriting the first.

5.3 RQ2: How should a limited feedback budget be allocated?

On Qwen3-14B and ScienceWorld, we examine feedback encoding, budget allocation, and final-bank utility on an independent audit set.

Feedback encoding at a fixed horizon. At $T = 768$, stochastic feedback with $r = 2, 4, 8$ yields mean J_{32} of .607, .636, and .645, respectively, compared with .650 for full precision and .556 for the fixed editor (Table 9). At $r = 4$, nearest-grid rounding reaches only .619. Greater precision improves the observed utility at equal update count. The rounding comparison measures the combined effect of changes in bias and variance.

Precision versus update count at a fixed budget. With $B = 3072$ bits, $r = 2, 4, 8$ permit $T = 1536, 768, 384$ updates (Figure 4). The default $r = 4$ has the highest mean J_{32} and AUC and the lowest mean MDD. Relative to $r = 2$, it reduces training time from 27.7 to 14.6 hours (47.3%) while increasing mean J_{32} from .630 to .636; $r = 8$ takes 7.9 hours but reaches .624. The best observed allocation balances feedback precision with the number of updates. Table 10 also reports environment episodes and tokens.

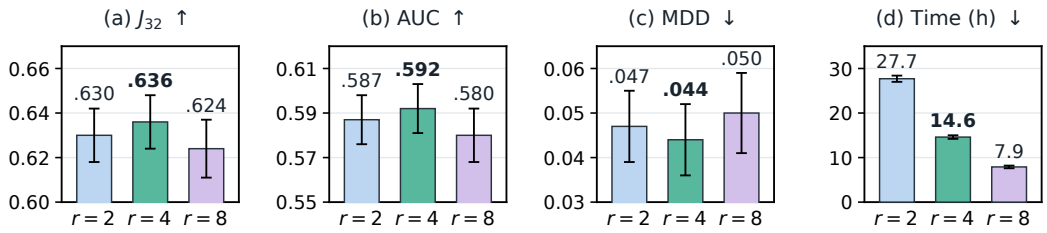

Figure 4: **Equal feedback budgets entail different training costs.** Qwen3-14B on ScienceWorld with $B = 3072$ bits: $r = 2, 4, 8$ correspond to $T = 1536, 768, 384$ updates. Bars show means with 95% CIs over ten paired runs; green denotes the default $r = 4$. Panels (a)–(c) use truncated vertical axes. Full results, including environment episodes and token counts, are in Appendix C.4, Table 10.

Final-bank utility and feedback reuse. The fixed-horizon banks' feedback–audit gaps increase from .013 to .021 to .026 as r rises, alongside audit utility of .607, .632, and .640 (Table 11). Full precision attains .642 audit utility with a .033 gap. A smaller gap can therefore accompany lower utility. These audit measurements assess the final bank under feedback reuse.

Takeaway (RQ2). Feedback compression is worthwhile only when useful learning survives it. A smaller feedback–audit gap should therefore be evaluated together with retained audit utility and the computation needed to obtain it.

5.4 RQ3: What drives edit gains: content or retrieval?

Separating the pathways. On ScienceWorld with Qwen3-14B, we compare the frozen editor (Fixed), the quarter-training checkpoint (ICME-Early), and the final editor (ICME-Late), using 320 edits per state on held-out queries. Four replays cross pre/post-edit content with pre/post-edit retrieval identities, separating content, retrieval, and interaction effects. The content pathway holds retrieval identities fixed and includes length-dependent context packing; interaction captures the joint, non-additive effect (Appendix B.9).

Content gains and retrieval disruption. The centroid's content coordinate rises from .293 to .413 (Figure 5). Content-dominant helpful edits increase from 18.8% to 39.7%, while retrieval-dominant harmful edits fall from 24.1% to 11.6%. Mean content gain rises from .00573 to .02075, the largest component increase; retrieval gain changes from −.00420 to .00045 and interaction from −.00152 to .00488 (Table 12). As mean query gain rises from .00001 to .02608, the improvement combines stronger content effects with reduced retrieval-related harm.

Takeaway (RQ3). Optimizing a memory bank requires preserving access to useful knowledge as well as improving its contents. Fixed-identity replay provides a check against mistaking retrieval reshuffling for better writing.

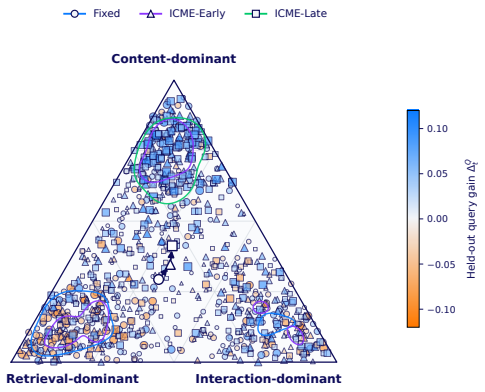

Figure 5: **Learning shifts edit gains toward content.** Points show normalized absolute content, retrieval, and interaction effects. Color gives query gain Δ_t^Q (orange: negative; blue: positive); area increases with $|\Delta_t^Q|$. Shapes and colored contours identify editor states. White centroids and arrows trace training progression.

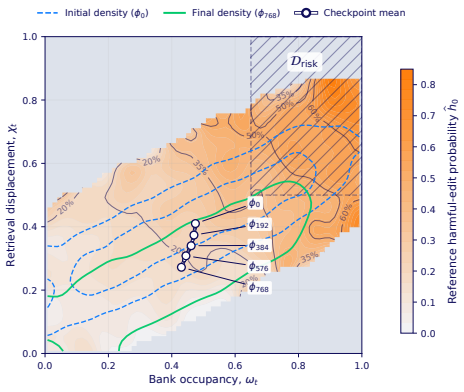

Figure 6: **ICME reduces visits to high-competition states.** Orange shows initial harmful-edit probability; thin contours mark equal risk, and gray cells lack support. Hatching marks high competition. Blue dashed and green solid contours show initial and final visitation. White centroids and arrows trace checkpoints.

5.5 RQ4: Does learning reduce exposure to risky memory states?

Tracking exposure on a fixed reference surface. On ScienceWorld with Qwen3-14B, we freeze the editor at training edits $0, 192, 384, 576, 768$ and evaluate 24 diagnostic streams of 32 events over ten paired seeds: 7,680 events per checkpoint. We track bank occupancy ω_t, aggregate retrieval displacement χ_t, and diagnostic gain Δ_t. The reference harmful-edit surface is estimated from ϕ_0 and held fixed; high competition uses the preliminary thresholds $\omega_t > 0.65$ and $\chi_t > 0.50$ (Appendix B.10). Using the same surface and support at every checkpoint makes exposure comparable within a common reference domain, where changes reflect shifts in the states visited by the editor.

Less exposure, persistent conditional risk. High-competition visitation decreases monotonically from 22.3% to 4.8%, while exposure to the fixed reference surface falls from .285 to .203 over supported states (Figure 6). Mean retrieval displacement decreases from .410 to .272, while mean occupancy changes more modestly, from .475 to .430. The supported domain covers 97.6%–99.9% of events at each checkpoint, so the exposure comparison includes nearly all diagnostic events. Overall harm falls from 28.5% to 20.5%, yet harm within the high-competition region stays between 48.2% and 51.3% (Table 13). Together, these observations are consistent with reduced exposure to fragile states, where conditional risk remains high.

Takeaway (RQ4). Lower aggregate harm can arise from avoiding fragile memory states even when those states remain risky. Robustness evaluation should therefore track where an editor operates as well as how often its edits fail.

6 Conclusion

This work addressed the joint challenge of learning useful memory edits and controlling exposure to repeatedly reused feedback. With the acting model fixed, ICME learns from the effects of edits on the entire memory bank through an explicitly limited feedback channel. Improved performance in matched task comparisons and gains over fixed editors across all nine transfer settings support memory editing as a reusable capability. Together with our analysis of feedback capacity, these findings motivate designing memory systems around both the consequences of their edits and the information used to learn them. An open direction is to determine how editors should allocate a fixed feedback budget across task streams as their demands evolve.

AI USE STATEMENT

We used generative AI tools to assist with organizing and interpreting experimental results, polishing the language, and formatting the manuscript. The authors take full responsibility for all text and results in this work, including content prepared with AI assistance.

REFERENCES

Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. QSGD: Communication-efficient SGD via gradient quantization and encoding. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://arxiv.org/abs/1610.02132.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-Embedding: Multi-Linguality, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. *arXiv preprint arXiv:2402.03216*, 2025. URL https://arxiv.org/abs/2402.03216.

Zihao Cheng, Zeming Liu, Yingyu Shan, Xinyi Wang, Xiangrong Zhu, Yunpu Ma, Hongru Wang, Yuhang Guo, Wei Lin, and Yunhong Wang. Mem^{2}Evolve: Towards Self-Evolving Agents via Co-Evolutionary Capability Expansion and Experience Distillation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2026. URL https://aclanthology.org/2026.acl-long.952/.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samuel Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2Web: Towards a Generalist Agent for the Web. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2306.06070.

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. Generalization in adaptive data analysis and holdout reuse. *arXiv preprint arXiv:1506.02629*, 2015. URL https://arxiv.org/abs/1506.02629.

Yao Fu, Dong-Ki Kim, Jaekyeom Kim, Sungryull Sohn, Lajanugen Logeswaran, Kyunghoon Bae, and Honglak Lee. AutoGuide: Automated generation and selection of context-aware guidelines for large language model agents. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-3811. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/d8efbb5dd415974eb095c3f06bff1f48-Abstract-Conference.html.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2106.09685.

Qinfeng Li, Yuntai Bao, Xinyan Yu, Hongze Chen, Yanming Liu, Huifeng Zhu, Yier Jin, Jintao Chen, Wenqi Zhang, and Xuhong Zhang. AttriMem: Attribution-guided process feedback for agent memory construction. *arXiv preprint arXiv:2607.21106v3*, 2026. URL https://arxiv.org/html/2607.21106v3.

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. ReasoningBank: Scaling agent self-evolving with reasoning memory. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=jL7fwchScm.

Qwen Team. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115*, 2025. URL https://arxiv.org/abs/2412.15115.

Daniel Russo and James Zou. Controlling bias in adaptive data analysis using information theory. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pp. 1232–1240. PMLR, 2016. URL https://proceedings.mlr.press/v51/russo16.html.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-0377. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

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. URL https://arxiv.org/abs/2010.03768.

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. URL https://arxiv.org/abs/2504.07952.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. AppWorld: A Controllable World of Apps and People for Benchmarking Interactive Coding Agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024. URL https://aclanthology.org/2024.acl-long.850/.

Jiongxiao Wang, Qiaojing Yan, Yawei Wang, Yijun Tian, Soumya Smruti Mishra, Zhichao Xu, Megha Gandhi, Panpan Xu, and Lin Lee Cheong. Reinforcement learning for self-improving agent with skill library. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1529–1550. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.69. URL https://aclanthology.org/2026.acl-long.69/.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. ScienceWorld: Is your Agent Smarter than a 5th Grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022. URL https://aclanthology.org/2022.emnlp-main.775/.

Yu Wang, Ryuichi Takanobu, Zhiqi Liang, Yuzhen Mao, Yuanzhe Hu, Julian McAuley, and Xiaojian Wu. Mem-α: Learning memory construction via reinforcement learning. *arXiv preprint arXiv:2509.25911*, 2025a. URL https://arxiv.org/abs/2509.25911.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent Workflow Memory. In *Proceedings of the International Conference on Machine Learning*, 2025b. URL https://proceedings.mlr.press/v267/wang25bx.html.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992. doi: 10.1007/BF00992696. URL https://doi.org/10.1007/BF00992696.

Aolin Xu and Maxim Raginsky. Information-theoretic analysis of generalization capability of learning algorithms. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/ad71c82b22f4f65b9398f76d8be4c615-Abstract.html.

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian Kersting, Jeff Z. Pan, Hinrich Schütze, Volker Tresp, and Yunpu Ma. Memory-R1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12805–12825. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.583. URL https://aclanthology.org/2026.acl-long.583/.

An Yang, Anfeng Li, Baosong Yang, et al. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Yi Yang, Zhennan Chen, Yihong Zhuang, Tiehan Fan, Yinan Chen, Jian Li, Jian Yang, and Ying Tai. RoMeRL: Balancing feedback coverage and the memory-reward trap in self-evolving agent memory via reduced-order utility states. *arXiv preprint arXiv:2608.02508*, 2026. URL https://arxiv.org/abs/2608.02508.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. WebShop: Towards Scalable Real-World Web Interaction with Grounded Language Agents. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2207.01206.

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. URL https://arxiv.org/abs/2210.03629.

Chongrui Ye, Yuxiang Liu, Yu Wang, Haofei Yu, Yining Zhao, Ge Liu, Julian McAuley, and Jiaxuan You. Auto-Dreamer: Learning Offline Memory Consolidation for Language Agents. *arXiv preprint arXiv:2605.20616*, 2026. URL https://arxiv.org/abs/2605.20616.

Guibin Zhang, Haotian Ren, Chong Zhan, Zhenhong Zhou, Junhao Wang, He Zhu, Wangchunshu Zhou, and Shuicheng Yan. MemEvolve: Meta-Evolution of Agent Memory Systems. In *Proceedings of the International Conference on Machine Learning*, 2026a. URL https://arxiv.org/abs/2512.18746.

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*, 2026b. URL https://arxiv.org/abs/2602.02474.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, Urmish Thakker, James Zou, and Kunle Olukotun. Agentic context engineering: Evolving contexts for self-improving language models. In *International Conference on Learning Representations*, 2026c. URL https://arxiv.org/abs/2510.04618.

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

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*, 2024. URL https://ojs.aaai.org/index.php/AAAI/article/view/29936.

Longtao Zheng, Rundong Wang, Xinrun Wang, and Bo An. Synapse: Trajectory-as-Exemplar Prompting with Memory for Computer Control. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2306.07863.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A Realistic Web Environment for Building Autonomous Agents. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2307.13854.

Appendix Contents

A Technical Details 14
- A.1 Memory representation and editor interface 14
- A.2 Editor parameterization and sampling 15
- A.3 Deterministic paired evaluation and training loop 15

B Experimental Details 17
- B.1 Benchmark coverage and scoring 17
- B.2 Data separation and transfer 17
- B.3 Comparison regimes and baseline selection 18
- B.4 Backbones, implementation, and resource settings 19
- B.5 Evaluation of the learned editor and the feedback channel 20
- B.6 Mechanism-specific metrics and statistical reporting 20
- B.7 Result sources and comparison protocols 21
- B.8 Preliminary study: protocol and diagnostics 22
- B.9 Content and retrieval: replay and aggregation protocol 23
- B.10 Retrieval competition: risk and visitation protocol 23

C Additional Experimental Results 24
- C.1 Preliminary study: supplementary results 24
- C.2 Web and offline-action breakdowns 25
- C.3 Editor transfer versus frozen-bank transfer 27
- C.4 Feedback precision, budget, and final-bank audit 27
- C.5 Content and retrieval decomposition 29
- C.6 Retrieval-competition risk and visitation 29

D Proofs 30
- D.1 Rounding law and the small-gain regime 30
- D.2 Local gradient and covariance 30
- D.3 Information capacity and expected feedback bias 30
- D.4 Adaptive-feedback generalization 31

A TECHNICAL DETAILS

A.1 MEMORY REPRESENTATION AND EDITOR INTERFACE

Entries and serialization. An entry has fields `id`, `when`, `strategy`, and `src`. The first is an integer identifier; the next two are nonempty strings; `src` identifies the visible episode that produced the entry. Normalize strings to Unicode NFC and strip leading and trailing whitespace. Store entries in increasing ID order. The editor receives a compact JSON object with keys `next_id` and `entries`; the latter is an array with entry fields in the order above, UTF-8 strings, standard JSON escaping, and no insignificant whitespace. The protocol assigns source pointers from the visible episode's stream and round indices.

Actor-facing content for one entry is the literal string `When: <when>` followed by a newline and `Strategy: <strategy>`. Separate entries by two newlines. A single frozen actor tokenizer defines ℓ, counting all content delimiters. Each entry must fit $\ell_{\max}$ tokens, the full bank must fit C tokens, and retrieved content must fit L tokens. IDs and source pointers are editor metadata; their prompt tokens count toward total model-token use in the matched cost comparison.

Retrieval. Let f_{emb} be the frozen embedding model. Score each entry by the cosine similarity between $f_{\text{emb}}(x)$ and the embedding of its actor-facing content; assign score zero when an embedding has zero norm. Sort by decreasing score, breaking ties by increasing ID. Examine only the first $\min\{k, |\mathcal{M}|\}$ entries. Append an entry exactly when the resulting serialized context fits L; skip an entry that does not fit. There is no truncation of retrieved entries and no backfilling below rank k. Retrieve once at task start, hold the bank read-only during that episode, and discard task-local state at its end. Both visible and feedback executions use this rule.

