INFO-SKILL: A STATE-CONDITIONAL INFORMATION BOTTLENECK FOR SKILL-AUGMENTED AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Skill-augmented language agents often retrieve procedural skills from a persistent library, but most existing methods inject each retrieved skill as full text. This ignores a basic property of procedural guidance: the useful part of a skill depends on the current decision state. As a result, full-text prompting wastes context, exposes the policy to irrelevant instructions, and leaves library maintenance without a reliable measure of skill value. We propose INFO-SKILL, a predictive-rate framework that unifies state-conditional skill compression with skill-library evolution. At decision time, INFO-SKILL encodes the retrieved skill with a state-conditional stochastic bottleneck and optimizes outcome prediction under a variational rate penalty, so the same skill can preserve different information at different stages of an interaction. The resulting code is mapped to a fixed number of continuous prefix embeddings, allowing a frozen backbone with lightweight adaptation to use retrieved skills without full-text prompting. At the library level, we use held-out predictive fidelity gain, variational rate, and redundancy to score candidate additions and existing entries, and paired rollout tests to validate each update. This gives a single information-based view of both how a retrieved skill should be used and which skills a library should retain. Experiments on embodied interaction, search-based question answering, and web navigation show that INFO-SKILL improves the strongest benchmark-level metrics by 2.0% on average while reducing the context consumed by retrieved skills by 92.6%, delivering the strongest aggregate performance across all three backbones.[1]

1 INTRODUCTION

LLMs are increasingly used as long-horizon interactive agents that reason, act, and call tools over extended trajectories (Erdogan et al., 2025; Yang et al., 2026a; Hu et al., 2026). To make use of sparse environment feedback, recent work stores reusable skills extracted from past trajectories in libraries (Tang et al., 2026; Ma et al., 2026). During inference, the agent retrieves a skill as procedural guidance and reuses prior experience to improve future decisions (Jiang et al., 2026; Ling et al., 2026). This retrieval-based view is appealing because it lets the agent accumulate reusable experience without forcing every useful behavior into the model parameters.

Most current skill-augmented agents use a retrieved skill by appending its full text to the interaction history (Badhe et al., 2026; Yang et al., 2026d). This design assumes that the same skill content remains useful throughout an episode. In practice, a skill often describes a complete procedure, while the agent may retrieve it after some steps are finished or before later steps become relevant (Yang et al., 2025b; Chen et al., 2026; Guo et al., 2026). Full-text prompting therefore mixes relevant instructions with completed, premature, and locally irrelevant content. This is not just a context-efficiency issue. It also changes the information available to the next decision by forcing the policy to process skill content that does not support the current action. Figure 1(a) shows that the clause that matters most changes with the current state and often aligns with the current task step.

The same limitation appears in library maintenance. Retrieval can surface a broadly relevant skill, but it does not determine which part of that skill matters for the current decision (Wang et al., 2026b;c; Li et al., 2026). Generic prompt compression can shorten the input, but it does not explicitly

[1] Source code and data are included in the supplementary material and will be released upon acceptance.

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.

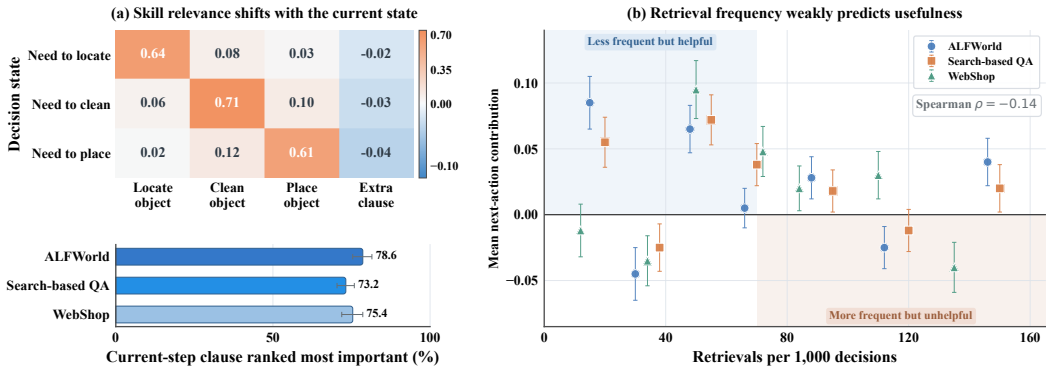

Figure 1: Results from frozen full-text skill-prompting agents on agentic tasks. (a) The clause that matters most changes with the agent state and aligns with the current task step across all three benchmarks. (b) Retrieval frequency is a weak indicator of usefulness at the next decision. Frequently retrieved skills can contribute little or even hurt, while less frequent skills can still be helpful.

preserve the skill information that is most useful for the next action in the current state (Jiang et al., 2024; Li et al., 2025). Existing library update rules rely on frequency, semantics, or reflection signals, but these signals do not indicate whether a skill improves outcome prediction in the states where it is actually used. Figure 1(b) shows that retrieval frequency is a weak proxy for decision utility. A useful skill can be infrequent, and a frequently retrieved skill can be redundant or harmful.