Patch schema and state transition. The four patch forms are

```
{"op":"ADD","when":"...","strategy":"..."}
{"op":"REPLACE","id":7,"when":"...","strategy":"..."}
{"op":"DELETE","id":7}
{"op":"KEEP"}
```

The displayed ID is illustrative; a target ID must exist in the current bank. Require exactly the fields prescribed by the selected operation, correct field types, and one complete JSON object. ADD receives the next unused ID after all validity and capacity checks pass. The next-ID counter is stored as bank metadata, increases only for valid additions, and is retained across stream resets. REPLACE preserves the target ID and sets its source pointer to the current visible episode. If both normalized content strings are unchanged, preserve the entire old entry, including its source pointer. DELETE removes its target; KEEP changes nothing. Every malformed, incomplete, invalid-reference, empty-content, and over-capacity patch maps to the unchanged bank. No regeneration, automatic eviction, reward-dependent rejection, or background consolidation occurs.

Fixed editor prompt. The editor receives the following system instruction. Its user message supplies the serialized JSON bank and experience, represented below by `<bank>` and `<experience>`.

> You edit a procedural memory bank for an agent. Use only the memory and visible experience below. Write lessons that can help future tasks. Choose exactly one operation: ADD, REPLACE, DELETE, KEEP. ADD and REPLACE require nonempty when and strategy strings. REPLACE and DELETE require an existing integer id. KEEP changes nothing. Return exactly one JSON object using only the fields required by your operation. Do not include explanations, markdown, source pointers, or new identifiers.
>
> `MEMORY_JSON: <bank>`
> `EXPERIENCE_JSON: <experience>`

The operation schemas above are appended verbatim to the system message, replacing the illustrative ID by the phrase `existing integer id`. Experience has keys `episode`, `outcome`, `task`, and `trajectory`, in that order. The episode field contains the stream and round indices. The next two fields contain visible text; the last is an array of action–observation text records. Each

text record has fields `text` and `truncated`, with the latter initially false. It contains no feedback-set reward. The score-function state s_t is the resulting model input, including the fixed chat template.

Deterministic truncation. Let H be the editor's context limit minus its maximum generated length $m_{\max}$. Reserve the complete system message, complete memory object, delimiters, and the episode field with empty text records; configurations in which this exceeds H are invalid. Add experience fields in this priority order: native outcome, task instruction, then action–observation records from newest to oldest. For each field, keep its complete text if the full serialized prompt fits H. Otherwise keep the longest prefix of its editor-token sequence whose decoded text and JSON delimiters fit; mark that field as truncated and stop adding fields. If even an empty marked field does not fit, omit that field and stop. Restore retained trajectory records to chronological order before serialization. Memory entries are never truncated in the editor input. The length check includes the truncation marker and final chat template.

A.2 Editor parameterization and sampling

Only the editor's attention query/value projections receive LoRA adapters (Hu et al., 2022). For projection l,

$$W_\phi^{(l)} = W_0^{(l)} + A_l B_l^\top, \qquad A_l \in \mathbb{R}^{d_l^{\text{out}} \times d}, \quad B_l \in \mathbb{R}^{d_l^{\text{in}} \times d}. \tag{9}$$

Here $W_0^{(l)}$ is frozen, d is the adapter rank, and $\phi = \{A_l, B_l\}_l$ comprises all trainable parameters. Set $A_l = 0$ and draw the entries of B_l independently from $\mathcal{N}(0, 1/d_l^{\text{in}})$ with a prespecified initialization seed. There is no additional adapter scaling coefficient. The actor is a separate frozen instance, including when the two models share a base checkpoint.

Sample from the editor's full autoregressive softmax at temperature one, with dropout disabled and maximum length $m_{\max}$. If $p = (p_1, \ldots, p_m)$ is the raw sampled sequence, its log-probability is

$$\log q_\phi(p \mid s) = \sum_{j=1}^{m} \log q_\phi(p_j \mid s, p_{<j}). \tag{10}$$

The sum includes sampled termination tokens. A sequence reaching the length limit without termination is incomplete and leaves the bank unchanged; no synthetic termination token enters its likelihood. Parsing does not change the sampled sequence used for the gradient. The finite token limit gives a parameter-independent support. Compute each gradient under the same parameters and sampling rule that generated its patch. Apply one SGD ascent step with the fixed learning rate η, without momentum, weight decay, clipping, a KL penalty, replay, or token-length normalization. At decoded reward zero, the exact update is zero and backpropagation can be skipped.

A.3 Deterministic paired evaluation and training loop

Scoring and cache. Use one deterministic rollout per bank and feedback task, restoring its read-only snapshot and environment seed zero. Actor decoding is greedy with token-ID tie breaking; tools and the model backend are deterministic. The benchmark's native score, normalized by its published range to $[0, 1]$, defines u. Memory is read-only during evaluation. Native task failures, invalid actions, and exhaustion of the benchmark's action limit receive zero. Tasks requiring uncontrolled external state are excluded from this replay protocol.

The evaluator-private cache is keyed by the exact serialization of ordered entry IDs and actor-facing content, task ID, and evaluator configuration. The key includes all bank state that affects actor execution, excluding source pointers and the next-ID counter. A miss triggers one rollout; a hit returns the stored score. Entries are never evicted. Score the empty bank once and reuse its vector at all stream boundaries. This policy is independent of reward signs.

A host interruption triggers one retry of the affected rollout from the identical snapshot, without resampling the edit. A second host interruption invalidates the run; it is recorded as a failed run, with no replacement score and no selection of a successful checkpoint. Failed runs remain in the run inventory. The theoretical analysis uses the unconditional law of the specified fixed-horizon protocol.

Algorithm 1 ICME with direct memory transitions

Require: Frozen actor and execution protocol; initial editor ϕ_0; visible tasks $x_{1:T}$; reset rounds $\mathcal{B}$ with $1 \in \mathcal{B}$; isolated D_F; memory limits; $\eta > 0$; integer $r \geq 2$ with $Tr \leq B_{\max}$.

$\phi \leftarrow \phi_0$; initialize empty $\mathcal{M}$ with next ID 1; $B \leftarrow 0$; $K \leftarrow 2^{r-1} - 1$; $h \leftarrow 1/K$.
Evaluator caches the score vector for the empty bank.
for $t = 1, \ldots, T$ **do**
  if $t \in \mathcal{B}$ **then**
    Clear memory entries and stream-local experience; retain the next-ID counter.
  end if
  $\mathcal{M}^- \leftarrow \mathcal{M}$; execute x_t with $\mathcal{M}^-$ to obtain e_t.
  $s \leftarrow \mathrm{Serialize}(\mathcal{M}^-, e_t)$; sample $p \sim q_\phi(\cdot \mid s)$.
  $\mathcal{M} \leftarrow \mathrm{Apply}(\mathcal{M}^-, p)$; install its retrieval index.
  Evaluator obtains cached score vectors, executing cache misses.
  Internally compute $\Delta \leftarrow \widehat{J}_F(\mathcal{M}) - \widehat{J}_F(\mathcal{M}^-)$.
  $\xi \leftarrow K(\Delta + 1)$; $j \leftarrow \min\{2K - 1, \lfloor \xi \rfloor\}$; $\rho \leftarrow \xi - j$.
  Draw $\beta \sim \mathrm{Bernoulli}(\rho)$; release only $c \leftarrow j + \beta$.
  $\widehat{\Delta} \leftarrow h(c - K)$; $B \leftarrow B + r$.
  $\phi \leftarrow \phi + \eta \widehat{\Delta} \nabla_\phi \log q_\phi(p \mid s)$.
end for
Freeze training and return $(\phi, \mathcal{M})$.

Information isolation. Only the fixed-length code c_t crosses from evaluator to learner. Feedback tasks, trajectories, score vectors, retrieval logs, caches, and mutable files remain in the evaluator. The editor input is restricted to the bank and visible experience. A round-synchronous interface hides cache-hit status, execution duration, and other metadata. Diagnostic logs are opened only after all training decisions are locked. Design choices, task order, stream boundaries, and the final round T are fixed without accessing D_F. Prescheduled stream resets clear memory but retain ϕ, the global ID counter, and the bit ledger. The output is the final pair $(\phi_T, \mathcal{M}_T)$, without reward-based checkpoint selection.

Cost. Without infrastructure retries, training uses T visible episodes and at most $(T+1)n$ feedback episodes: n for the empty bank and at most n for each newly encountered bank. Resets incur no additional empty-bank evaluation. There are T editor samples and at most T backward passes. At $T = 192$ and $n = 12$, this bound is 2508 per condition and seed. Retry attempts, audit, and transfer evaluation add to this count. Episode counts exclude editor-generation and backward-pass costs; the matched fixed-bit experiment also records total model-token use and training wall time.

B EXPERIMENTAL DETAILS

B.1 BENCHMARK COVERAGE AND SCORING

Table 3 specifies the evaluation suite. Evaluation uses official scoring code and task-ID manifests. Scores from different partitions, action interfaces, or numbers of attempts are reported separately.

Table 3: Benchmark coverage. SR denotes strict task success rate. The listed action caps apply to all methods in our controlled comparisons.

Benchmark	Evaluation partition and interface	Reported native metrics
ALFWorld	Text interface; all six families; 140 seen and 134 unseen evaluation tasks, reported separately; 50 actions.	Task-weighted SR.
ScienceWorld	Official test variations across all 30 task types; fixed simplification profile; 100 actions.	Mean official score divided by 100; strict SR.
WebShop	Official held-out test instructions; text interface and fixed product catalogue; 15 actions.	Mean reward in $[0, 1]$ and strict SR (reward $= 1$).
AppWorld	Official test-normal (168 tasks) and test-challenge (417); sandboxed APIs; 30 agent turns.	Task goal completion (TGC), scenario goal completion (SGC).
WebArena	Self-hosted snapshots; ReasoningBank's 684 non-Maps tasks and website groups; accessibility-tree text; 30 actions.	Task-weighted SR, per-website SR, mean actions on all tasks.
Mind2Web	Official cross-task, cross-website, and cross-domain partitions; fixed DOM and candidate-element ranking.	Element accuracy, action F1, step SR, task SR.

Matching published protocols. ReasoningBank evaluates 684 non-Maps WebArena tasks, whereas Auto-Dreamer evaluates 117 tasks from three website families and a 660-task, seven-type ScienceWorld subset (Ouyang et al., 2026; Ye et al., 2026). Our controlled runs use the full suites specified in Table 3. The controlled ScienceWorld configuration enables `teleportAction`, `openDoors`, and `selfWateringFlowerPots` for every method. Mind2Web comparisons share frozen candidate lists and observation preprocessing.

Native scores and training utility. Benchmark tables retain official partial-credit scores, including terminal scores at the action cap. The private utility u follows Appendix A.3: normalized native score, with zero for the failure conditions specified there. Benchmark performance and query utility are reported in separate tables. ScienceWorld reports partial progress separately from strict success, and AppWorld SGC requires all tasks in a scenario to succeed. SR denotes the first attempt. Mind2Web is an offline action-prediction diagnostic, evaluated separately from interactive feedback episodes.

B.2 DATA SEPARATION AND TRANSFER

The protocol separates visible experience tasks E, design tasks D_P, private feedback tasks D_F, an audit set D_A, and final evaluation tasks before collecting trajectories. Within training data, all paraphrases and reset variants of the same task instance share one split; AppWorld uses whole scenarios. The split manifest records grouping keys and shared environment assets. The official train/test partitions can share broad task families; we preserve this overlap. Official development data serve as D_P where available; otherwise, 10% of training groups are reserved for design. From the remaining training groups, 10% are reserved for D_A, a disjoint collection of complete groups containing at least 12 tasks forms D_F, and the rest form E. Thus $n = |D_F|$ may exceed 12 to avoid splitting a group; the split manifest specifies the realized n. No official evaluation tasks are used for parameter tuning or private feedback, and any duplicate task instances crossing an official boundary are removed from training.

AppWorld requires scenario-level separation: its 105 training tasks comprise 35 scenarios, and its 60 development tasks comprise 20 scenarios. We reserve four training scenarios (12 tasks) for

D_F, four for D_A, and use the remaining 27 (81 tasks) for visible experience; the official development scenarios provide D_P. Unique-task and episode counts distinguish repeated passes through a small experience pool. Final normal/challenge sets remain sealed. For ALFWorld, the named `valid_seen` and `valid_unseen` splits are reserved for final evaluation.

In-domain and cross-domain evaluation. Separate editors are trained on ALFWorld, ScienceWorld, WebShop, and AppWorld training partitions. To test procedural transfer, the ScienceWorld-trained editor is also deployed on ALFWorld and WebArena without target-domain parameter updates, following the transfer question posed by Auto-Dreamer (Ye et al., 2026). WebArena is used only for deployment evaluation. Its persistent website state is restored to the prescribed snapshot before every task. External live services are outside the deterministic feedback protocol.

Two distinct transfer objects. For *editor transfer*, both ϕ_T and the initialization ϕ_0 start from empty banks on identical fresh experience streams. Parameters stay frozen, and neither editor accesses D_F. Banks are evaluated read-only on separate query tasks at editing rounds $0, 8, 16, 24, 32$. For each final evaluation pool, five fixed group folds rotate the query fold; the other four supply up to 32 distinct experience tasks in a prespecified order. Every query task is scored only with a bank that has never encountered its group. Query observations and scores are never returned to the deployed editor. Each fold starts a separate bank without sharing memory state; the fold assignment is frozen before evaluation. For AppWorld the grouping unit remains the scenario, including when computing SGC.

For *memory transfer*, the final training bank $\mathcal{M}_T$ is inserted into a fresh frozen actor and evaluated with all writing disabled, against empty memory and a token-matched bank from the fixed editor. Each domain uses its own bank. A separate online benchmark pass measures first-attempt predict-then-update SR in the same fixed task order for all methods: only earlier visible episodes may inform later memory. These online results, held-out-query summaries, and frozen-bank results answer different questions and occupy separate panels.

B.3 Comparison regimes and baseline selection

Frozen-actor memory comparisons. The main suite compares ReAct without cross-task memory (Yao et al., 2023), ExpeL (Zhao et al., 2024), AutoGuide (Fu et al., 2024), AWM (Wang et al., 2025b), ReasoningBank (Ouyang et al., 2026), ACE (Zhang et al., 2026c), and Auto-Dreamer (Ye et al., 2026). ExpeL and AutoGuide are relevant on ALFWorld/WebShop; AWM and Synapse (Zheng et al., 2024) on WebArena/Mind2Web; ACE and Dynamic Cheatsheet in cumulative mode (DC-CU) (Suzgun et al., 2025) on AppWorld; and Auto-Dreamer on ALFWorld/ScienceWorld/WebArena. This selection follows the comparison tables of ReasoningBank (AWM, Synapse), ACE (ReAct, Dynamic Cheatsheet), and Auto-Dreamer (including ExpeL, AWM, and ReasoningBank). Each method is evaluated on these overlapping benchmarks; further ports are labeled as adaptations.

The shared-actor comparison preserves each baseline's defining memory representation, retrieval, and consolidation mechanism. The actor checkpoint, environment, observable outcomes, task order, and action budget are identical; auxiliary writers, reflectors, and curators start from the same base checkpoint within a row. Trajectory-memory baselines retain their native entry schemas. Controlled comparisons use bank capacity $C = 1024$ and injected-context budget $L = 512$, preserving native entry schemas within these limits. Truncation and capacity adaptations are documented; configurations unable to represent a valid item are marked infeasible. Only the ICME ablations hold representation and retrieval exactly fixed, supporting attribution specifically to editor learning or reward encoding.

Expanded-system and actor-training comparisons. MemRL (Zhang et al., 2026d) is evaluated on ALFWorld with its utility-aware retrieval intact. MemEvolve (Zhang et al., 2026a) (ICML 2026) forms a separately labeled AppWorld adaptation retaining its memory-architecture search; its native web/research benchmarks provide context for this adaptation. Mem^{2}Evolve (Cheng et al., 2026) (ACL 2026) is compared on ALFWorld/WebShop with its asset creation enabled. These systems evaluate memory together with changes to retrieval, orchestration, or tools. Newly generated tools may compose only the shared primitive APIs, and their internal primitive calls count toward the ac-

tion cap. SAGE (Wang et al., 2026) (ACL 2026) is an AppWorld actor-training comparison because its SFT/RL updates the acting policy. Its initial checkpoint and training data access are matched; it is excluded from the frozen-actor ranking.

Attempts, feedback, and compute. Controlled task results report first-attempt SR from one trajectory per task. Resource reporting is restricted to the matched fixed-bit experiment in Appendix C.4: total training environment episodes, total model input/output tokens, and elapsed training time on the same eight-GPU allocation. Counts include visible experience, evaluator cache misses, and infrastructure retries; tokens include actor, editor, and evaluator calls. Wall time includes generation, evaluation, and adapter updates. Audit and deployment evaluation are outside this training-cost comparison. Interaction and compute costs vary even at equal feedback capacity.

All methods may use the same native outcomes on visible tasks. In the common reward-access regime, trainable memory baselines may query a private scalar-reward service on the same D_F, with their required candidate/group evaluations charged and its task text, traces, and labels kept private. ICME receives only its r-bit code. Methods requiring richer supervision occupy a separate access regime. The matched full-precision ICME control changes only the feedback encoder. Published API-model results appear in separate reference rows; reported improvements use matched open-backbone runs.

B.4 Backbones, implementation, and resource settings

Qwen3-14B overlaps Auto-Dreamer's writer/consolidator, Qwen2.5-32B-Instruct overlaps SAGE's actor, and Qwen3-4B overlaps AttriMem's memory model (Ye et al., 2026; Wang et al., 2026; Li et al., 2026). These open, trainable backbones support controlled comparisons with shared actors. Each controlled run uses separate actor/editor instances of one base checkpoint. Qwen3 thinking mode is disabled, and the actor decodes greedily. Training uses BF16 frozen weights, FP32 adapter master parameters and SGD accumulation, gradient checkpointing, and model sharding as needed on eight A100 80GB GPUs. The reproducibility manifest specifies software versions and sharding layouts.

Table 4: ICME configuration. Hyperparameters are selected on D_P and fixed before feedback training.

Component	Setting
Training horizon	$T = 768$: 24 streams of 32 edits, resetting only memory.
Private feedback	Target $n = 12$ tasks, enlarged to preserve whole groups; actual size recorded in the split manifest.
Reward encoder	$r = 4$ ($K = 7$, 15 levels), exact zero, stochastic rounding; matched controls use $r \in \{2, 4, 8\}$.
Bank and retrieval	$C = 1024$, $\ell_{\max} = 128$, $k = 4$, $L = 512$ actor tokens; frozen BGE-M3 dense embeddings (Chen et al., 2025), cosine ranking.
LoRA	Query/value projections in all attention layers; rank $d = 8$; unit effective scaling, zero dropout.
Update	One sequence-level SGD ascent step per edit; $\eta = 10^{-6}$.
Editor decoding	Temperature 1; full softmax; no top-k/top-p truncation; $m_{\max} = 256$ sampled tokens, including termination.
Context and actions	32,768-token editor context, 16,384-token actor context; actor maximum 1,024 generated tokens per turn; fixed observation truncation, benchmark caps in Table 3.
Evaluation randomness	Environment seed 0 for paired replay; ten paired seeds (0–9) for initialization, editing, and task order; separate RNG streams for rounding.

The LoRA initialization, full-sequence likelihood, invalid-patch behavior, cache, and direct-application rule are exactly those in Appendix A; there is no reward gate or automatic eviction. The PEFT configuration uses effective scale one (e.g., alpha equal to rank) and the stated Gaussian initialization. There is no Adam optimizer, momentum, clipping, KL penalty, or token-length normalization. Enabling the evaluator cache requires identical deterministic replay; infrastructure retries are recorded separately. Invalid actions end the controlled rollout; benchmark-native scores remain distinct from the private training utility.

Hyperparameters and prompts are fixed using only D_P before feedback training. Neither D_F nor D_A is used to select hyperparameters, checkpoints, or prompts; the evaluated editor is the final ϕ_T. With $T = 768$, $n = 12$, and no infrastructure retries, the evaluator-cache bound is $T + (T + 1)n = 9996$ environment episodes per condition and run, before audit/transfer evaluation; at $r = 4$, feedback capacity is at most 3072 bits. These are upper bounds on resource use; measured costs are reported in Table 10.

B.5 Evaluation of the learned editor and the feedback channel

Editor transfer and error propagation. Evaluate ϕ_0 and ϕ_T separately, each frozen and starting from empty memory on the same fresh experience streams. Use the training transition rule, retrieval limits, and sampling distribution. Neither run queries D_F or updates parameters. At prespecified edit counts, copy the bank into a separate frozen actor and measure utility on a disjoint query set; these measurements never enter the editing process. Summarize the checkpoint utilities by final utility, normalized area under the curve, and maximum drawdown $\max_{a<b}(J_Q(\mathcal{M}_a) - J_Q(\mathcal{M}_b))_+$, where J_Q is mean query-set utility. These metrics assess editing quality throughout the stream.

Memory transfer. Separately transplant the predetermined final training bank $\mathcal{M}_T$ into a fresh frozen actor with writing disabled. Compare it with empty memory and a length-matched bank under the same retrieval and execution budgets. No editor parameters, interaction history, or archive accompany the transplant. This evaluation measures stored-content utility; the empty-bank experiment evaluates the learned editing rule. Audit tasks are used after training is locked, followed by the final unseen test set without further design changes.

Mechanism comparisons. On ScienceWorld with Qwen3-14B, compare a frozen editor ($\eta = 0$), ICME, and learning from full-precision paired gains using the same actor, initialization, task order, direct transition, and proposal count. Compare rates $r \in \{2, 4, 8\}$ at fixed proposal count; separately report fixed-total-bit comparisons, which change the number of proposals. A deterministic nearest-grid encoder, with exact ties toward zero, measures the overall effect of changing rounding rules. The comparison captures the combined change in bias and variance. ReasoningBank and ACE supply experience-driven memory baselines (Ouyang et al., 2026; Zhang et al., 2026c). Controlled shared-actor comparisons and published end-to-end systems are reported separately.

Final-bank audit. The same fixed-horizon ScienceWorld controls supply final banks for a read-only comparison on D_F and the disjoint audit set D_A. Report each mean utility and the signed difference, using matched task-family weights. The frozen-editor bank is constructed from the same prespecified visible-task schedule without feedback or parameter updates. Auditing occurs once after training and output selection are complete. The signed gap measures the final bank's empirical optimism relative to the independent audit set.

B.6 Mechanism-specific metrics and statistical reporting

Let J_j be mean query utility after t_j deployment edits, with $0 = t_0 < \cdots < t_m = H$. Metrics comprise final utility J_m, normalized trapezoidal area and maximum drawdown

$$\mathrm{AUC} = \frac{1}{H}\sum_{j=1}^{m}(t_j - t_{j-1})\frac{J_{j-1} + J_j}{2}, \qquad \mathrm{MDD} = \max_{0 \le i < j \le m}(J_i - J_j)_+,$$

and editor-transfer gain $G_{\mathrm{edit}} = J_m(\phi_T) - J_m(\phi_0)$ under paired streams. MDD measures deterioration at the resolution of the observed checkpoints. The final-bank audit gap is $\widehat{J}_F(\mathcal{M}_T) - \widehat{J}_A(\mathcal{M}_T)$, using the same utility and matched task-family weights within ScienceWorld. This read-only audit occurs once after the final output is fixed and measures the bank's empirical optimism on the reused feedback tasks.

The additional mechanism analysis uses Qwen3-14B on ScienceWorld at $C = 1024$ and $L = 512$, with the same grouped split, actor, initialization seeds, and task order across controls. The controls are the frozen editor ($\eta = 0$), full-precision paired gains, ICME at $r \in \{2, 4, 8\}$, and nearest-grid rounding at $r = 4$ with ties toward zero. Learned controls share $T = 768$; the frozen editor supplies the untrained reference. A separate fixed-bit comparison uses $B = 3072$ and $T = \lfloor B/r \rfloor$, retaining

32-edit resets and charging the resulting extra interaction cost. For each paired seed, all horizons use prefixes of one prespecified 1536-edit task sequence whose 768-edit prefix matches the default run. Full precision is compared at the fixed horizon. Deterministic rounding measures the joint effect of changes in bias and variance.

Statistical reporting aggregates ten paired runs. The 95% paired-bootstrap intervals use 10,000 resamples of complete runs, preserving all folds, streams, and checkpoints. Resampling complete runs preserves dependence among episodes that share evolving memory. Audit-gap and editor-gain intervals are computed from paired run-level differences. Marginal intervals summarize uncertainty around each mean; comparisons describe the observed means. Failed runs are retained in the run inventory. Reproducibility records retain task/split manifests, checkpoint and software revisions, prompts, environment/scoring configurations, and seeds.

B.7 Result sources and comparison protocols

The result tables distinguish published reference scores from controlled measurements. A reference row preserves the original actor, adaptation regime, benchmark partition, and metric. ICME improvements are computed within controlled blocks that share the actor and evaluation protocol. An em dash denotes a benchmark excluded from that row's coverage. Controlled rows follow Appendix B; Tables 1, 2, and 8 report mean $\pm$ 95% confidence intervals over ten paired runs. Tables 6 and 7 report point estimates. Source-reported uncertainty summaries remain distinct from controlled confidence intervals. In Tables 1–2, bold and underline mark the best and second-best controlled means within each model. MemRL/Mem^{2}Evolve and actor-training rows are unranked. The annotations rank observed means.

The compact benchmark tables use bank capacity $C = 1024$ and injected context $L = 512$, retaining each baseline's entry schema and retrieval mechanism. Infeasible capacity configurations are distinguished from missing measurements. All controlled task comparisons use this common budget; Appendix B.3 specifies the method adaptations. ReasoningBank on WebShop is an adaptation. Expanded-system rows retain their utility-based retrieval, architecture search, or tool creation. The ICME/fixed-editor pair isolates editor learning.

AppWorld. Table 2 reports controlled measurements using the official normal/challenge TGC and SGC metrics. ACE uses online memory without ground-truth labels for its reflector; DC-CU uses cumulative online memory (Zhang et al., 2026c; Suzgun et al., 2025). All methods share the 30-turn interface and ten paired runs. MemEvolve retains its architecture search. SAGE starts from Qwen2.5-32B-Instruct, performs actor SFT/RL, and uses same-scenario skill retrieval (Wang et al., 2026). Its initial checkpoint and training-data access are matched, and it is excluded from the frozen-actor ranking.

WebArena and Mind2Web. Tables 6 and 7 reproduce the Gemini-2.5-Flash rows of Tables 1 and 3 in Ouyang et al. (2026). WebArena's 684 non-Maps tasks and website grouping match the chosen evaluation pool, but the proprietary actor, sampling temperature, and memory pipeline differ from the controlled Qwen configuration. The reference block uses the single-trajectory setting and averages actions over all tasks. Mind2Web reports offline action-prediction metrics for each generalization split. Controlled rows share frozen DOM preprocessing and candidate lists.

Other published protocols. Auto-Dreamer's Table 1 uses Qwen3.5-9B actors on ALFWorld/ScienceWorld and a Gemini actor on WebArena (Ye et al., 2026). Qwen3-14B serves as its writer/consolidator. Its combined ALFWorld pool, seven-type ScienceWorld subset, and 117-task WebArena subset differ from our separate seen/unseen, full-type, and 684-task evaluations. MemRL reports last-epoch or cumulative multi-epoch success (Zhang et al., 2026d), and Mem^{2}Evolve reports GPT-5-chat scores (Cheng et al., 2026); our controlled comparisons use first-attempt SR with shared Qwen actors. MemEvolve's AppWorld row is our architecture-search adaptation (Zhang et al., 2026a).

B.8 PRELIMINARY STUDY: PROTOCOL AND DIAGNOSTICS

This subsection details the feedback-free study in Section 3 and the analysis underlying Figure 2.

B.8.1 STUDY PROTOCOL

The study uses only the design split D_P, disjoint from the training-feedback set D_F. Six independent ALFWorld streams contain 32 episodes each, and each stream starts with an empty bank of capacity $C = 1024$ tokens. At event t, the frozen writer g_0 receives the episode experience e_t and current bank $\mathcal{M}_t^-$. A fixed prompt requests one concise, reusable lesson of at most 128 tokens, which is appended directly:

$$m_t = g_0(e_t, \mathcal{M}_t^-), \qquad \mathcal{M}_t^+ = \mathcal{M}_t^- \cup \{m_t\}. \tag{11}$$

The successor bank becomes the next episode's incumbent within the same stream. The actor, retriever, and writer remain frozen, and every generated lesson is retained regardless of its measured utility. All banks remain below capacity throughout the study.

After each append, we compare the before/after banks on 12 fixed design-only probes, restoring identical initial states within each pair. These post-hoc diagnostics remain separate from the writer. A separate frozen judge scores whether the lesson is supported by its source episode and appears reusable on a one-to-five scale. This semantic score is used only in the analysis.

B.8.2 DIAGNOSTIC DEFINITIONS

Utility and affected probes. For probe z_i, let $u(\mathcal{M}, z_i) \in \{0, 1\}$ denote task success with bank $\mathcal{M}$. With $n_P = 12$, define

$$d_{t,i} = u(\mathcal{M}_t^+, z_i) - u(\mathcal{M}_t^-, z_i), \qquad \Delta_t = \frac{1}{n_P} \sum_{i=1}^{n_P} d_{t,i}. \tag{12}$$

Each $d_{t,i}$ belongs to $\{-1, 0, 1\}$. The positive, negative, and affected probe counts are

$$N_t^+ = \sum_i \mathbf{1}[d_{t,i} = 1], \qquad N_t^- = \sum_i \mathbf{1}[d_{t,i} = -1], \qquad A_t = N_t^+ + N_t^-. \tag{13}$$

Thus $\Delta_t = (N_t^+ - N_t^-)/12$. Helpful, zero-gain, and harmful writes have positive, zero, and negative Δ_t, respectively. Among zero-gain writes, $A_t = 0$ identifies unchanged probe outcomes, whereas $A_t > 0$ identifies cancellation. The near-zero condition $|\Delta_t| \le 1/12$ means $|N_t^+ - N_t^-| \le 1$ and can include several affected probes.

Occupancy and retrieval displacement. Occupancy is measured before writing as $\omega_t = \ell(\mathcal{M}_t^-)/C$, where ℓ counts memory tokens. For these append-only streams, this equals the preceding lessons' cumulative token count divided by 1024. Retrieval displacement averages a change indicator over probes and repeated diagnostic runs:

$$\chi_t = \frac{1}{R n_P} \sum_{r=1}^{R} \sum_{i=1}^{n_P} \mathbf{1}\left[\mathcal{R}^{(r)}(\mathcal{M}_t^-, z_i) \neq \mathcal{R}^{(r)}(\mathcal{M}_t^+, z_i)\right], \tag{14}$$

where R counts diagnostic runs and $\mathcal{R}^{(r)}$ is the retrieval result in run r. The repeated-run average $\chi_t \in [0, 1]$ measures changes in retrieved content; A_t measures changes in probe success. The utility fingerprint retains the twelve signed success differences in (12).

B.8.3 FIGURE CONSTRUCTION

Behavioral fingerprints. Figure 2a sorts writes by decreasing semantic score, breaking ties by decreasing gain. Each row contains twelve probe effects: blue for improvement, orange for degradation, and light gray for no change. The side tracks show semantic score on $[1, 5]$, mean gain, and displacement on $[0, 1]$. Gain shares a diverging scale from $-1/3$ to $+1/3$ across both panels. Row indices denote semantic rank.

Phase portrait. Figure 2b plots (ω_t, χ_t) for each write. Marker color encodes Δ_t; area is $20 + 12A_t$ square points, so zero-effect writes remain visible. Dashed boxes mark the two comparison regions. For the contours, the unit square is partitioned into a 38×38 grid. Total and harmful write counts are separately smoothed with the same Gaussian filter (standard deviation 2.4 bins, reflected boundaries, truncation at four standard deviations). The displayed harmful fraction is the ratio of smoothed harmful counts to smoothed total counts, retained only where the latter exceeds 0.08. Line contours mark 25%, 40%, and 55%. The region summaries use the unsmoothed observations.

B.9 Content and retrieval: replay and aggregation protocol

Protocol. We use Qwen3-14B on ScienceWorld with $C = 1024$, $L = 512$, stochastic feedback at $r = 4$, and the grouped split, task order, and ten paired seeds of Appendix B.6. FIXED, ICME-EARLY, and ICME-LATE correspond to training edits 0, 192, and 768. Each contributes 320 analyzed edits from matched trajectories. Disjoint queries D_Q use the deterministic actor and utility protocol of Appendix A.3.