We propose INFO-SKILL, a predictive-rate framework for state-conditional skill use and skill-library evolution. The framework has two tightly connected components. *State-Conditional Skill Compression* encodes the retrieved skill together with the current state and maps the resulting code to a small set of continuous prefix embeddings for a frozen backbone. This gives the policy access to decision-relevant skill information without carrying the full skill text in the prompt. *Predictive-Rate Skill-Library Evolution* uses held-out predictive fidelity gain, variational rate, and redundancy to score candidate additions and existing entries, then applies paired rollout tests before each library update. The first component determines what information the policy keeps from a retrieved skill at each decision step. The second uses the same predictive-rate view to decide which skills the library should keep over time. This connection between skill use and library maintenance is the central design choice of INFO-SKILL.

We evaluate INFO-SKILL on three representative agentic domains: ALFWorld (Shridhar et al., 2021), WebShop (Yao et al., 2022), and Search-based QA (Jin et al., 2025). Across training-free prompting, outcome-only RL, and skill-distillation baselines, INFO-SKILL improves the strongest benchmark-level headline metrics by 2.0% on average and reduces the context positions consumed by retrieved skills by 92.6% on average. It delivers the strongest aggregate performance across all three backbones and remains strong under reduced interaction budgets, unseen-task transfer, and noisy or misleading skills. These results show that skill use and skill-library maintenance both improve when skill value is defined relative to the current state and the information that the policy can actually extract.

2 RELATED WORK

Reinforcement Learning for Agentic LLMs. Outcome-only policy optimization methods improve long-horizon agents from sparse rewards. GRPO (Shao et al., 2024) uses group-relative advantages, Search-R1 extends this setup to multi-turn search (Jin et al., 2025), and RAGEN studies stable training for long-horizon agents (Wang et al., 2025). Skill-GRPO and Skill-GRPO* condition the policy on retrieved skills during training, with only the latter retaining skill access at inference (Wu et al., 2026a). Other work focuses on credit assignment and self-distillation, including Agent Lightning (Luo et al., 2025), SPARK (Wu et al., 2026b), OPSD (Zhao et al., 2026), RLSD (Yang et al., 2026b), SDAR (Lu et al., 2026a), Skill-SD (Wang et al., 2026a), and ADRS (Zhang et al., 2026). These methods improve learning signals for agent training, but they do not address how a retrieved skill should be selectively represented at each decision step or how an external skill library should be updated over time.

Skill-Augmented Language Agents. A large line of work augments language agents with retrieved skills or memories at inference time. Voyager (Wang et al., 2023), Reflexion (Shinn et al., 2023), ExpeL (Zhao et al., 2024), and Skill-Prompt* (Wu et al., 2026a) retrieve reusable experience as direct guidance. Other methods build more structured or evolving memories, such as A-MEM (Xu et al., 2025), GAM (Wu et al., 2026c), and Agentic Plan Caching (Zhang et al., 2025a). A separate line distills retrieved or hindsight skills into the policy, including SKILL0 (Lu et al., 2026b), Skill-SD (Wang et al., 2026a), OPID (Yang et al., 2026c), and ADRS (Zhang et al., 2026). In contrast, INFO-SKILL focuses on two questions that these settings leave open: how to compress a retrieved skill for the current state, and how to maintain a skill library using held-out predictive value and representational cost.

3 PROBLEM SETUP AND PRELIMINARIES

Problem Setup. We consider a partially observable environment $\mathcal{M} = (\mathcal{E}, \mathcal{A}, \mathcal{O}, \mathcal{T}, \Omega, \mathcal{R}, \gamma)$, where $\mathcal{E}$, $\mathcal{A}$, and $\mathcal{O}$ are the latent-state, action, and observation spaces. The transition model, observation model, reward function, and discount factor are $\mathcal{T}(e' \mid e, a)$, $\Omega(o \mid e)$, $\mathcal{R}(e, a) \in \mathbb{R}$, and $\gamma \in [0, 1]$. Each episode is associated with a task instruction $x \in \mathcal{X}$. At step t, the environment is in latent state $e_t \in \mathcal{E}$, emits $o_t \sim \Omega(\cdot \mid e_t)$, receives $a_t \in \mathcal{A}$, and transitions to $e_{t+1} \sim \mathcal{T}(\cdot \mid e_t, a_t)$ with reward $r_t = \mathcal{R}(e_t, a_t)$. The model-visible state is the interaction history $s_t := (x, o_{\le t}, a_{<t}) \in \mathcal{S}$.

The agent maintains a skill library $\mathcal{B} = \{z_j\}_{j=1}^{K} \subset \mathcal{Z}$, where K is the library size and $\mathcal{Z}$ is the space of natural-language skills. Given s_t, the retriever selects a skill $z_t \in \mathcal{B}$, the encoder samples a latent code $\tilde{z}_t \sim p_\theta(\tilde{Z} \mid z_t, s_t)$, and the policy samples an action $a_t \sim \pi_\eta(\cdot \mid s_t, \tilde{z}_t)$, where $\tilde{Z}$ denotes the code variable and $\tilde{z}_t$ its realization. For a fixed library $\mathcal{B}$, the objective is $J(\eta, \theta; \mathcal{B}) := \mathbb{E}[\sum_{t=0}^{H-1} \gamma^t r_t]$, where H is the episode horizon. For G rollouts of the same task, let $R_i = \sum_{t=0}^{H-1} \gamma^t r_t^{(i)}$ be the return of rollout i. We define the normalized outcome target

$$Y_i = \frac{R_i - \bar{R}}{\sqrt{G^{-1} \sum_{j=1}^{G} (R_j - \bar{R})^2 + \varepsilon}}, \tag{1}$$

where $\bar{R} = G^{-1} \sum_{j=1}^{G} R_j$ and $\varepsilon > 0$. This target provides a denser supervision signal than a terminal reward and gives the compression objective access to graded outcome across rollouts.