Counterfactual decomposition. Let $\mathcal{C}_t^0/\mathcal{C}_t^1$ map IDs to pre/post-edit content, and let $\mathcal{R}_t^0(z)/\mathcal{R}_t^1(z)$ be the corresponding ordered retrieval lists. Restoring the same task snapshot, we compute all four utilities

$$U_{ab}(t) = \frac{1}{|D_Q|} \sum_{z \in D_Q} u\big(z; \mathcal{C}_t^a, \mathcal{R}_t^b(z)\big), \qquad (a,b) \in \{0,1\}^2. \tag{15}$$

U_{00} and U_{11} give the pre/post-edit utilities; U_{10} changes content at fixed retrieval identities, and U_{01} changes identities at fixed content. Define

$$\begin{aligned} g_t^{\text{cont}} &= U_{10}(t) - U_{00}(t), & g_t^{\text{retr}} &= U_{01}(t) - U_{00}(t), \\ g_t^{\text{int}} &= U_{11}(t) - U_{10}(t) - U_{01}(t) + U_{00}(t), & \Delta_t^Q &= g_t^{\text{cont}} + g_t^{\text{retr}} + g_t^{\text{int}}. \end{aligned} \tag{16}$$

An absent ID contributes empty content without delimiters. Replays preserve list order without reranking or backfilling and append a whole entry only if the serialized context fits L. The content effect includes the resulting length-dependent packing changes.

Visualization and aggregation. With $S_t = \sum_{k \in \{\text{cont},\text{retr},\text{int}\}} |g_t^k|$, simplex coordinates are

$$\alpha_t^k = |g_t^k|/S_t, \qquad (x_t, y_t) = \left(\alpha_t^{\text{int}} + \tfrac{1}{2}\alpha_t^{\text{cont}}, \tfrac{\sqrt{3}}{2}\alpha_t^{\text{cont}}\right).$$

Dominance uses the largest absolute component; helpful/harmful labels use the sign of Δ_t^Q. Each state's contour is the 90th percentile of Gaussian KDE density values on a triangle-masked 220×220 grid, with bandwidth factor 0.20. Centroids average per-edit coordinates. Table 12 pools edits within each state; percentage denominators are 320.

B.10 Retrieval competition: risk and visitation protocol

Diagnostic protocol. We use Qwen3-14B on ScienceWorld with $C = 1024$, $L = 512$, $r = 4$, and $T = 768$. Editors ϕ_j, $j \in \{0, 192, 384, 576, 768\}$, are frozen and evaluated on the same 24 diagnostic streams of 32 events for each of ten paired seeds. Rollouts use no training-feedback queries or parameter updates; diagnostic scores are not returned to the editor. The resulting 38,400 records contain normalized bank occupancy ω_t, aggregate retrieval displacement χ_t, and paired gain Δ_t on 12 design-only probes. An event is harmful when $\Delta_t < 0$.

Fixed reference landscape. Let p_ϕ denote the editor's empirical distribution over (ω, χ). Using only ϕ_0 events, we bin total counts N_0 and harmful counts H_0 on a 64×64 grid over $[0,1]^2$ and estimate

$$\widehat{h}_0(\omega, \chi) = \frac{(G_\sigma * H_0)(\omega, \chi)}{(G_\sigma * N_0)(\omega, \chi)}. \tag{17}$$

Here G_σ is a Gaussian filter with standard deviation 2.2 bins, reflected boundaries, and truncation at four standard deviations. Supported cells have smoothed event counts at least .08. For exposure, we bilinearly interpolate between cell centers and retain points whose contributing cells are all supported; call this domain $\mathcal{S}_0$. The surface and support stay fixed across checkpoints.

Visitation and exposure. The high-competition region is $\mathcal{D}_{\text{risk}} = \{(\omega, \chi) : \omega > .65, \chi > .50\}$. We measure

$$V_{\text{risk}}(\phi) = \Pr_{p_\phi}[(\omega, \chi) \in \mathcal{D}_{\text{risk}}],$$
$$E_{0,\text{supp}}(\phi) = \mathbb{E}_{p_\phi}\left[\widehat{h}_0(\omega, \chi) \mid (\omega, \chi) \in \mathcal{S}_0\right]. \tag{18}$$

We first compute each metric within a seed and then report the mean and a 95% Student-t interval across ten seeds. Region-conditional harmful rates pool events within each checkpoint; coverage is the fraction in $\mathcal{S}_0$.

Figure construction. Figure 6 displays the fixed risk surface and equal-risk contours at $.20, .35, .50, .60$. Initial and final visitation counts use the same grid with Gaussian standard deviation 2.0 bins; their contours mark 28% and 72% of each density's peak. Circles average all event coordinates within each checkpoint.

C Additional Experimental Results

C.1 Preliminary study: supplementary results

Reported fractions weight each write equally. The low-competition region uses $\omega_t < 0.30$ and $\chi_t < 0.30$; the populated, strongly perturbed region uses $\omega_t > 0.65$ and $\chi_t > 0.50$. Table 5 reports harmful writes in these regions and the high-semantic-score subset.

Table 5: Harmful writes in the three comparison groups.

Group	Writes	Harmful
Semantic score ≥ 4	122	29 (23.8%)
Low competition	60	11 (18.3%)
Populated / perturbed	22	13 (59.1%)

Most writes (77.1%) have $|\Delta_t| \leq 1/12$; these small net effects include both sparse changes and cancellation between positive and negative probe outcomes. The semantic-score correlation reported in the main text uses all 192 writes and average ranks for ties.

C.2 WEB AND OFFLINE-ACTION BREAKDOWNS

Table 6: **WebArena by website.** SR is in percent; actions are averaged over all tasks. Published reference values are from ReasoningBank, Table 1. The source and controlled blocks share the 684-task pool and use different actors and inference protocols. Overall SR is task-weighted.

	Website SR ↑					Overall	
Method	Shop	Admin	GitLab	Reddit	Multi	SR ↑	Actions ↓
Published reference: Gemini-2.5-Flash, ReasoningBank protocol							
No memory (ReAct)	39.0	44.5	33.9	55.7	10.3	40.5	9.7
Synapse	40.6	45.1	35.6	59.4	10.3	42.1	9.2
AWM	44.4	46.7	37.2	62.3	3.4	44.1	9.0
ReasoningBank	49.7	51.1	40.6	67.0	13.8	48.8	8.3
Controlled: Qwen3-4B							
ReasoningBank	27.5	31.8	23.9	42.6	7.1	29.2	10.4
ICME	**33.8**	**37.9**	**29.2**	**48.6**	**10.8**	**35.0**	**9.1**
Controlled: Qwen3-14B							
ReAct	31.7	37.4	29.6	48.1	8.2	34.2	10.8
Synapse	33.9	38.6	31.2	50.9	8.4	36.0	10.1
AWM	35.2	40.1	32.8	52.6	6.1	37.3	9.8
ReasoningBank	39.4	43.7	36.5	56.3	11.6	41.2	9.0
Auto-Dreamer	40.7	44.2	37.1	57.0	11.8	42.0	8.8
ICME (fixed editor)	37.2	41.5	34.6	54.0	10.3	39.1	9.4
ICME	**44.8**	**48.6**	**41.9**	**61.8**	**15.4**	**46.4**	**8.2**
Controlled: Qwen2.5-32B-Instruct							
ReasoningBank	42.6	47.3	39.4	59.1	13.5	44.3	8.7
ICME	**48.5**	**52.7**	**44.6**	**64.5**	**17.8**	**49.8**	**7.9**

Sources: Ouyang et al. (2026); Synapse (Zheng et al., 2024), AWM (Wang et al., 2025b), and Auto-Dreamer (Ye et al., 2026). Website task counts are 187/182/180/106/29. Overall SR is task-weighted. Action counts should be interpreted alongside SR. Controlled ICME deploys the ScienceWorld-trained editor without target-domain parameter updates.

Table 7: **Mind2Web generalization (%).** Element accuracy (EA), action F1, step success, and task success are reported separately for each split; all are higher-is-better. Reference values are from ReasoningBank, Table 3. Controlled methods share candidate ranking and DOM preprocessing.

	Cross-task				Cross-website				Cross-domain			
Method	EA	F1	Step	Task	EA	F1	Step	Task	EA	F1	Step	Task
Published reference: Gemini-2.5-Flash, ReasoningBank protocol												
No memory	46.0	59.1	40.3	3.3	39.8	45.1	31.7	1.7	35.8	37.9	31.9	1.0
Synapse	47.0	59.5	41.2	3.5	40.3	46.0	32.1	1.9	36.3	38.5	32.4	1.1
AWM	46.3	56.1	41.0	3.5	39.1	42.2	31.7	2.1	33.3	36.5	30.1	0.7
ReasoningBank	52.1	60.4	44.9	4.8	44.3	52.6	33.9	2.3	40.6	41.3	36.6	1.6
Controlled: Qwen3-4B												
ReasoningBank	43.8	54.7	38.1	2.8	37.5	43.2	30.2	1.4	33.6	35.4	29.7	0.8
ICME	**47.9**	**57.8**	**41.6**	**3.7**	**40.7**	**47.5**	**32.6**	**1.9**	**36.8**	**39.0**	**33.1**	**1.2**
Controlled: Qwen3-14B												
ReAct	44.9	56.5	39.4	3.0	38.7	44.0	30.9	1.5	34.8	36.9	30.5	0.9
Synapse	46.2	57.1	40.5	3.2	39.6	45.2	31.5	1.7	35.5	37.8	31.2	1.0
AWM	45.8	54.4	40.2	3.3	38.9	41.6	31.1	1.8	33.0	35.8	29.7	0.7
ReasoningBank	50.6	59.0	43.8	4.3	43.0	50.8	33.1	2.1	39.2	40.3	35.4	1.4
ICME (fixed)	49.3	58.1	42.7	4.0	42.1	49.2	32.5	1.9	38.2	39.5	34.4	1.3
ICME	**54.7**	**62.3**	**47.5**	**5.6**	**46.9**	**54.8**	**36.1**	**2.8**	**43.0**	**44.9**	**39.1**	**2.0**
Controlled: Qwen2.5-32B-Instruct												
ReasoningBank	53.8	61.5	46.8	5.2	45.9	53.8	35.0	2.5	41.9	42.8	37.8	1.8
ICME	**58.2**	**65.7**	**50.9**	**6.6**	**49.8**	**57.9**	**38.6**	**3.4**	**46.1**	**48.5**	**42.0**	**2.5**

Source: Ouyang et al. (2026); baseline definitions follow Table 6. All methods within a controlled block share candidate ranking and DOM preprocessing. Offline labels are used only for scoring, separately from private-feedback training. This diagnostic uses the frozen ScienceWorld-trained editor and a read-only bank constructed from disjoint training-side web experience; the query split never updates the bank. The fixed-editor row shares that experience and retrieval configuration.

C.3 EDITOR TRANSFER VERSUS FROZEN-BANK TRANSFER

Table 8 separates learning to edit a new bank from reusing an existing bank. Editor transfer starts both ϕ_0 and ϕ_T from empty memory, freezes their parameters, and evaluates on disjoint query groups. ICME increases J_{32} by 0.048–0.085 over the fixed editor, with higher AUC and lower MDD in all nine conditions. Frozen-bank evaluation disables writing and contrasts $\mathcal{M}_T$ with empty memory and a token-matched fixed-editor bank; ICME banks improve query utility by 0.063–0.084 over fixed-editor banks. These measurements remain separate from the online first-attempt results in Tables 1–2.

Table 8: **Transfer of the editor and the memory bank.** Qwen3-14B; each ICME/fixed-editor pair shares its actor, experience stream, retrieval, and capacity. Query utility J, AUC, and maximum drawdown (MDD) lie in $[0, 1]$; G_{edit} is an absolute utility difference. Values are mean ± 95% CI over ten paired runs on held-out query groups.

Source → target	Editor	J_{32} ↑	AUC ↑	MDD ↓	G_{edit} ↑
(a) Editor transfer: empty initial banks, no parameter updates					
ALFWorld → seen	Fixed	.667±.014	.631±.013	.078±.011	—
	ICME	**.742±.013**	**.699±.012**	**.041±.008**	**.075±.015**
ALFWorld → unseen	Fixed	.598±.015	.564±.014	.092±.012	—
	ICME	**.679±.014**	**.635±.013**	**.049±.009**	**.081±.016**
ScienceWorld → ScienceWorld	Fixed	.556±.013	.527±.012	.083±.011	—
	ICME	**.636±.012**	**.592±.011**	**.044±.008**	**.080±.014**
WebShop → WebShop	Fixed	.541±.012	.511±.012	.088±.011	—
	ICME	**.614±.011**	**.575±.011**	**.046±.008**	**.073±.013**
AppWorld → normal	Fixed	.601±.015	.567±.014	.097±.012	—
	ICME	**.676±.014**	**.630±.013**	**.052±.009**	**.075±.016**
AppWorld → challenge	Fixed	.431±.016	.398±.015	.116±.014	—
	ICME	**.516±.015**	**.474±.014**	**.061±.010**	**.085±.017**
ScienceWorld → ALFWorld seen	Fixed	.619±.015	.590±.014	.091±.012	—
	ICME	**.671±.014**	**.626±.013**	**.058±.009**	**.052±.015**
ScienceWorld → ALFWorld unseen	Fixed	.548±.016	.517±.015	.103±.013	—
	ICME	**.603±.015**	**.560±.014**	**.066±.010**	**.055±.016**
ScienceWorld → WebArena	Fixed	.331±.014	.306±.013	.121±.014	—
	ICME	**.379±.013**	**.348±.012**	**.074±.010**	**.048±.014**

Evaluation pool	Empty memory	Fixed-editor bank	**ICME bank**
(b) Memory transfer: final training banks, all writing disabled; query utility ↑			
ALFWorld seen	.607±.014	.676±.013	**.739±.012**
ALFWorld unseen	.539±.015	.604±.014	**.678±.013**
ScienceWorld	.498±.013	.559±.012	**.634±.011**
WebShop	.486±.012	.547±.011	**.619±.010**
AppWorld normal	.548±.015	.610±.014	**.683±.013**
AppWorld challenge	.376±.016	.437±.015	**.521±.014**

Panel (a) uses five group folds and editing checkpoints $0, 8, 16, 24, 32$; ALFWorld's seen and unseen pools remain separate. $G_{\text{edit}} = J_{32}(\phi_T) - J_{32}(\phi_0)$ is reported only for the learned/fixed pair. Panel (b) uses each domain's own training bank and a token-matched fixed-editor bank. Native task metrics remain in the benchmark tables; query utility follows Appendix A.3. Means and 95% intervals aggregate ten paired runs.

C.4 FEEDBACK PRECISION, BUDGET, AND FINAL-BANK AUDIT

The matched analysis uses Qwen3-14B on ScienceWorld with the split, retrieval, and deployment protocol of Appendix B. Table 9 varies the feedback encoder at a fixed training horizon. Table 10 fixes total feedback capacity while exposing the resulting training cost. Table 11 evaluates the final training banks separately from editor transfer. The fixed-editor and default $r = 4$ editor-transfer results correspond to the ScienceWorld conditions in Table 8(a).

Table 9: **Feedback ablation at a fixed horizon.** Learned controls use $T = 768$; the fixed editor is the untrained reference. $B = Tr$ counts feedback bits. A dash denotes an inapplicable bit width or an unassigned full-precision budget. Utility summaries use mean ± 95% CI over ten paired runs.

Editor / feedback	r	B (bits)	J_{32} ↑	AUC ↑	MDD ↓
ICME (fixed editor)	—	0	.556±.013	.527±.012	.083±.011
ICME (full precision)	—	—	**.650±.011**	**.606±.011**	**.038±.007**
ICME (nearest grid)	4	3072	.619±.013	.575±.012	.057±.009
ICME (stochastic)	2	1536	.607±.013	.565±.012	.063±.010
ICME (stochastic)	4	3072	.636±.012	.592±.011	.044±.008
ICME (stochastic)	8	6144	.645±.011	.600±.011	.041±.008

Table 10: **Fixed feedback budget: performance and training cost.** All rows use stochastic ICME with $B = 3072$, $T = \lfloor B/r \rfloor$, and a bank reset every 32 edits. Costs include all training episodes and model input/output tokens; elapsed time uses the same 8×A100 allocation. Both panels report mean ± 95% CI over ten paired runs.

			Editor transfer		
Method	r	T	J_{32} ↑	AUC ↑	MDD ↓
ICME	2	1536	.630±.012	.587±.011	.047±.008
ICME	4	768	**.636±.012**	**.592±.011**	**.044±.008**
ICME	8	384	.624±.013	.580±.012	.050±.009
			Measured training cost		
Method	r	T	Env. episodes	Tokens (10^6)	Time (h)
ICME	2	1536	15824±286	128.9±2.8	27.7±0.7
ICME	4	768	8073±176	66.4±1.6	14.6±0.4
ICME	8	384	4126±108	34.7±0.9	7.9±0.3

Table 11: **Final-bank feedback–audit gap.** Final training banks come from the fixed-horizon conditions in Table 9; the fixed-editor bank follows the same visible-task schedule. Evaluation is read-only with matched task-family weights. Values are mean ± 95% CI of each utility and the paired signed gap. The signed gap measures empirical optimism of the final bank.

Editor / feedback	r	B (bits)	$\widehat{J}_F$ ↑	$\widehat{J}_A$ ↑	$\widehat{J}_F - \widehat{J}_A$
ICME (fixed editor)	—	0	.561±.013	.558±.012	+.003±.006
ICME (full precision)	—	—	**.675±.012**	**.642±.012**	+.033±.010
ICME (nearest grid)	4	3072	.628±.013	.612±.012	+.016±.009
ICME (stochastic)	2	1536	.620±.013	.607±.013	+.013±.009
ICME (stochastic)	4	3072	.653±.012	.632±.011	+.021±.008
ICME (stochastic)	8	6144	.666±.012	.640±.011	+.026±.009

C.5 CONTENT AND RETRIEVAL DECOMPOSITION

All 960 edits have $S_t > 0$. Table 12 reports the signed component gains, simplex coordinates, and edit counts for each editor state. The replay definitions and aggregation protocol are given in Appendix B.9.

Table 12: **Content, retrieval, and interaction effects.** Component gains are signed means; simplex coordinates average normalized absolute contributions. Counts and percentages use all 320 edits per state.

Statistic	FIXED	ICME-EARLY	ICME-LATE
Mean g^{cont}	.00573	.01299	.02075
Mean g^{retr}	$-.00420$	$-.00012$	.00045
Mean g^{int}	$-.00152$	.00435	.00488
Mean Δ^{Q}	.00001	.01722	.02608
Mean α^{cont}	.293	.346	.413
Mean α^{retr}	.401	.338	.299
Mean α^{int}	.306	.316	.288
Content-dominant helpful edits	60 (18.8%)	87 (27.2%)	127 (39.7%)
Retrieval-dominant harmful edits	77 (24.1%)	50 (15.6%)	37 (11.6%)
All harmful edits	148 (46.3%)	112 (35.0%)	83 (25.9%)

C.6 RETRIEVAL-COMPETITION RISK AND VISITATION

Table 13 reports visitation, exposure, and harmful-event rates across the five frozen editor checkpoints. The reference landscape and statistical definitions are given in Appendix B.10.

Table 13: **Risk visitation and exposure across editor checkpoints.** Visitation, exposure, and overall harm show means $\pm$ 95% interval half-widths. All rates except $E_{0,\text{supp}}$ are percentages. Each checkpoint contains 7,680 events.

Checkpoint	V_{risk}	$E_{0,\text{supp}}$	Overall harm	Region harm	Coverage
ϕ_0	22.3 ± 0.8	$.285\pm.004$	28.5 ± 0.9	50.8	99.9
ϕ_{192}	17.2 ± 0.7	$.266\pm.003$	26.3 ± 0.6	49.1	99.8
ϕ_{384}	12.8 ± 0.9	$.247\pm.004$	24.2 ± 1.5	51.3	99.6
ϕ_{576}	8.3 ± 0.7	$.225\pm.005$	22.1 ± 0.9	48.2	99.1
ϕ_{768}	4.8 ± 0.6	$.203\pm.004$	20.5 ± 1.4	49.0	97.6

Mean occupancy decreases from .475 to .430, while mean displacement decreases from .410 to .272. The high-competition region contains 1,713, 1,321, 982, 641, and 367 events across the five checkpoints, respectively.

D PROOFS

D.1 ROUNDING LAW AND THE SMALL-GAIN REGIME

For any fixed $x \in [-1, 1]$, write $x = h(j + \rho - K)$ using the index convention in (3). The encoded value is $\widehat{x} = h(j + \beta - K)$ with $\beta \sim \text{Bernoulli}(\rho)$. Consequently,

$$\mathbb{E}\widehat{x} = x, \qquad \mathbb{E}(\widehat{x} - x)^2 = h^2\rho(1-\rho) \le h^2/4. \tag{19}$$

Both endpoints and all grid points are exact. In particular, zero has index K and is transmitted deterministically. Nonzero outcomes never reverse the sign of x. For $0 < |x| < h$, the law reduces to $\widehat{x} = \text{sgn}(x)h$ with probability $|x|/h$, and zero otherwise. Hence

$$\frac{\sqrt{\text{Var}(\widehat{x} \mid x)}}{|x|} = \sqrt{\frac{h}{|x|} - 1}, \qquad 0 < |x| < h. \tag{20}$$

The ratio is the conditional standard deviation relative to the signal magnitude. It diverges as a nonzero gain approaches zero despite exact encoding at zero itself. For $x = 0.01$, it is approximately 9.95 at $r = 2$ and 3.64 at $r = 4$. At $r = 4$, a gain of 0.01 produces the nonzero update with probability 0.07. These values follow from the rounding law. Higher precision lowers the worst-case variance bound; variance at a particular gain can increase because the grids are not nested.

Unbiased encoding of both signs with exact zero requires at least three output levels. One symbol must represent zero, and the codebook's convex hull must contain both -1 and 1. The protocol therefore requires integer $r \ge 2$. Its r-bit interface has $2^r - 1$ used symbols; the remaining codeword is invalid. Encoder arithmetic uses the exact reported score representation, with no tuned zero tolerance. For rational benchmark scores, integer numerator/denominator arithmetic determines the grid interval and rounding probability.

D.2 LOCAL GRADIENT AND COVARIANCE

Fix the state s, feedback set D_F, and pre-update parameters ϕ. Let $\Delta(p)$ be the paired gain after applying patch p. The finite sequence support is fixed, and the application rule and frozen actor do not depend on ϕ once p and s are given. Differentiating the finite sum yields

$$\begin{aligned} \nabla_\phi \mathcal{J}_F(\phi; s) &= \sum_p \Delta(p) \nabla_\phi q_\phi(p \mid s) \\ &= \mathbb{E}_p[\Delta(p) \nabla_\phi \log q_\phi(p \mid s)] = \mathbb{E}_{p,\beta}[\widehat{\Delta}(p) \nabla_\phi \log q_\phi(p \mid s)]. \end{aligned} \tag{21}$$

The final equality uses unbiased rounding conditional on the patch. The incumbent score is independent of p, and $\mathbb{E}_p \nabla_\phi \log q_\phi(p \mid s) = 0$, so subtracting that score preserves the gradient of expected successor utility. This baseline ensures unbiasedness; variance minimization is a separate criterion. If d_i is the utility change on task i and v is a patch's score vector, then $n^{-1} \sum_i d_i v = \Delta v$; task grouping is unnecessary for this mean-utility estimator.

Write $v(p) = \nabla_\phi \log q_\phi(p \mid s)$. The law of total covariance gives

$$\text{Cov}_{p,\beta}(\widehat{\Delta} v) = \text{Cov}_p(\Delta v) + \mathbb{E}_p[\sigma_r^2(\Delta(p)) v(p) v(p)^\top]. \tag{22}$$

The extra term is positive semidefinite and has trace $\mathbb{E}_p[\sigma_r^2(\Delta(p)) \|v(p)\|_2^2]$. This separates patch-sampling variance from encoder variance, whose magnitude depends on the score norms. The method estimates a conditional one-step gradient at each visited state, holding earlier updates fixed. Memory quality can decrease along a stream: direct application allows a harmful edit to affect subsequent experiences. We therefore evaluate terminal utility and drawdown over complete trajectories.

D.3 INFORMATION CAPACITY AND EXPECTED FEEDBACK BIAS

Coarse bounds. Let W contain the design resources, exogenous tasks, reset schedule, initialization, and all independent random tapes, including evaluator rounding randomness but excluding realized adaptive trajectories. When the only feedback is $c_{1:T}$, the communication budget $B_T = Tr$ bounds the information in the full learned output:

$$I(D_F; \phi_T, \mathcal{M}_T \mid W) \le B_T \ln 2. \tag{23}$$

If additionally $D_F \sim \mathcal{P}^n$ is independent of W, bounded-utility information inequalities give

$$\left|\mathbb{E}[\widehat{J}_F(\mathcal{M}_T) - J(\mathcal{M}_T)]\right| \le \min\left\{1, \sqrt{\frac{B_T \ln 2}{2n}}\right\}, \qquad J(\mathcal{M}) = \mathbb{E}_{z\sim\mathcal{P}} u(\mathcal{M}, z). \tag{24}$$

These bounds include the editor parameters in the information accounting but evaluate the final bank in the utility gap. Theorem 1 sharpens the bank guarantee using its causal prefix and private evaluator entropy.

Capacity proof. Let $\mathcal{H}_t$ contain the entire learner-side state: editor parameters, memory, ID counter, visible experience, retrieval index, and received transcript. Let W contain design resources, the exogenous task sequence, reset schedule, initialization, and all independent random tapes. It excludes realized trajectories, which depend on earlier edits. Initially $\mathcal{H}_0$ depends only on W. If $\mathcal{H}_{t-1}$ is determined by $(W, c_{1:t-1})$, then its next experience, serialized state, sampled patch, direct memory transition, and score vector are determined by the same variables. The parameter update additionally uses only c_t. Induction gives $\mathcal{H}_T = \Psi(W, c_{1:T})$, and therefore

$$\begin{aligned} I(D_F; \phi_T, \mathcal{M}_T \mid W) &\le I(D_F; c_{1:T} \mid W) \\ &\le \sum_{t=1}^{T} H(c_t \mid W, c_{<t}) \le T \ln(2^r - 1) \le Tr \ln 2. \end{aligned} \tag{25}$$

Here H is Shannon entropy in nats. Real-valued parameter updates are downstream of the same transcript and create no additional feedback information. The tighter alphabet bound uses the unused codeword; Tr is the actual fixed-length communication budget. The budget accumulates across bank resets. Reusing the same feedback set across additional training runs requires accounting for their transcripts together.

Expected signed-gap proof. Assume $D_F \sim \mathcal{P}^n$ is independent of W, and $u(\mathcal{M}, z) \in [0, 1]$ is a fixed scoring functional. Define $g(D_F, \mathcal{M}) = \widehat{J}_F(\mathcal{M}) - J(\mathcal{M})$. For fixed $W = w$, let μ_w be the joint law of $(D_F, \mathcal{M}_T)$ and let $\nu_w = \mathcal{P}^n \otimes \mathcal{L}(\mathcal{M}_T \mid W = w)$ be the product of its marginals. Hoeffding's lemma gives $\log \mathbb{E}_{\nu_w} \exp(\lambda g) \le \lambda^2/(8n)$. Applying the variational relative-entropy inequality to both signs of g, for $\lambda > 0$,

$$|\mathbb{E}_{\mu_w} g| \le \frac{D_{\mathrm{KL}}(\mu_w \| \nu_w)}{\lambda} + \frac{\lambda}{8n} = \frac{I(D_F; \mathcal{M}_T \mid W = w)}{\lambda} + \frac{\lambda}{8n}. \tag{26}$$

Here $\mathcal{L}$ denotes a probability law and D_{KL} is relative entropy. Optimize λ, average over W, and use Jensen's inequality to obtain

$$|\mathbb{E} g| \le \sqrt{\frac{I(D_F; \mathcal{M}_T \mid W)}{2n}} \le \sqrt{\frac{Tr \ln 2}{2n}}. \tag{27}$$

Combining this with $|g| \le 1$ proves (24). The argument instantiates established information-theoretic generalization bounds (Russo & Zou, 2016; Xu & Raginsky, 2017). Here $|\mathbb{E} g|$ measures expected signed bank optimism; $\mathbb{E}|g|$ also includes sampling fluctuations. Per-run deviations and editor transfer require separate analyses.

Numerical scale. For the displayed bound to be at most $\epsilon \in (0, 1)$, it suffices that $n \ge Tr \ln 2/(2\epsilon^2)$. At $\epsilon = 0.05$, this requires approximately $138.63Tr$ feedback tasks. For $T = 192$ and $r = 4$, that is at least 106468 tasks; at $n = 12$, the clipped bound is 1. The capacity bound quantifies information obtained from D_F, conditional on visible experience. Its statistical interpretation requires i.i.d. feedback tasks and independence from the learner's initial resources, in addition to disjoint task IDs.

D.4 Adaptive-feedback generalization

This subsection proves the main-text bank guarantee (Theorem 1) and develops its refinements: Theorems 2–4 and Corollary 1. The refinements account for conditional information, private rounding entropy, distinct outputs, and the binary-KL deviation exponent. The proofs build on information and description-length bounds (Russo & Zou, 2016; Xu & Raginsky, 2017; Dwork et al., 2015). The lower bound concerns the general finite-feedback interface at fixed confidence.

D.4.1 Statistical setting and the relevant feedback prefix

Assumption 1 (Isolated, fixed-horizon feedback)**.** $D_F = (Z_1, \dots, Z_n) \sim \mathcal{P}^n$. The variable V, containing design resources, exogenous tasks, reset schedules, initialization, and all *learner-side* random tapes, is independent of D_F. The evaluator uses independent private uniforms $U_1, \dots, U_T$, independent of (D_F, V), for rounding. These uniforms are *excluded* from V. All learner-side dependence on D_F passes through $c_{1:T}$; no scores, timing, diagnostic selection, or other side channels are released. The horizon and reported output rule are fixed before querying D_F.

The evaluator's rounding remains random conditional on V; the variable W in Appendix D.3 includes that randomness. Conditional on $(V, c_{<t})$, the current bank, experience, patch, and score vector are determined. In particular, direct application precedes the current reward, so there are deterministic maps satisfying

$$\mathcal{M}_T = F_T(V, c_{1:T-1}), \qquad (\phi_T, \mathcal{M}_T) = G_T(V, c_{1:T}). \tag{28}$$

The final bank therefore uses a prefix of length $\tau = T-1$, while the full learned output uses $\tau = T$. The communication ledger remains Tr bits in both cases.

More generally, let $O = \Psi(V, c_{1:\tau})$ be a reported output and let $f_v(o, z) \in [0, 1]$ be a scoring functional fixed independently of D_F conditional on $V = v$. For the final bank, take $O = \mathcal{M}_T$ and $f_v(o, z) = u(o, z)$. Write

$$\widehat{\mu}(o; v) = \frac{1}{n} \sum_{i=1}^{n} f_v(o, Z_i), \qquad \mu(o; v) = \mathbb{E}_{Z \sim \mathcal{P}} f_v(o, Z), \qquad G = \widehat{\mu}(O; V) - \mu(O; V). \tag{29}$$

Let $M_r = 2^r - 1$ be the used alphabet size. For fixed v, let $\mathcal{O}_\tau(v)$ be the image of all length-τ code strings under $\Psi(v, \cdot)$, allowing impossible histories to be completed by any fixed learner-side rule. Its cardinality $N_\tau(v)$ obeys $1 \le N_\tau(v) \le M_r^\tau$; different transcripts can produce the same output. All logarithms below are natural. Set $h_2(p) = -p \ln p - (1-p) \ln(1-p)$ and use the usual continuous boundary conventions for binary relative entropy $\mathrm{kl}(p \| q)$.

D.4.2 Probability conventions and elementary KL facts

All random variables take values in standard Borel spaces, all maps and scores are measurable, and all conditional statements are understood almost surely. This includes the finite strings, real-valued parameters, and countable random tapes of the protocol. Write $D = D_F$, $C_{<t} = c_{1:t-1}$, and $C_{1:\tau} = c_{1:\tau}$. Empty transcripts are deterministic. Conditional on $V = v$, reported outputs have finite support even when their ambient space is continuous. The sample size is an integer $n \ge 1$. Relative entropy and its binary specialization are

$$\begin{aligned} \mathrm{KL}(P \| Q) &= \int \ln\left(\frac{dP}{dQ}\right) dP \qquad \text{if } P \ll Q, \quad \text{and } +\infty \text{ otherwise,} \\ \mathrm{kl}(p \| q) &= p \ln \frac{p}{q} + (1-p) \ln \frac{1-p}{1-q}. \end{aligned} \tag{30}$$

We set $0 \ln 0 = 0$, use the continuous extensions at matching endpoints, and assign $+\infty$ when a positive mass is divided by zero. Conditional mutual information is the average KL divergence between a conditional joint law and the product of its conditional marginals.