Preliminaries: Conditional Information Bottleneck. Our goal is to formalize state-conditional skill compression under a fixed data distribution induced by a frozen training window. At library-evolution window k, the current policy, retriever, encoder, and library induce a joint distribution $\rho_{\pi_{\eta_k}, \theta_k, \mathcal{B}_k}(S, Z, \tilde{Z}, A, Y)$. We freeze the behavior policy, retriever, and collected data at the start of the window and obtain a snapshot distribution $p_{\mathcal{D}_k}(s, z, y)$, which we split into disjoint fitting and evaluation folds. We then define the auxiliary joint $P_{k,\theta}(s, z, y, \tilde{z}) := p_{\mathcal{D}_k}(s, z, y)\, p_\theta(\tilde{z} \mid z, s)$.

Under $P_{k,\theta}$, the code is resampled from the frozen pair (z, s), so $Y \perp \tilde{Z} \mid (Z, S)$ holds by construction. This isolates the representation question from on-policy control and lets us ask how much outcome-predictive information the code preserves about a retrieved skill once the current state is given. Let I_k and H_k denote mutual information and entropy under $P_{k,\theta}$. We use $I_k(\tilde{Z}; Y \mid S)$ to measure how much outcome-predictive information the code retains given the state, and $I_k(\tilde{Z}; Z \mid S)$ to measure how strongly the code depends on the retrieved skill beyond the state:

$$I_k(\tilde{Z}; Y \mid S) = \mathbb{E}_{P_{k,\theta}}\left[\log \frac{P_{k,\theta}(Y \mid \tilde{Z}, S)}{p_{\mathcal{D}_k}(Y \mid S)}\right], \quad I_k(\tilde{Z}; Z \mid S) = \mathbb{E}_{P_{k,\theta}}\left[\log \frac{p_\theta(\tilde{Z} \mid Z, S)}{p_{k,\theta}(\tilde{Z} \mid S)}\right], \tag{2}$$

where the snapshot state-conditional code marginal is $p_{k,\theta}(\tilde{z} \mid s) := \sum_{z \in \mathcal{B}_k} p_\theta(\tilde{z} \mid z, s)\, p_{\mathcal{D}_k}(z \mid s)$.

These quantities define the fixed-distribution conditional information bottleneck objective

$$\min_{p_\theta(\tilde{Z} \mid Z, S)} I_k(\tilde{Z}; Z \mid S) \quad \text{s.t.} \quad I_k(\tilde{Z}; Y \mid S) \ge \kappa, \tag{3}$$

where $\kappa \ge 0$ controls how much outcome-predictive information the code is required to preserve. Conditioning on S is essential here: it allows the same retrieved skill to map to different codes at

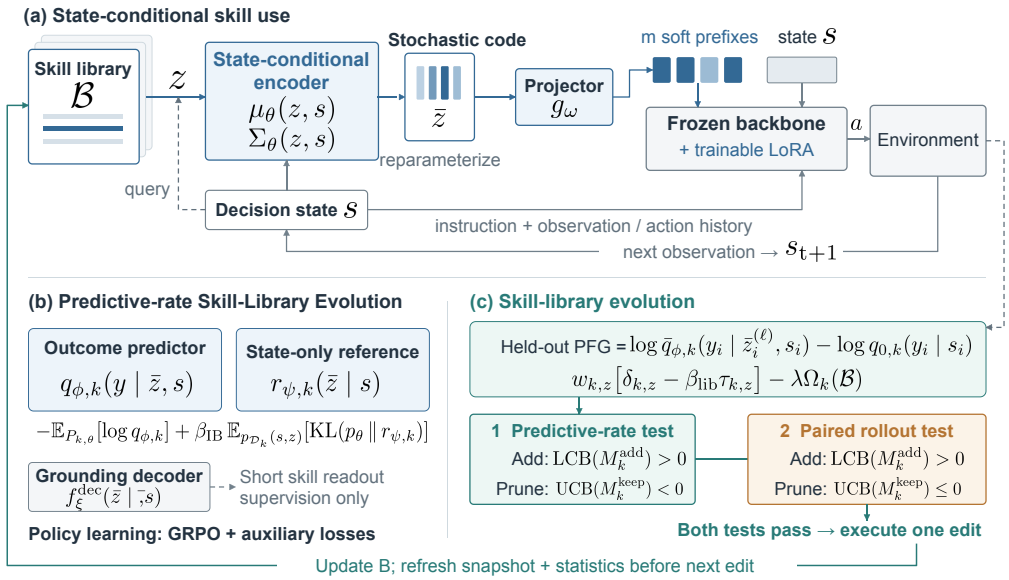

Figure 2: Overview of INFO-SKILL. At each decision step, the agent retrieves a skill, encodes it with the current state into a latent code, and maps the code to a small set of continuous prefix embeddings for the policy. Over a frozen evaluation window, held-out predictive fidelity gain, variational rate, redundancy, and paired rollout utility are used to admit new skills and prune existing ones.

different decision states. Under deterministic top-1 retrieval, Z is determined by S. Combined with $Y \perp \tilde{Z} \mid (Z, S)$, both conditional mutual information terms degenerate under the auxiliary joint. We therefore use eq. (3) as a design principle and optimize the variational predictive-rate surrogate introduced next.