Lemma 1 (KL calculus)**.** *Relative entropy is nonnegative, jointly convex, and decreases under a common Markov kernel. It obeys the joint-law chain rule. Moreover, for $p, q \in [0, 1]$,*

$$\mathrm{kl}(p \| q) \ge 2(p - q)^2. \tag{31}$$

Proof. We give the details used in the later conditional arguments. If $P \ll Q$ has density L, convexity of $x \ln x$ gives

$$\mathrm{KL}(P \| Q) = \int L \ln L \, dQ \ge \left(\int L \, dQ \right) \ln \left(\int L \, dQ \right) = 1 \ln 1 = 0. \tag{32}$$

If absolute continuity fails the claim is immediate. For nonnegative numbers a_j, b_j with sums $a, b > 0$, let $\alpha_j = a_j/a$ and $\beta_j = b_j/b$. Nonnegativity applied to these discrete laws proves the log-sum inequality:

$$\begin{aligned} \sum_j a_j \ln \frac{a_j}{b_j} - a \ln \frac{a}{b} &= a \sum_j \alpha_j \ln \frac{\alpha_j}{\beta_j} = a \,\mathrm{KL}(\alpha \| \beta) \geq 0, \\ \left(\sum_j a_j \right) \ln \frac{\sum_j a_j}{\sum_j b_j} &\leq \sum_j a_j \ln \frac{a_j}{b_j}. \end{aligned} \tag{33}$$

Zero masses follow by limits. Apply this pointwise to $a_j = \lambda_j p_j(x)$ and $b_j = \lambda_j q_j(x)$ for densities with respect to a common dominating measure. Integration gives joint convexity:

$$\begin{aligned} \mathrm{KL}\left(\sum_j \lambda_j P_j \,\middle\|\, \sum_j \lambda_j Q_j \right) &= \int \left(\sum_j \lambda_j p_j \right) \ln \frac{\sum_j \lambda_j p_j}{\sum_j \lambda_j q_j} \, dx \\ &\leq \int \sum_j \lambda_j p_j \ln \frac{p_j}{q_j} \, dx = \sum_j \lambda_j \,\mathrm{KL}(P_j \| Q_j). \end{aligned} \tag{34}$$

The same proof with conditional Jensen establishes the version for an arbitrary mixing distribution, which we use when averaging over V and O.

For joint laws P_{XY}, Q_{XY} with finite divergence, disintegration factors the density ratio into marginal and conditional ratios. Integrating its logarithm gives

$$\begin{aligned} \mathrm{KL}(P_{XY} \| Q_{XY}) &= \mathbb{E}_{P_{XY}} \left[\ln \frac{dP_X}{dQ_X}(X) + \ln \frac{dP_{Y|X}}{dQ_{Y|X}}(Y \mid X) \right] \\ &= \mathrm{KL}(P_X \| Q_X) + \mathbb{E}_{P_X} \,\mathrm{KL}(P_{Y|X} \| Q_{Y|X}) \geq \mathrm{KL}(P_X \| Q_X). \end{aligned} \tag{35}$$

For a common kernel $K(dy \mid x)$, apply this identity to PK and QK. Their conditional laws of Y given X agree, whereas their conditional laws of X given Y need not. Consequently,

$$\begin{aligned} \mathrm{KL}(P \| Q) &= \mathrm{KL}(PK \| QK) \\ &= \mathrm{KL}((PK)_Y \| (QK)_Y) + \mathbb{E}_{(PK)_Y} \,\mathrm{KL}((PK)_{X|Y} \| (QK)_{X|Y}) \\ &\geq \mathrm{KL}((PK)_Y \| (QK)_Y). \end{aligned} \tag{36}$$

Here PK on the first two lines denotes the joint law of (X, Y). Infinite-divergence cases follow by the same extended-valued chain rule and are trivial for the inequalities used below. Applying (35) to a joint law versus its marginal product also yields the mutual-information chain rule, conditionally on any additional variable.

Finally fix $q \in (0, 1)$ and define $F_q(p) = \mathrm{kl}(p \| q) - 2(p - q)^2$. Direct differentiation on $(0, 1)$ gives

$$\begin{aligned} F_q(q) &= F_q'(q) = 0, \\ F_q''(p) &= \frac{1}{p} + \frac{1}{1-p} - 4 = \frac{1}{p(1-p)} - 4 \geq 0, \\ F_q(p) &\geq F_q(q) + F_q'(q)(p - q) = 0. \end{aligned} \tag{37}$$

Continuity in p treats $p = 0, 1$; for $q = 0, 1$ the divergence is either zero at $p = q$ or infinite. This proves (31) on the whole square. □

D.4.3 Causal reconstruction and private rounding entropy

Lemma 2 (Prefix reconstruction)**.** *Under Assumption 1, the learner state immediately before the round-t reward is a deterministic function of $(V, C_{<t})$. The state immediately after the update is a deterministic function of $(V, C_{1:t})$. The rounding parameters (j_t, ρ_t) are deterministic functions of $(D, V, C_{<t})$, and*

$$H(c_t \mid D, V, C_{<t}) = \mathbb{E} h_2(\rho_t). \tag{38}$$

Proof. Let L_t contain the entire learner-side state after update t: parameters, bank, identifier counter, visible history, and all local persistent variables. Initially L_0 is determined by V. Suppose inductively that $L_{t-1} = \Lambda_{t-1}(V, C_{<t})$. The predetermined reset, if present, reads no evaluator-private data. The next visible task and the learner's sampling tape are fixed by V. Actor execution, retrieval, serialization, patch sampling, and direct application therefore give deterministic maps

$$\begin{aligned}(s_t, p_t, \mathcal{M}_t, v_t) &= \Gamma_t(V, C_{<t}),\\ \Delta_t &= \widehat{J}_D(\mathcal{M}_t) - \widehat{J}_D(\mathcal{M}_t^-) = d_t(D, V, C_{<t}),\\ L_t &= \Lambda_t(V, C_{<t}, c_t).\end{aligned} \tag{39}$$

The first line is independent of the *current* code because a valid edit is installed before that code is received. The second line holds because feedback replay and its score representation are deterministic. The third line includes the parameter update, which reads the decoded code and already reconstructed learner-side quantities. This closes the induction. Taking the memory component before reward T and the full component after reward T proves (28). After a reset, the parameters and other retained learner state still depend on earlier codes.

Conditional on $(D, V, C_{<t})$, the interval index j_t and probability ρ_t are fixed. Earlier codes depend only on $U_{<t}$, so U_t remains uniform and independent under this conditioning. The conditional code law is exactly

$$\begin{aligned}\Pr(c_t = j_t \mid D, V, C_{<t}) &= 1 - \rho_t,\\ \Pr(c_t = j_t + 1 \mid D, V, C_{<t}) &= \rho_t,\\ H(c_t \mid D = d, V = v, C_{<t} = c_{<t}) &= -(1-\rho_t)\ln(1-\rho_t) - \rho_t \ln \rho_t.\end{aligned} \tag{40}$$

This includes deterministic endpoint and grid-point codes, for which $\rho_t \in \{0, 1\}$. Averaging proves (38). Conditioning additionally on U_t makes the code deterministic given the dataset and removes its conditional entropy $h_2(\rho_t)$. Keeping evaluator randomness outside V therefore yields the entropy refinement; W gives the coarser bound. □

Proposition 1 (Exact channel accounting and output contraction)**.** *For every output $O = \Psi(V, C_{1:\tau})$ under the assumptions of Theorem 2, the identity* (43) *and the first line of* (44) *hold.*

Proof. All code alphabets are finite. Their conditional entropy chain rule follows by factoring the conditional probability of a string and taking the expectation of minus its logarithm. Subtract the corresponding factorization conditional on D, and apply Lemma 2:

$$\begin{aligned}I(D; C_{1:\tau} \mid V) &= H(C_{1:\tau} \mid V) - H(C_{1:\tau} \mid D, V)\\ &= \sum_{t=1}^{\tau} [H(c_t \mid V, C_{<t}) - H(c_t \mid D, V, C_{<t})]\\ &= \sum_{t=1}^{\tau} [H(c_t \mid V, C_{<t}) - \mathbb{E}h_2(\rho_t)]\\ &\le \sum_{t=1}^{\tau} [\ln M_r - \mathbb{E}h_2(\rho_t)] \le \tau \ln M_r.\end{aligned} \tag{41}$$

Each conditional code law has at most M_r symbols. Its entropy is at most $\ln M_r$: the KL divergence from the uniform law equals $\ln M_r$ minus that entropy, and is nonnegative by Lemma 1. Each summand before this entropy bound is a nonnegative conditional mutual information.

For fixed v, at most M_r^τ strings are supplied to $\Psi(v, \cdot)$. Merging identical outputs gives a family $\mathcal{O}_\tau(v)$ with size $N_\tau(v) \le M_r^\tau$, regardless of the ambient dimension or precision of its parameters. Data processing and conditional output entropy give two distinct bounds:

$$\begin{aligned}I(D; O \mid V) &\le I(D; C_{1:\tau} \mid V) = \mathcal{I}_\tau,\\ I(D; O \mid V) &= H(O \mid V) - H(O \mid D, V)\\ &\le H(O \mid V)\\ &= \mathbb{E}_V\left[\ln N_\tau(V) - \mathrm{KL}(\mathcal{L}(O \mid V) \,\|\, \mathrm{Unif}(\mathcal{O}_\tau(V)))\right]\\ &\le \mathbb{E} \ln N_\tau(V) \le \tau \ln M_r.\end{aligned} \tag{42}$$

The entropy of O is the discrete entropy of its finite conditional support given V. Taking the smaller bound proves the proposition. When $\tau = 0$, the transcript and conditional output are deterministic, all these information quantities vanish, and the same reasoning applies without a separate limiting argument. □

D.4.4 INDEPENDENT-SAMPLE INFORMATION AND THE EXPECTED-GAP THEOREM

Theorem 2 (Causal information budget and expected generalization)**.** *Under Assumption 1, define the transcript information $\mathcal{I}_\tau$ and output budget $\mathcal{K}_O$ by*

$$\begin{aligned} \mathcal{I}_\tau &:= I(D_F; c_{1:\tau} \mid V) = \sum_{t=1}^{\tau} \{H(c_t \mid V, c_{<t}) - \mathbb{E}h_2(\rho_t)\}, \\ \mathcal{K}_O &:= \min\{\mathcal{I}_\tau,\ \mathbb{E}\ln N_\tau(V)\}. \end{aligned} \tag{43}$$

Here ρ_t is the rounding probability in (3)*; the expectation includes the dataset and the adaptive history. Then*

$$\begin{aligned} I(D_F; O \mid V) \le \mathcal{K}_O \le \mathcal{I}_\tau \le \tau \ln M_r - \sum_{t=1}^{\tau} \mathbb{E}h_2(\rho_t) \le \tau \ln M_r, \\ \mathrm{kl}(\mathbb{E}\widehat{\mu}(O;V) \,\|\, \mathbb{E}\mu(O;V)) \le \frac{I(D_F; O \mid V)}{n} \le \frac{\mathcal{K}_O}{n}, \\ |\mathbb{E}G| \le \min\left\{1, \sqrt{\frac{\mathcal{K}_O}{2n}}\right\}. \end{aligned} \tag{44}$$

The binary-KL inequality retains the mean-dependent constraint before its quadratic relaxation. In particular, the final ICME bank satisfies

$$\left|\mathbb{E}\big[\widehat{J}_F(\mathcal{M}_T) - J(\mathcal{M}_T)\big]\right| \le \min\left\{1, \sqrt{\frac{(T-1)\ln(2^r-1)}{2n}}\right\}. \tag{45}$$

Lemma 3 (Conditional Bernoulli reduction)**.** *Let $D \sim \mathcal{P}^n$ be independent of V, let O have finite conditional support, and let $f_v(o,z) \in [0,1]$ be fixed conditional on V. Define*

$$b(o,v) = \mathbb{E}[\widehat{\mu}(O;V) \mid O = o, V = v], \qquad q(o,v) = \mu(o;v). \tag{46}$$

Then

$$n\,\mathbb{E}_{O,V}\,\mathrm{kl}\big(b(O,V) \| q(O,V)\big) \le I(D; O \mid V). \tag{47}$$

Proof. Fix v outside a null set. Independence ensures that Z_i is independent of $Z_{<i}$ under this conditional law. By two applications of the mutual-information chain rule,

$$\begin{aligned} I(O; Z_i \mid Z_{<i}, V = v) &= I(O, Z_{<i}; Z_i \mid V = v) - I(Z_{<i}; Z_i \mid V = v) \\ &= I(O, Z_{<i}; Z_i \mid V = v) \\ &= I(O; Z_i \mid V = v) + I(Z_{<i}; Z_i \mid O, V = v) \\ &\ge I(O; Z_i \mid V = v), \\ I(O; D \mid V = v) = \sum_{i=1}^{n} I(O; Z_i \mid Z_{<i}, V = v) &\ge \sum_{i=1}^{n} I(O; Z_i \mid V = v). \end{aligned} \tag{48}$$

Nonnegativity of the discarded conditional mutual informations proves the inequality even when conditioning on O induces dependence.

Let $P_{i,v} = \mathcal{L}(O, Z_i \mid V = v)$, let $P_{O,v} = \mathcal{L}(O \mid V = v)$, and define

$$\overline{P}_v = \frac{1}{n}\sum_{i=1}^{n} P_{i,v}, \qquad Q_v = P_{O,v} \otimes \mathcal{P}. \tag{49}$$

Joint convexity from Lemma 1 yields

$$\begin{aligned}\frac{1}{n} I(O; D \mid V = v) &\geq \frac{1}{n} \sum_{i=1}^{n} \mathrm{KL}(P_{i,v} \| Q_v) \\ &\geq \mathrm{KL}\left(\frac{1}{n} \sum_{i=1}^{n} P_{i,v} \,\middle\|\, \frac{1}{n} \sum_{i=1}^{n} Q_v \right) = \mathrm{KL}(\overline{P}_v \| Q_v).\end{aligned} \tag{50}$$

Now apply the common Markov kernel that preserves o and, given (o, z), produces an auxiliary $B \in \{0, 1\}$ with probability $f_v(o, z)$ of $B = 1$. This auxiliary coin is used only in the proof. Under $\overline{P}_v$, its conditional mean given o is $b(o, v)$; under Q_v, it is $q(o, v)$. Both induced laws have the same O marginal $P_{O,v}$. Data processing followed by the KL chain rule therefore gives

$$\begin{aligned}\mathrm{KL}(\overline{P}_v \| Q_v) &\geq \mathrm{KL}(P_{O,v}(do)\,\mathrm{Ber}(b(o,v))(dB) \,\|\, P_{O,v}(do)\,\mathrm{Ber}(q(o,v))(dB)) \\ &= \mathrm{KL}(P_{O,v} \| P_{O,v}) + \mathbb{E}_{O|V=v}\, \mathrm{kl}\big(b(O,v) \| q(O,v)\big) \\ &= \mathbb{E}_{O|V=v}\, \mathrm{kl}\big(b(O,v) \| q(O,v)\big).\end{aligned} \tag{51}$$

Combine (50) and (51), multiply by n, and integrate over V. The right side becomes the conditional mutual information by its definition, proving (47). All quantities here are finite in the application by Proposition 1; boundary Bernoulli means are covered by the conventions of (30). □

Complete proof of Theorem 2. Proposition 1 proves the budget identity and its upper bounds. Lemma 3 applies because Assumption 1 supplies independent examples conditional on V, and causal reconstruction gives a finite output family conditional on V. Joint convexity and the tower property now give the mean-dependent inequality

$$\begin{aligned}\mathrm{kl}(\mathbb{E}\widehat{\mu}(O;V) \,\|\, \mathbb{E}\mu(O;V)) &= \mathrm{kl}(\mathbb{E}b(O,V) \,\|\, \mathbb{E}q(O,V)) \\ &\leq \mathbb{E}\, \mathrm{kl}\big(b(O,V) \| q(O,V)\big) \\ &\leq \frac{I(D; O \mid V)}{n} \leq \frac{\mathcal{K}_O}{n}.\end{aligned} \tag{52}$$

One may retain this binary-KL constraint and invert it numerically without a quadratic relaxation. To obtain an explicit distribution-free bound, the conditional form also gives

$$\begin{aligned}|\mathbb{E}G| &= \big|\mathbb{E}\big[b(O,V) - q(O,V)\big]\big| \\ &\leq \mathbb{E}\,|b(O,V) - q(O,V)| \\ &\leq \mathbb{E}\sqrt{\frac{1}{2}\,\mathrm{kl}\big(b(O,V) \| q(O,V)\big)} \\ &\leq \sqrt{\frac{1}{2}\mathbb{E}\,\mathrm{kl}\big(b(O,V) \| q(O,V)\big)} \\ &\leq \sqrt{\frac{I(D; O \mid V)}{2n}} \leq \sqrt{\frac{\mathcal{K}_O}{2n}}.\end{aligned} \tag{53}$$

The second inequality uses (31); the third uses concavity of the square root. Since both scores belong to $[0, 1]$, $|\mathbb{E}G| \leq 1$ as well. Finally, for $O = \mathcal{M}_T$, Lemma 2 sets $\tau = T - 1$, and Proposition 1 gives $\mathcal{K}_O \leq (T - 1) \ln M_r$. This proves (45), including the case $T = 1$, when the expected signed gap is exactly zero.