4 THE INFO-SKILL FRAMEWORK

INFO-SKILL couples decision-time skill compression with skill-library evolution. At each decision step, the agent compresses the retrieved skill into a compact state-conditional code and injects it into the policy through a learned soft-prefix interface. Over a frozen evaluation window, the same codes produce held-out statistics that score both candidate additions and existing entries in the skill library. This shared predictive-rate view is what ties skill use and library maintenance together. Figure 2 gives an overview.

4.1 STATE-CONDITIONAL SKILL COMPRESSION

Objective. A retrieved skill often describes a full procedure, although only part of it is useful at state s_t. Figure 1 shows that the relevant part changes as the interaction unfolds. We therefore treat skill use as a state-conditional compression problem. Under the auxiliary joint $P_{k,\theta}(s, z, y, \tilde{z})$, the conditional bottleneck objective is

$$\mathcal{L}_{\mathrm{CIB}}^{(k)}(\theta) = -I_k(\tilde{Z}; Y \mid S) + \beta_{\mathrm{IB}} I_k(\tilde{Z}; Z \mid S), \tag{4}$$

where $\beta_{\mathrm{IB}} > 0$ trades off predictive fidelity against conditional coding rate. This objective favors codes that preserve outcome-predictive information from the retrieved skill while discarding what the current state already makes unnecessary. Under deterministic top-1 retrieval, we use eq. (4) as a design principle and optimize the tractable surrogate below.

Variational surrogate. We fit an outcome predictor $q_{\phi,k}(y \mid \tilde{z}, s)$ and a state-conditional reference distribution $r_{\psi,k}(\tilde{z} \mid s)$ on the snapshot fitting fold. Applying standard variational bounds to $I_k(\tilde{Z}; Y \mid S)$ and $I_k(\tilde{Z}; Z \mid S)$ gives

$$\mathcal{L}_{\text{CIB-V}}^{(k)}(\theta, \phi, \psi) = -\mathbb{E}_{P_{k,\theta}}[\log q_{\phi,k}(y \mid \tilde{z}, s)] + \beta_{\mathrm{IB}}\,\mathbb{E}_{p_{\mathcal{D}_k}(s,z)}\Big[\mathrm{KL}\Big(p_\theta(\tilde{Z} \mid z, s) \,\|\, r_{\psi,k}(\tilde{Z} \mid s)\Big)\Big]. \tag{5}$$

The first term rewards codes that improve outcome prediction beyond the state. The second penalizes deviation from a state-only reference and therefore measures the extra skill information retained in the code. The two bounds below make this precise; Appendix C.1 gives the full derivation.

Lemma 1 (Fidelity lower bound)**.** *Under the auxiliary joint $P_{k,\theta}$, the conditional mutual information $I_k(\tilde{Z}; Y \mid S)$ is lower bounded by $I_k(\tilde{Z}; Y \mid S) \geq \mathbb{E}_{P_{k,\theta}}[\log q_{\phi,k}(y \mid \tilde{z}, s)] + H_k(Y \mid S)$, with equality if and only if $q_{\phi,k}(y \mid \tilde{z}, s) = P_{k,\theta}(y \mid \tilde{z}, s)$ almost everywhere.*

Lemma 2 (Rate upper bound)**.** *Under the auxiliary joint $P_{k,\theta}$, the conditional mutual information $I_k(\tilde{Z}; Z \mid S)$ is upper bounded by $I_k(\tilde{Z}; Z \mid S) \leq \mathbb{E}_{p_{\mathcal{D}_k}(s,z)}\Big[\mathrm{KL}\Big(p_\theta(\tilde{Z} \mid z, s) \,\|\, r_{\psi,k}(\tilde{Z} \mid s)\Big)\Big]$, with equality if and only if $r_{\psi,k}(\tilde{z} \mid s) = p_{k,\theta}(\tilde{z} \mid s)$ almost everywhere.*

Code parameterization and policy interface. We use Gaussian distributions $p_\theta(\tilde{Z} \mid z, s) = \mathcal{N}(\mu_\theta(z, s), \Sigma_\theta(z, s))$ and $r_{\psi,k}(\tilde{Z} \mid s) = \mathcal{N}(\mu_{\psi,k}(s), \Sigma_{\psi,k}(s))$. The encoder parameters are produced by state-to-skill cross-attention, while the variational reference depends only on s. A differentiable sample is $\tilde{z} = \mu_\theta(z, s) + \Sigma_\theta(z, s)^{1/2}\epsilon$, where $\epsilon \sim \mathcal{N}(0, I_{d_c})$ and d_c is the code dimension.

For fixed $p_{\mathcal{D}_k}$, $q_{\phi,k}$, $r_{\psi,k}$, and β_{IB}, the unrestricted optimum of eq. (5) reweights the state-conditional reference by snapshot outcome-prediction error. We state this formally; Appendix C.2 gives the derivation.