The argument averages the conditional bias $b(O, V) - q(O, V)$ in (53). The realized gap G also contains sampling fluctuations, which persist for data-independent outputs. Thus the theorem bounds expected signed optimism around $\mu(O; V)$; absolute deviations and optimization error are separate quantities. □

Complete proof of Theorem 1. Assumption 1 formalizes the main-text conditions. Set $O = \mathcal{M}_T$ and $\tau = T - 1$. Prefix reconstruction and Proposition 1 yield

$$\begin{aligned} 0 \le I(D_F; \mathcal{M}_T \mid V) &\le I(D_F; c_{1:T-1} \mid V) \\ &= \sum_{t=1}^{T-1} [H(c_t \mid V, c_{<t}) - \mathbb{E}h_2(\rho_t)] \\ &\le (T-1)\ln(2^r - 1) - \sum_{t=1}^{T-1} \mathbb{E}h_2(\rho_t) = \kappa_T \\ &\le (T-1)\ln(2^r - 1). \end{aligned} \tag{54}$$

Thus κ_T is nonnegative. Lemma 3 and the argument in (53) then give

$$\begin{aligned} \left|\mathbb{E}[\widehat{J}_F(\mathcal{M}_T) - J(\mathcal{M}_T)]\right| &\le \sqrt{\frac{I(D_F; \mathcal{M}_T \mid V)}{2n}} \\ &\le \sqrt{\frac{\kappa_T}{2n}} \le \sqrt{\frac{(T-1)\ln(2^r-1)}{2n}}. \end{aligned} \tag{55}$$

Both utilities lie in $[0, 1]$, so clipping these bounds at 1 proves (8). When $T = 1$, all sums are empty, $\kappa_T = 0$, and the bank is independent of D_F conditional on V, so the expected signed gap is zero as claimed. □

D.4.5 A REFINEMENT USING ZERO-CODE PROBABILITIES

Proposition 2 (Zero-code entropy refinement)**.** *Let $K = 2^{r-1} - 1$ be the code of zero decoded reward, and define the conditional and marginal nonzero probabilities*

$$a_t(V, C_{<t}) = \Pr(c_t \ne K \mid V, C_{<t}), \qquad \bar{a}_t = \Pr(c_t \ne K) = \mathbb{E}a_t(V, C_{<t}). \tag{56}$$

Then the information in Theorem 2 obeys

$$\begin{aligned} \mathcal{I}_\tau &\le \sum_{t=1}^{\tau} \mathbb{E}\left[h_2(a_t) + a_t \ln(M_r - 1) - h_2(\rho_t)\right] \\ &\le \sum_{t=1}^{\tau} \left[h_2(\bar{a}_t) + \bar{a}_t \ln(M_r - 1) - \mathbb{E}h_2(\rho_t)\right] \\ &\le \tau \ln M_r - \sum_{t=1}^{\tau} \mathbb{E}h_2(\rho_t). \end{aligned} \tag{57}$$

Proof. Write $R_t = \mathbf{1}\{c_t \ne K\}$ and condition on a fixed history $(V, C_{<t}) = (v, c_{<t})$. Because R_t is a function of the code and the zero-code branch contains only one symbol, entropy chain rules give

$$\begin{aligned} H(c_t \mid v, c_{<t}) &= H(c_t, R_t \mid v, c_{<t}) \\ &= H(R_t \mid v, c_{<t}) + H(c_t \mid R_t, v, c_{<t}) \\ &= h_2(a_t) + (1 - a_t)\,0 + a_t H(c_t \mid R_t = 1, v, c_{<t}) \\ &\le h_2(a_t) + a_t \ln(M_r - 1). \end{aligned} \tag{58}$$

Terms multiplied by zero are interpreted as zero. Average this inequality, subtract the exact private entropy from (38), and sum over t to obtain the first line of (57). Concavity of binary entropy, seen from $h_2''(a) = -1/[a(1-a)] < 0$, implies

$$\mathbb{E}h_2(a_t) + \mathbb{E}a_t \ln(M_r - 1) \le h_2(\mathbb{E}a_t) + \mathbb{E}a_t \ln(M_r - 1) = h_2(\bar{a}_t) + \bar{a}_t \ln(M_r - 1). \tag{59}$$

For the last line, compare a Bernoulli mean a to $(M_r - 1)/M_r$:

$$\begin{aligned} h_2(a) + a\ln(M_r - 1) &= \ln M_r - \mathrm{kl}\left(a \,\middle\|\, \frac{M_r - 1}{M_r}\right) \\ &\le \ln M_r. \end{aligned} \tag{60}$$

Combining these steps proves the proposition. The probabilities average over the entire adaptive protocol. Estimation from observed zero counts requires accounting for feedback reuse. Since small nonzero gains can round to zero, $\bar{a}_t$ is the probability of a nonzero code, which can be smaller than the probability of a nonzero exact gain. □

D.4.6 SHARP CHERNOFF EXPONENTS FOR A FIXED BOUNDED SCORE

Lemma 4 (Bernoulli envelope and two-sided KL tail)**.** *Let $X_1, \dots, X_n$ be independent identically distributed variables in $[0,1]$ with mean q, and set $\overline{X} = n^{-1}\sum_i X_i$. For $q \in (0,1)$, $x \in [q,1]$, and $y \in [0,q]$,*

$$\Pr(\overline{X} \ge x) \le e^{-n\,\mathrm{kl}(x\|q)}, \qquad \Pr(\overline{X} \le y) \le e^{-n\,\mathrm{kl}(y\|q)}. \tag{61}$$

For every $s > 0$, including $q = 0, 1$,

$$\begin{aligned} \Pr\left(\overline{X} > q,\ n\,\mathrm{kl}(\overline{X}\|q) > s\right) &\le e^{-s}, \\ \Pr\left(\overline{X} < q,\ n\,\mathrm{kl}(\overline{X}\|q) > s\right) &\le e^{-s}, \\ \Pr\left(n\,\mathrm{kl}(\overline{X}\|q) > s\right) &\le 2e^{-s}. \end{aligned} \tag{62}$$

Proof. For every real λ, convexity of $x \mapsto e^{\lambda x}$ on $[0,1]$ bounds it by the chord through the endpoints. Thus

$$\begin{aligned} e^{\lambda x} &\le (1-x)e^0 + xe^\lambda, \qquad 0 \le x \le 1, \\ \mathbb{E}e^{\lambda X_i} &\le 1 - q + qe^\lambda, \\ \mathbb{E}e^{\lambda \sum_{i=1}^n X_i} &= \prod_{i=1}^n \mathbb{E}e^{\lambda X_i} \le (1-q+qe^\lambda)^n. \end{aligned} \tag{63}$$

For $q < x < 1$, Markov's inequality holds for every $\lambda > 0$. Optimizing over those valid bounds gives the chain

$$\begin{aligned} \Pr(\overline{X} \ge x) &\le \inf_{\lambda > 0} e^{-n\lambda x}\mathbb{E}e^{\lambda \sum_i X_i} \\ &\le \inf_{\lambda > 0} \exp\big(n\big[\ln(1-q+qe^\lambda) - \lambda x\big]\big) \\ &= \exp\left(-n\left[x\ln\frac{x}{q} + (1-x)\ln\frac{1-x}{1-q}\right]\right), \end{aligned} \tag{64}$$

Here the last infimum is attained at $\lambda_x = \ln(x(1-q)/(q(1-x))) > 0$. Differentiating verifies both optimality and the value of the exponent:

$$\begin{aligned} \frac{d}{d\lambda}\big[\ln(1-q+qe^\lambda) - \lambda x\big] &= \frac{qe^\lambda}{1-q+qe^\lambda} - x, \\ \frac{d^2}{d\lambda^2}\ln(1-q+qe^\lambda) &= \frac{q(1-q)e^\lambda}{(1-q+qe^\lambda)^2} > 0, \\ 1 - q + qe^{\lambda_x} &= \frac{1-q}{1-x}, \\ \ln(1-q+qe^{\lambda_x}) - \lambda_x x &= -x\ln\frac{x}{q} - (1-x)\ln\frac{1-x}{1-q}. \end{aligned} \tag{65}$$

The case $x = q$ follows from the trivial bound 1, and $x = 1$ follows by sending λ to $+\infty$. For $0 < y < q$, choose $\lambda_y = \ln(y(1-q)/(q(1-y))) < 0$. The direction of the exponential comparison reverses, yielding

$$\begin{aligned} \Pr(\overline{X} \le y) &= \Pr\left(e^{\lambda_y \sum_i X_i} \ge e^{n\lambda_y y}\right) \\ &\le e^{-n\lambda_y y}\mathbb{E}e^{\lambda_y \sum_i X_i} \\ &\le \exp\big(n\big[\ln(1-q+qe^{\lambda_y}) - \lambda_y y\big]\big) \\ &= e^{-n\,\mathrm{kl}(y\|q)}. \end{aligned} \tag{66}$$

Again $y = q$ is trivial and $y = 0$ follows by a limit. These calculations prove (61).

To obtain the KL-level events, observe that

$$\frac{\partial}{\partial x}\,\mathrm{kl}(x\|q) = \ln\frac{x(1-q)}{q(1-x)} \begin{cases} < 0, & x < q, \\ = 0, & x = q, \\ > 0, & x > q. \end{cases} \tag{67}$$

If the upper-side event in (62) is nonempty, its lower threshold $x_s \in (q, 1)$ satisfies $n\,\mathrm{kl}(x_s\|q) = s$; the event is contained in $\{\overline{X} \geq x_s\}$. If s reaches or exceeds the upper endpoint divergence, the strict event is empty. The upper tail bound therefore gives probability at most e^{-s}. The decreasing lower branch gives the second statement by the same threshold argument. The branches are disjoint, and their union contains the two-sided strict event, proving the last statement. If $q = 0$ or $q = 1$, boundedness forces every $X_i = q$ almost surely, so all three events are empty. Bernoulli scores attain the envelope, preserving the full binary-KL exponent. □

D.4.7 Output-weighted high-probability certificates

Theorem 3 (Output-weighted KL certificate)**.** *Under Assumption 1, let weights $\pi_v(o) > 0$, $\sum_o \pi_v(o) \leq 1$, be fixed before observing D_F, conditional on $V = v$. For $\delta \in (0, 1)$, with probability at least $1 - \delta$, simultaneously for all $o \in \mathcal{O}_\tau(V)$ and the selected output O,*

$$\begin{aligned} n\,\mathrm{kl}(\widehat{\mu}(o; V)\,\|\,\mu(o; V)) &\leq \ln \frac{2}{\delta \pi_V(o)}, \\ |G| &\leq \min\left\{1, \sqrt{\frac{\ln(1/\pi_V(O)) + \ln(2/\delta)}{2n}}\right\}. \end{aligned} \tag{68}$$

For a one-sided optimism certificate, the factor 2 inside the logarithm is unnecessary: with probability at least $1 - \delta$, either $G \leq 0$, or $n\,\mathrm{kl}(\widehat{\mu}(O; V)\|\mu(O; V)) \leq \ln(1/(\delta\pi_V(O)))$.

Complete proof of Theorem 3. Fix $V = v$. The finite family $\mathcal{O}_\tau(v)$ and its weights are now fixed, while D still has law $\mathcal{P}^n$. For every fixed o in this family, the variables $f_v(o, Z_i)$ are i.i.d., bounded, and have mean $\mu(o; v)$. Let

$$s_v(o) = \ln \frac{2}{\delta \pi_v(o)}, \qquad E_{v,o} = \left\{n\,\mathrm{kl}\left(\widehat{\mu}(o; v)\|\mu(o; v)\right) > s_v(o)\right\}. \tag{69}$$

Lemma 4 and a finite weighted union bound give

$$\begin{aligned} \Pr\left(\bigcup_{o \in \mathcal{O}_\tau(v)} E_{v,o} \,\middle|\, V = v\right) &\leq \sum_{o \in \mathcal{O}_\tau(v)} \Pr(E_{v,o} \mid V = v) \\ &\leq \sum_{o \in \mathcal{O}_\tau(v)} 2e^{-s_v(o)} \\ &= \delta \sum_{o \in \mathcal{O}_\tau(v)} \pi_v(o) \leq \delta. \end{aligned} \tag{70}$$

On the complement, the KL inequality holds for *every* candidate, including the one selected adaptively using D and the evaluator's private uniforms. Since this event depends only on (D, V), adjoining the independent private uniforms does not change its probability. Integrating the conditional failure probability gives

$$\begin{aligned} \Pr\left(n\,\mathrm{kl}\left(\widehat{\mu}(O; V)\|\mu(O; V)\right) > \ln \frac{2}{\delta \pi_V(O)}\right) &\leq \mathbb{E}_V \Pr\left(\bigcup_o E_{V,o} \,\middle|\, V\right) \\ &\leq \mathbb{E}_V \delta = \delta. \end{aligned} \tag{71}$$

On the successful event, Lemma 1 yields the explicit deviation bound through the successive inequalities

$$\begin{aligned} 2n|G|^2 &\leq n\,\mathrm{kl}\left(\widehat{\mu}(O; V)\|\mu(O; V)\right) \\ &\leq \ln \frac{2}{\delta \pi_V(O)} = \ln \frac{1}{\pi_V(O)} + \ln \frac{2}{\delta}, \\ |G| &\leq \min\left\{1, \sqrt{\frac{\ln(1/\pi_V(O)) + \ln(2/\delta)}{2n}}\right\}. \end{aligned} \tag{72}$$

For the one-sided version replace $E_{v,o}$ by its intersection with $\{\widehat{\mu}(o; v) > \mu(o; v)\}$ and use threshold $\ln(1/(\delta\pi_v(o)))$. The first line of (62) contributes the one-sided tail e^{-s}, and every other step of (70)–(71) is unchanged. This proves the theorem. □

Inverting the KL certificate without Pinsker's relaxation. For an observed empirical score x and certificate level s, define

$$\mathcal{C}_n(x, s) = \{q \in [0, 1] : n\,\mathrm{kl}(x\|q) \le s\}. \tag{73}$$

Theorem 3 places $\mu(O; V)$ in this set at $x = \widehat{\mu}(O; V)$ and $s = \ln(2/(\delta\pi_V(O)))$. For $x \in (0, 1)$, differentiation in q gives $(q - x)/(q(1 - q))$, so the sublevel set is an interval with endpoints found by monotone scalar root finding on either side of x. At empirical endpoints the interval has the exact forms

$$\mathcal{C}_n(0, s) = [0, 1 - e^{-s/n}], \qquad \mathcal{C}_n(1, s) = [e^{-s/n}, 1]. \tag{74}$$

These follow by substituting $\mathrm{kl}(0\|q) = -\ln(1 - q)$ and $\mathrm{kl}(1\|q) = -\ln q$. At an all-success empirical score, the lower confidence limit therefore has a local s/n scale. This refinement is specific to the observed endpoint score.

D.4.8 Sample complexity and certified paired improvement

Corollary 1 (Final-bank sample complexity)**.** *Set $b_T = (T - 1)\ln(2^r - 1)$. For $\epsilon, \delta \in (0, 1)$, sufficient feedback sample sizes are*

$$\begin{aligned} n \ge \max\left\{1, \left\lceil \frac{b_T}{2\epsilon^2} \right\rceil\right\} \quad &\Longrightarrow \quad \left|\mathbb{E}\big[\widehat{J}_F(\mathcal{M}_T) - J(\mathcal{M}_T)\big]\right| \le \epsilon, \\ n \ge \left\lceil \frac{b_T + \ln(2/\delta)}{2\epsilon^2} \right\rceil \quad &\Longrightarrow \quad \Pr\Big(\left|\widehat{J}_F(\mathcal{M}_T) - J(\mathcal{M}_T)\right| > \epsilon\Big) \le \delta. \end{aligned} \tag{75}$$

In the second statement, $\ln(2/\delta)$ can be replaced by $\ln(1/\delta)$ if only optimistic bias is controlled. The realized certificate can use $\ln N_{T-1}(V)$ in place of b_T.

The first statement bounds an expected signed gap; the second controls an individual run. The per-run certificate uses the output-count penalty to cover all possible outputs simultaneously.