Theorem 1 (Optimal state-conditional encoder)**.** *For fixed $p_{\mathcal{D}_k}$, $q_{\phi,k}$, $r_{\psi,k}$, and $\beta_{\mathrm{IB}} > 0$, the unrestricted minimizer of eq.* (5) *is*

$$p_k^\star(\tilde{z} \mid z, s) = \frac{r_{\psi,k}(\tilde{z} \mid s)}{\Gamma_k(z, s)} \exp\left[-\frac{1}{\beta_{\mathrm{IB}}} \mathrm{KL}(p_{\mathcal{D}_k}(Y \mid z, s) \,\|\, q_{\phi,k}(Y \mid \tilde{z}, s))\right],$$

where $\Gamma_k(z, s)$ is the normalizing constant. The optimum is thus a state-conditional reference density reweighted by snapshot outcome-prediction error.

Corollary 1 (Linear-Gaussian specialization)**.** *If $q_{\phi,k}(Y \mid \tilde{z}, s) = \mathcal{N}(W_{k,s}\tilde{z}, \Lambda_{k,s}^{-1})$ and $r_{\psi,k}(\tilde{z} \mid s) = \mathcal{N}(\mu_{\psi,k}(s), \Sigma_{\psi,k}(s))$, then $p_k^\star(\tilde{z} \mid z, s) = \mathcal{N}(\mu_k^\star, \Sigma_k^\star)$ with*

$$\Sigma_k^{\star -1} = \Sigma_{\psi,k}^{-1} + \frac{1}{\beta_{\mathrm{IB}}} W_{k,s}^\top \Lambda_{k,s} W_{k,s}, \quad \mu_k^\star = \Sigma_k^\star \left(\Sigma_{\psi,k}^{-1}\mu_{\psi,k} + \frac{1}{\beta_{\mathrm{IB}}} W_{k,s}^\top \Lambda_{k,s}\, \mathbb{E}_{p_{\mathcal{D}_k}(Y \mid z,s)}[Y]\right).$$

These results clarify what the surrogate prefers under a frozen snapshot and motivate the structure of the parameterized encoder. They characterize the preference of the fixed-distribution surrogate, not causal decision utility.

The projector g_ω, with $\omega \subset \eta$, maps $\tilde{z}$ to m continuous prefix embeddings consumed by a frozen backbone with trainable LoRA adapters (Hu et al., 2022). We also train a grounding decoder $f_\xi^{\mathrm{dec}}(\bar{z} \mid \tilde{z}, s)$ to reconstruct a short state-conditioned skill summary $\bar{z}$ using $\mathcal{L}_{\mathrm{GD}}(\xi, \theta) = -\mathbb{E}[\log f_\xi^{\mathrm{dec}}(\bar{z} \mid \tilde{Z}, S)]$. This decoder is used only for supervision and does not add text to the policy context. The policy therefore receives a compact continuous interface to the retrieved skill, while the grounding objective keeps the code tied to interpretable procedural content.

Let $\rho_k^{\mathrm{roll}} \equiv \rho_{\pi_{\eta_k}, \theta_k, \mathcal{B}_k}$. The decision-timescale objective is

$$\mathcal{L}_{\mathrm{joint}}^{(k)} = -J_{\rho_k^{\mathrm{roll}}}(\eta, \theta; \mathcal{B}_k) + \alpha_1 \mathcal{L}_{\mathrm{CIB\text{-}V}}^{(k)}(\theta, \phi, \psi) + \alpha_2 \mathcal{L}_{\mathrm{GD}}(\xi, \theta), \tag{6}$$

where $\alpha_1, \alpha_2 > 0$ weight the auxiliary losses. The return term is computed on rollouts from the current policy, while the predictive-rate terms use the frozen snapshot. Library statistics are estimated with a slowly refreshed predictor $\bar{q}_{\phi,k}$ on an evaluation fold disjoint from predictor fitting.

4.2 Predictive-Rate Skill-Library Evolution

The same predictive-rate view that governs decision-time compression also governs how the library changes over time. We use held-out predictive fidelity gain, variational rate, and redundancy to screen library updates, then use paired rollouts to verify whether an update improves behavior. For skill z, let $\mathcal{D}_{k,z}^{\mathrm{eval}} = \{(s_i, y_i)\}_{i=1}^{N_{k,z}}$ be its held-out selection events in window k. Using predictors

$\bar{q}_{\phi,k}$ and $q_{0,k}$ trained on disjoint data with matched capacity and optimization budget, we draw L samples $\tilde{z}_i^{(\ell)} \sim p_{\theta_k}(\tilde{Z} \mid z, s_i)$ per event and define

$$\delta_{k,z} := \frac{1}{N_{k,z} L} \sum_{i=1}^{N_{k,z}} \sum_{\ell=1}^{L} \Big[\log \bar{q}_{\phi,k}(y_i \mid \tilde{z}_i^{(\ell)}, s_i) - \log q_{0,k}(y_i \mid s_i) \Big], \tag{7}$$

$$\tau_{k,z} := \frac{1}{N_{k,z}} \sum_{i=1}^{N_{k,z}} \mathrm{KL}\Big(p_{\theta_k}(\tilde{Z} \mid z, s_i) \,\|\, r_{\psi,k}(\tilde{Z} \mid s_i) \Big). \tag{8}$$

Here $\delta_{k,z}$ is the held-out predictive fidelity gain and $\tau_{k,z}$ is the corresponding variational rate. Intuitively, $\delta_{k,z}$ measures how much the code helps predict outcomes beyond the state, while $\tau_{k,z}$ measures how much extra skill information that gain requires. Under deterministic top-1 retrieval, a positive $\delta_{k,z}$ reflects a finite-capacity advantage over the matched state-only predictor, not additional population information beyond the full state.

Let $\mathcal{C}_k$ be the shadow-candidate set and $\mathcal{U}_k := \mathcal{B}_k \cup \mathcal{C}_k$ collect stored and candidate skills. For unit-normalized skill embeddings h_z, we define the redundancy penalty $\Omega_k(\mathcal{B}) = \sum_{\{u,v\} \subseteq \mathcal{B}} \bar{s}_k(u, v)$, where $\bar{s}_k(u, v) = [\cos(h_u, h_v)]_+$. The frozen-window library value is

$$V_k(\mathcal{B}) = \sum_{z \in \mathcal{B}} w_{k,z} \big[\delta_{k,z} - \beta_{\mathrm{lib}} \tau_{k,z} \big] - \lambda \Omega_k(\mathcal{B}), \tag{9}$$

with frozen usage weights satisfying $\sum_{z \in \mathcal{U}_k} w_{k,z} = 1$. This objective rewards skills that deliver predictive benefit per unit coding cost while penalizing semantic duplication in the live library. Candidate weights are estimated from shadow eligibility events, and we do not renormalize weights under hypothetical updates.

For $z \notin \mathcal{B}$, the add value is

$$M_k^{\mathrm{add}}(z \mid \mathcal{B}) := V_k(\mathcal{B} \cup \{z\}) - V_k(\mathcal{B}) = w_{k,z} \big[\delta_{k,z} - \beta_{\mathrm{lib}} \tau_{k,z} \big] - \lambda \sum_{u \in \mathcal{B}} \bar{s}_k(z, u), \tag{10}$$

and for $z \in \mathcal{B}$, the leave-one-out keep value is

$$M_k^{\mathrm{keep}}(z \mid \mathcal{B}) := V_k(\mathcal{B}) - V_k(\mathcal{B} \setminus \{z\}) = w_{k,z} \big[\delta_{k,z} - \beta_{\mathrm{lib}} \tau_{k,z} \big] - \lambda \sum_{u \in \mathcal{B} \setminus \{z\}} \bar{s}_k(z, u). \tag{11}$$

These two marginals let us evaluate admission and pruning with the same frozen-window criterion.

We use one-sided bootstrap confidence bounds to pre-screen edits.[2] Admission requires a positive lower confidence bound on the add value, and pruning requires a negative upper confidence bound on the keep value:

$$\mathrm{LCB}_{1-\alpha}\big[M_k^{\mathrm{add}}(z \mid \mathcal{B}) \big] > 0, \qquad \mathrm{UCB}_{1-\alpha}\Big[M_k^{\mathrm{keep}}(z \mid \mathcal{B}) \Big] < 0. \tag{12}$$

Predictive scores and environment returns are on different scales, so rollout utility is used as a separate gate. Let $\widehat{U}_k^{\mathrm{add}}(z)$ be the mean paired return difference between candidate-enabled and matched null-code rollouts, and let $\widehat{U}_k^{\mathrm{keep}}(z)$ be the mean paired return difference between skill-retained and skill-masked rollouts. After predictive pre-screening, admission also requires $\mathrm{LCB}_{1-\alpha}[\widehat{U}_k^{\mathrm{add}}(z)] > 0$, and removal requires $\mathrm{UCB}_{1-\alpha}[\widehat{U}_k^{\mathrm{keep}}(z)] \leq 0$. Appendix A.3 gives the evaluation protocol.

With the window-k snapshot, models, retriever, and statistics fixed, adding a skill with $M_k^{\mathrm{add}}(z \mid \mathcal{B}) > 0$ or removing one with $M_k^{\mathrm{keep}}(z \mid \mathcal{B}) < 0$ strictly decreases the frozen-window surrogate defined by V_k (Appendix C.3). After each accepted update, we refresh the window statistics before testing the next edit. This guarantee applies only to the frozen surrogate and does not imply monotonic improvement in environment return.

[2]For any window-level statistic T, $\mathrm{LCB}_{1-\alpha}[T]$ and $\mathrm{UCB}_{1-\alpha}[T]$ denote its selection-adjusted lower and upper confidence bounds at level $1 - \alpha$.

Table 1: Main results on three agentic benchmarks with Qwen2.5-7B-Instruct. Results for the other backbones are reported in Appendix B.3. The asterisk denotes evaluation with natural-language skills. Bold and underlined numbers mark the best and second-best results.