Complete proof of Corollary 1. For the final bank, $\tau = T - 1$ by Lemma 2. Let $b_T = (T - 1)\ln M_r$. The expected-gap theorem immediately gives

$$n \ge \frac{b_T}{2\epsilon^2} \quad \Longrightarrow \quad \frac{\mathcal{K}_{\mathcal{M}_T}}{2n} \le \frac{b_T}{2n} \le \epsilon^2 \quad \Longrightarrow \quad \left|\mathbb{E}[\widehat{J}_F(\mathcal{M}_T) - J(\mathcal{M}_T)]\right| \le \epsilon. \tag{76}$$

The maximum with 1 enforces $n \ge 1$. For the per-run result set $\pi_v(o) = 1/N_{T-1}(v)$. With probability at least $1 - \delta$,

$$\begin{aligned} |\widehat{J}_F(\mathcal{M}_T) - J(\mathcal{M}_T)| &\le \min\left\{1, \sqrt{\frac{\ln N_{T-1}(V) + \ln(2/\delta)}{2n}}\right\} \\ &\le \min\left\{1, \sqrt{\frac{(T - 1)\ln M_r + \ln(2/\delta)}{2n}}\right\} \\ &\le \epsilon \quad \text{if } n \ge \frac{b_T + \ln(2/\delta)}{2\epsilon^2}. \end{aligned} \tag{77}$$

Taking ceilings yields the stated integer sample requirements. Applying the one-sided part of Theorem 3 gives the analogous result with $\ln(1/\delta)$. The simultaneous guarantee depends on the worst-case output family. □

Proposition 3 (Population improvement over a fixed baseline)**.** *Let $\mathcal{M}_0(V)$ be a baseline bank fixed independently of D conditional on V, and let $O = \mathcal{M}_T$. Write $\widehat{D}(O) = \widehat{J}_F(O) - \widehat{J}_F(\mathcal{M}_0(V))$ and $D_\star(O) = J(O) - J(\mathcal{M}_0(V))$. For any weights as in Theorem 3, with probability at least $1 - \delta$,*

$$D_\star(O) \ge \widehat{D}(O) - \min\left\{2, \sqrt{\frac{2\ln(1/(\delta\pi_V(O)))}{n}}\right\}. \tag{78}$$

In particular, $n \ge \lceil 2(b_T + \ln(1/\delta))/\epsilon^2 \rceil$ suffices to make this one-sided paired deviation at most ϵ with uniform weights.

Proof. For each fixed (o, v) transform the paired score into a bounded score:

$$\begin{aligned} f_v^{\mathrm{pair}}(o,z) &= \frac{u(o,z) - u(\mathcal{M}_0(v),z) + 1}{2} \in [0,1], \\ \widehat{\mu}_{\mathrm{pair}}(o;v) &= \frac{\widehat{D}(o)+1}{2}, \qquad \mu_{\mathrm{pair}}(o;v) = \frac{D_\star(o)+1}{2}, \\ \widehat{\mu}_{\mathrm{pair}}(o;v) - \mu_{\mathrm{pair}}(o;v) &= \frac{\widehat{D}(o) - D_\star(o)}{2}. \end{aligned} \tag{79}$$

Theorem 3 applies to this fixed score family. On its one-sided successful event either $\widehat{D}(O) \le D_\star(O)$, in which case the conclusion is immediate, or

$$\begin{aligned} \frac{n}{2}\big(\widehat{D}(O) - D_\star(O)\big)^2 &= 2n\big(\widehat{\mu}_{\mathrm{pair}}(O;V) - \mu_{\mathrm{pair}}(O;V)\big)^2 \\ &\le n\,\mathrm{kl}\left(\widehat{\mu}_{\mathrm{pair}}(O;V) \| \mu_{\mathrm{pair}}(O;V)\right) \\ &\le \ln \frac{1}{\delta \pi_V(O)}. \end{aligned} \tag{80}$$

Taking the positive square root and using the deterministic range bound $\widehat{D}(O) - D_\star(O) \le 2$ proves (78). With uniform weights, $\ln(1/\pi_V(O)) = \ln N_{T-1}(V) \le b_T$; solving $2(b_T + \ln(1/\delta))/n \le \epsilon^2$ proves the sample statement. Evaluating both banks on the same tasks preserves the dependence of their paired scores. A positive population improvement is certified when the empirical improvement exceeds its confidence radius. □

D.4.9 Moment tools for the lower bound

Lemma 5 (First-moment and event lower bounds)**.** *If X has finite fourth moment and $\mathbb{E}X^2 > 0$, then*

$$\mathbb{E}|X| \ge \frac{(\mathbb{E}X^2)^{3/2}}{(\mathbb{E}X^4)^{1/2}}. \tag{81}$$

If $Y \ge 0$ has a positive mean and finite second moment, then for $\theta \in (0,1)$,

$$\Pr(Y \ge \theta \mathbb{E}Y) \ge (1-\theta)^2 \frac{(\mathbb{E}Y)^2}{\mathbb{E}Y^2}. \tag{82}$$

Proof. Hölder's inequality with exponents $3/2$ and 3 gives the interpolation

$$\begin{aligned} \mathbb{E}X^2 &= \mathbb{E}\big[|X|^{2/3}|X|^{4/3}\big] \\ &\le \Big(\mathbb{E}(|X|^{2/3})^{3/2}\Big)^{2/3} \Big(\mathbb{E}(|X|^{4/3})^3\Big)^{1/3} \\ &= (\mathbb{E}|X|)^{2/3}(\mathbb{E}X^4)^{1/3}, \\ (\mathbb{E}X^2)^{3/2} &\le \mathbb{E}|X|\,(\mathbb{E}X^4)^{1/2}. \end{aligned} \tag{83}$$

Division proves (81). For the event inequality let $E = \{Y \ge \theta \mathbb{E}Y\}$. Splitting its expectation and applying Cauchy–Schwarz on E yields

$$\begin{aligned} \mathbb{E}Y &= \mathbb{E}[Y \mathbf{1}_{E^c}] + \mathbb{E}[Y \mathbf{1}_E] \\ &\le \theta \mathbb{E}Y + \sqrt{\mathbb{E}Y^2\,\mathbb{E}\mathbf{1}_E^2} \\ &= \theta \mathbb{E}Y + \sqrt{\mathbb{E}Y^2 \Pr(E)}, \\ (1-\theta)^2(\mathbb{E}Y)^2 &\le \mathbb{E}Y^2 \Pr(E). \end{aligned} \tag{84}$$

Division by the positive second moment proves (82). □

D.4.10 CAPACITY LOWER BOUND: STATEMENT AND COMPLETE CONSTRUCTION

Theorem 4 (Lower bound for the finite-feedback interface)**.** *Let $m \geq 2$, $\tau \geq 1$, and $d = \lfloor \tau \log_2 m \rfloor$. There exist an i.i.d. task distribution, 2^d outputs with utilities in $\{0, 1\}$, and a learner receiving only τ symbols from an m-symbol evaluator such that its generalization gap G satisfies*

$$\begin{aligned} \mathbb{E}G &\geq \frac{1}{2}\sqrt{\frac{d}{3(n-1)+d}} \geq \frac{1}{4}\min\left\{1, \sqrt{\frac{d}{n}}\right\}, \\ \Pr\left(G \geq \frac{1}{8}\min\left\{1, \sqrt{\frac{d}{n}}\right\}\right) &\geq \frac{1}{16}. \end{aligned} \tag{85}$$

Thus the dependence on $\tau \ln m$ and n is unimprovable up to universal constants for guarantees based solely on a finite-feedback interface, including at fixed confidence. The construction permits arbitrary finite-feedback encoders and learners. The rate comparison is up to universal constants at fixed confidence; the $\ln(1/\delta)$ dependence is treated only by the upper bound.

Complete proof of Theorem 4. Let $d = \lfloor \tau \log_2 m \rfloor \geq 1$. A task is $Z = (J, Y)$, where J is uniform on $\{1, \ldots, d\}$ and Y is an independent uniform sign in $\{-1, +1\}$. Outputs are sign vectors $a \in \{-1, +1\}^d$. Fix the scoring rule

$$f(a, (j, y)) = \mathbf{1}\{a_j = y\} = \frac{1 + a_j y}{2}, \qquad \mu(a) = \mathbb{E}f(a, Z) = \frac{1}{2} \quad \text{for every } a. \tag{86}$$

For a sample $D = ((J_i, Y_i))_{i=1}^n$, define coordinate imbalances S_j and an empirical-majority output by

$$S_j = \sum_{i=1}^n \mathbf{1}\{J_i = j\} Y_i, \qquad \widehat{a}_j = \begin{cases} +1, & S_j \geq 0, \\ -1, & S_j < 0. \end{cases} \tag{87}$$

The evaluator computes $\widehat{a}$. There are $2^d \leq m^\tau$ such vectors, so fix in advance an injection from these vectors into length-τ strings over the m-symbol alphabet. The evaluator sends that string, and the learner decodes and outputs $\widehat{a}$. Nothing else is transmitted. This is a fixed-horizon finite-feedback protocol with constant V, and it proves that the desired output can be selected within the prescribed channel budget. The evaluator encodes the empirical-majority output directly.

Exact expression for the optimism. Population utility remains $1/2$ after any choice of output. On the sample, however, the signs maximize the coordinatewise contribution:

$$\begin{aligned} \widehat{\mu}(\widehat{a}) &= \frac{1}{n}\sum_{i=1}^n \frac{1 + \widehat{a}_{J_i} Y_i}{2} \\ &= \frac{1}{2} + \frac{1}{2n}\sum_{j=1}^d \widehat{a}_j \sum_{i=1}^n \mathbf{1}\{J_i = j\} Y_i \\ &= \frac{1}{2} + \frac{1}{2n}\sum_{j=1}^d |S_j|, \\ G = \widehat{\mu}(\widehat{a}) - \mu(\widehat{a}) &= \frac{1}{2n}\sum_{j=1}^d |S_j| \in [0, 1/2]. \end{aligned} \tag{88}$$

The upper range bound follows from $\sum_j |S_j| \leq \sum_i |Y_i| = n$. The calculation allows dependence among different S_j.

Second and fourth moments of one coordinate. For fixed j, let $X_{ij} = \mathbf{1}\{J_i = j\} Y_i$. Across i these variables are independent, symmetric, and satisfy

$$\mathbb{E}X_{ij} = \mathbb{E}X_{ij}^3 = 0, \qquad \mathbb{E}X_{ij}^2 = \mathbb{E}X_{ij}^4 = \frac{1}{d}. \tag{89}$$

Expand the square and fourth power of $S_j = \sum_i X_{ij}$. Terms with an odd power of any independent factor have expectation zero. The remaining fourth-order terms have either one repeated index or two pairs, giving

$$\begin{aligned}\mathbb{E}S_j^2 &= \sum_{i=1}^{n} \mathbb{E}X_{ij}^2 + 2\sum_{i<k} \mathbb{E}X_{ij}\mathbb{E}X_{kj} = \frac{n}{d},\\ \mathbb{E}S_j^4 &= \sum_{i=1}^{n} \mathbb{E}X_{ij}^4 + 6\sum_{i<k} \mathbb{E}X_{ij}^2\mathbb{E}X_{kj}^2 \\ &= \frac{n}{d} + \frac{3n(n-1)}{d^2} \le \frac{n}{d} + 3\left(\frac{n}{d}\right)^2.\end{aligned} \tag{90}$$

Lemma 5 now lower-bounds each absolute imbalance. Keeping the exact moment first, and then relaxing only $n(n-1) \le n^2$, yields

$$\begin{aligned}\mathbb{E}|S_j| &\ge \frac{(n/d)^{3/2}}{\sqrt{n/d + 3n(n-1)/d^2}}\\ &\ge \frac{(n/d)^{3/2}}{\sqrt{n/d + 3(n/d)^2}} = \frac{n/d}{\sqrt{1+3n/d}},\\ \mathbb{E}G &= \frac{1}{2n}\sum_{j=1}^{d} \mathbb{E}|S_j| \\ &\ge \frac{d}{2n}\frac{(n/d)^{3/2}}{\sqrt{n/d + 3n(n-1)/d^2}} = \frac{1}{2}\sqrt{\frac{d}{3(n-1)+d}}\\ &\ge \frac{1}{2}\sqrt{\frac{d}{3n+d}}.\end{aligned} \tag{91}$$

If $d \le n$, then $3n + d \le 4n$; if $d \ge n$, then $3n + d \le 4d$. Consequently,

$$\frac{1}{2}\sqrt{\frac{d}{3n+d}} \ge \begin{cases}\frac{1}{4}\sqrt{d/n}, & d \le n,\\ \frac{1}{4}, & d \ge n,\end{cases} \quad = \quad \frac{1}{4}\min\{1, \sqrt{d/n}\}. \tag{92}$$

This proves the expectation part of (85).

A constant-probability lower bound. Cauchy–Schwarz for the finite vector $(|S_1|, \dots, |S_d|)$ and (90) imply

$$\begin{aligned}\mathbb{E}G^2 &= \frac{1}{4n^2}\mathbb{E}\left(\sum_{j=1}^{d}|S_j|\right)^2\\ &\le \frac{d}{4n^2}\sum_{j=1}^{d}\mathbb{E}S_j^2 = \frac{d}{4n},\\ \mathbb{E}G^2 &\le \frac{1}{4}\min\{1, d/n\},\end{aligned} \tag{93}$$

where the last line also uses $G \le 1/2$. By (91), the squared-mean to second-moment ratio satisfies

$$\begin{aligned}\frac{(\mathbb{E}G)^2}{\mathbb{E}G^2} &\ge \frac{d/(3n+d)}{\min\{1, d/n\}}\\ &= \begin{cases}n/(3n+d), & d \le n,\\ d/(3n+d), & d \ge n\end{cases} \ge \frac{1}{4}.\end{aligned} \tag{94}$$

Apply (82) with $Y = G$ and $\theta = 1/2$, then use (92):

$$\begin{aligned}\Pr\left(G \ge \frac{1}{8}\min\{1, \sqrt{d/n}\}\right) &\ge \Pr\left(G \ge \tfrac{1}{2}\mathbb{E}G\right)\\ &\ge \frac{1}{4}\frac{(\mathbb{E}G)^2}{\mathbb{E}G^2} \ge \frac{1}{16}.\end{aligned} \tag{95}$$

This proves the probability statement and completes the theorem. □

Rate optimality for finite-feedback protocols. Since $d = \lfloor \tau \log_2 m \rfloor$ is within a constant factor of $\tau \ln m$ when $m \geq 2$ and $\tau \geq 1$, the upper and lower rates match up to universal constants at fixed confidence. More explicitly, any expected-gap guarantee valid for *every* protocol in this interface class and every task distribution, with tolerance $\epsilon \in (0, 1/4)$, must meet the necessary condition

$$\begin{aligned} \epsilon \geq \frac{1}{2}\sqrt{\frac{d}{3(n-1)+d}} \quad &\Longrightarrow \quad 4\epsilon^2(3(n-1)+d) \geq d \\ &\Longrightarrow \quad n \geq 1 + \frac{d(1-4\epsilon^2)}{12\epsilon^2}. \end{aligned} \tag{96}$$

Thus the ϵ^{-2} and capacity dependence of Corollary 1 is sharp up to constants for the full interface class. ICME further restricts paired utility, stochastic rounding, reachable memory states, and parameter updates. The matching-rate result applies at fixed confidence; behavior as δ approaches zero is covered by the upper bound.

D.4.11 Interpretation and numerical scale

The causal refinement replaces $Tr \ln 2$ by $(T-1)\ln(2^r - 1)$ for the predetermined final bank; the full learned pair still has prefix length T. Entropy subtraction can further reduce an expected-information bound, and merging output-equivalent transcripts can improve both information and finite-family certificates. The actual fixed-length communication cost remains Tr.

At $T = 192$ and $r = 4$, the bank's worst-case information allowance is $191 \ln 15 \approx 517.24$ nats. The explicit expected-gap bound at $n = 12$ is approximately 4.642, and is clipped to 1. For tolerance 0.05, the sufficient expected-gap sample size is $\lceil 191 \ln 15/(2 \cdot 0.05^2) \rceil = 103{,}448$; the two-sided 95% certificate adds $\ln 40$ in the numerator and requires $104{,}186$ tasks. These sample sizes are conservative sufficient conditions for the stated distribution-free certificates.

Statistical independence of the feedback sample and initial learner resources is required in addition to disjoint task IDs. Repeated runs and feedback-dependent model selection require accounting for their full transcripts. Selecting a bank using the final code likewise requires the full prefix. The isolated interface also keeps timing, evaluator diagnostics, and interventions based on the reused dataset outside the learner's inputs. The theorem uses the unconditional law of the specified fixed-horizon score computation.

For $O = \mathcal{M}_T$, the result controls stored-bank utility on the feedback task distribution. An analogous theorem for ϕ_T building a new bank requires a bounded stream-level scoring functional and an independent sampling model for those streams.