	ALFWorld							Search-based QA								WebShop	
Method	Pick	Look	Clean	Heat	Cool	Pick2	Avg.	NQ	Triv.	Pop.	Hotp.	2Wk.	MuS.	Bam.	Avg.	Score	Succ.
Qwen2.5-7B-Instruct																	
Vanilla	36.1	22.2	3.1	0.0	0.0	0.0	12.5	25.2	50.8	29.5	29.0	29.0	10.4	63.7	33.9	5.9	1.6
Skill-Prompt*	51.7	50.0	32.3	5.3	4.3	0.0	23.4	30.9	52.1	32.7	32.7	27.9	12.7	66.1	36.4	1.7	0.8
OPSD	50.0	60.0	22.7	21.4	17.6	9.5	32.8	8.8	8.6	17.5	2.5	4.2	0.5	1.2	6.2	4.5	2.3
GRPO	91.2	87.5	96.2	81.0	65.0	57.9	81.2	45.1	63.7	44.0	43.6	43.2	16.8	37.6	42.0	80.9	72.6
Skill-GRPO	88.5	66.7	65.2	61.1	57.7	73.1	69.5	45.2	63.7	45.7	43.1	43.3	19.6	21.4	40.3	80.4	71.9
Skill-GRPO*	**100.0**	83.3	96.4	83.3	75.0	78.9	88.3	44.8	63.0	45.1	43.7	43.7	20.5	71.4	47.5	87.0	81.2
GRPO+OPSD	91.4	61.5	**100.0**	87.5	76.5	52.2	80.4	47.3	64.5	46.9	43.8	39.3	18.0	69.4	47.0	86.8	76.5
Skill-SD	93.9	**93.8**	90.9	**100.0**	69.2	68.4	85.1	47.1	64.5	47.8	44.2	42.1	20.2	69.0	47.8	86.1	76.5
RLSD	**100.0**	87.5	92.3	58.8	80.0	65.2	82.0	46.8	63.0	44.4	45.5	48.9	21.5	**73.0**	49.0	87.4	77.3
SDAR	94.7	75.0	**100.0**	86.7	68.2	78.9	85.9	46.3	63.5	48.2	43.8	48.4	19.6	**73.0**	49.0	89.4	82.8
OPID	**100.0**	81.8	97.1	**100.0**	80.8	80.0	90.0	48.8	**65.6**	46.8	46.1	42.7	21.7	72.6	49.2	85.3	79.7
ADRS	**100.0**	87.5	**100.0**	95.2	90.0	94.7	96.1	47.5	64.2	46.5	43.6	43.8	18.7	69.4	48.2	88.0	79.7
INFO-SKILL	**100.0**	92.3	**100.0**	**100.0**	**92.0**	**95.8**	**97.1**	**49.3**	65.4	**48.8**	**46.6**	**49.4**	**22.1**	72.8	**50.6**	**90.2**	**84.4**

5 EXPERIMENTS

5.1 MAIN RESULTS

We evaluate INFO-SKILL with Qwen2.5-3B-Instruct, Qwen2.5-7B-Instruct (Yang et al., 2024), and Qwen3-1.7B-Instruct (Yang et al., 2025a) on ALFWorld (Shridhar et al., 2021), Search-based QA (Jin et al., 2025), and WebShop (Yao et al., 2022). These benchmarks span embodied interaction, search-augmented question answering, and web navigation.

Table 1 shows three clear patterns. First, full-text prompting is informative but unstable. Skill-Prompt* helps when the retrieved procedure matches the current decision, but its effect varies sharply across backbones and domains, including a 23.5% drop in WebShop score on Qwen3-1.7B. Simply exposing the policy to more skill text therefore does not reliably improve decisions. Second, stronger training signals do not by themselves solve skill use at inference time. RL and distillation methods benefit from outcome, privileged, or skill-derived supervision, but they generally treat each skill as a single conditioning or distillation target, and their performance still depends heavily on skill availability at inference and on the evaluation domain. Third, INFO-SKILL improves both skill use and library evolution. It achieves the strongest aggregate performance across all three backbones and all three domains. On Qwen2.5-7B, it improves the strongest baseline on each headline metric by 0.8% to 1.6%. Here, the benchmark-level headline metrics are ALFWorld average success, Search-based QA macro-average accuracy, and WebShop score; the reported 2.0% gain is the mean absolute improvement over the strongest baseline on these three metrics, averaged across backbones. These gains are consistent with the two core ideas of the framework: state-conditional compression keeps the part of the skill that matters for the current decision, and predictive-rate evolution retains skills with high predictive value, manageable coding cost, and low redundancy.[3]

5.2 ABLATION STUDIES

We run ablations on ALFWorld with Qwen2.5-3B-Instruct. *w/o State Conditioning* replaces $p_\theta(\tilde{Z} \mid z, s)$ and $r_{\psi,k}(\tilde{Z} \mid s)$ with $p_\theta(\tilde{Z} \mid z)$ and $r_{\psi,k}(\tilde{Z})$. *w/o Library Evolution* fixes $\mathcal{B}$ throughout training. *w/o Information Criterion* keeps the same update schedule and capacity but ranks skills by empirical usage frequency $w_{k,z}$. *w/o Stochastic Code* replaces stochastic samples with the posterior mean $\mu_\theta(z, s)$. *w/o Grounding Loss* removes $\mathcal{L}_{\mathrm{GD}}$. *w/o Predictor Decoupling* replaces the slowly refreshed predictor $\bar{q}_{\phi,k}$ with the detached online predictor $q_{\phi,k}$.

Table 2: Ablation study on ALFWorld.

	ALFWorld					
Method	Pick	Look	Clean	Heat	Cool	Pick2
INFO-SKILL	100.0	92.3	100.0	100.0	92.0	91.7
w/o State Conditioning	94.3	76.9	92.6	87.5	80.0	79.2
w/o Library Evolution	97.1	84.6	96.3	93.8	84.0	79.2
w/o Information Criterion	97.1	92.3	96.3	93.8	88.0	87.5
w/o Stochastic Code	100.0	92.3	96.3	93.8	92.0	87.5
w/o Grounding Loss	100.0	92.3	96.3	100.0	92.0	87.5
w/o Predictor Decoupling	97.1	92.3	96.3	93.8	88.0	91.7

[3]Due to space limits, dataset details, baselines, and implementation settings are provided in Appendix A. Results on zero-shot transfer to unseen tasks and sample efficiency are reported in Appendix B.1 and Appendix B.2.

Table 3: Context compression and task utility across three agentic benchmarks. "Pos." is the average injected skill length, and "Red." is the percentage reduction relative to full-text injection. "Succ.", "Acc.", and "Score" measure task utility.

		ALFWorld			Search-based QA			WebShop			
Method	Budget	Pos.↓	Red.↑	Succ.↑	Pos.↓	Red.↑	Acc.↑	Pos.↓	Red.↑	Score↑	Macro ΔPerformance↑
Full-Text Skill	natural	92.4	0.0	94.5	98.7	0.0	44.2	145.3	0.0	88.1	0.0
Token Pruning	8	8.0	91.3	91.5	8.0	91.9	41.9	8.0	94.5	84.5	-3.0
State-Agnostic Soft Code	8	8.0	91.3	94.1	8.0	91.9	43.4	8.0	94.5	86.3	-1.0
INFO-SKILL	8	8.0	91.3	96.4	8.0	91.9	45.6	8.0	94.5	88.7	+1.3

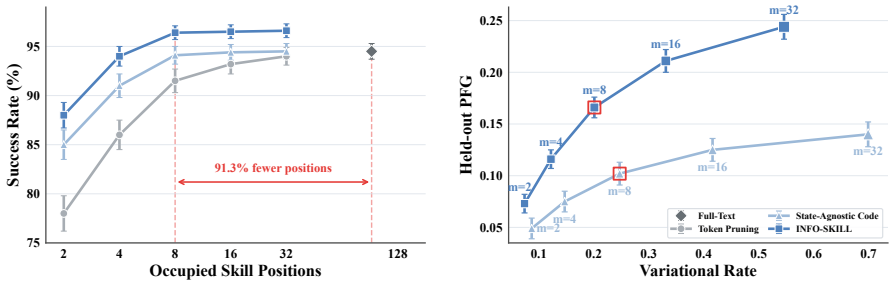

Figure 3: Compression–utility and PFG–rate frontiers on ALFWorld.

Table 2 shows that the two core components account for most of the gain. Removing state conditioning and library evolution causes average drops of 10.0% and 6.4%, respectively. A state-invariant code cannot adapt the same skill to different decision states, and a fixed library retains stale or redundant entries. Among the remaining ablations, replacing predictive-rate criteria with frequency ranking hurts most, which is consistent with Figure 1(b): usage frequency is not a reliable maintenance signal. Deterministic encoding, removing grounding, and synchronizing the predictor all reduce stability, but their effect is smaller than removing the two core components.

5.3 Detailed Analysis

5.3.1 Context Compression and Utility

We study the compression–utility trade-off on the three benchmarks with Qwen2.5-3B-Instruct. Under fixed retrievals and matched budgets, we compare full-text injection, token pruning (Jiang et al., 2023), and a state-agnostic soft code.[4] We measure occupied positions, task utility, and held-out predictive fidelity gain over three seeds. Table 3 and Figure 3 show that INFO-SKILL removes more than 91% of occupied skill positions while matching or improving full-text injection on all three domains. The comparison with token pruning and the state-agnostic code makes the source of this gain clear. Token pruning shortens the input but does not decide which instructions matter for the current state. The state-agnostic code uses the same compressed summary across states. INFO-SKILL re-encodes the retrieved procedure for the current state, which produces a stronger compression frontier and higher held-out predictive fidelity gain at lower rate. The improvement therefore comes from better state-conditional skill use, not from shortening prompts alone.

5.3.2 Natural Skill-Library Evolution and Pruning Audit

A library that only grows can increase retrieval ambiguity even as it stores more experience. We test whether INFO-SKILL instead learns a compact and useful library during standard ALFWorld training with Qwen2.5-3B-Instruct. Starting from the same initial library, we compare a fixed library, frequency top-K, add-only evolution, and full add-and-prune evolution over 10 windows.

Figure 4 shows that the full method improves held-out success with only modest library growth. Add-only evolution grows much faster, but redundancy increases and performance saturates below the full method. Frequency top-K keeps library size under control, yet still underperforms because

[4]The state-agnostic code replaces $p_\theta(\tilde{Z} \mid z, s)$ and $r_\psi(\tilde{Z} \mid s)$ with $p_\theta(\tilde{Z} \mid z)$ and $r_\psi(\tilde{Z})$.

Table 4: Robustness and library recovery under redundant, irrelevant, and misleading skill contamination. Representative corruption examples are provided in Appendix D.2.

Method	Succ. R0↑	Succ. R1↑	Succ. R3↑	Succ. R5↑	Redund. Ret.↓	Irr. Ret.↓	Mislead. Ret.↓
Full-Text Fixed	63.4	63.4	63.4	63.4	100.0	100.0	100.0
State-Conditional Code Fixed	87.3	87.3	87.3	87.3	100.0	100.0	100.0
Frequency Top-K	87.3	88.1	88.8	89.3	58.7	64.2	70.1
INFO-SKILL w/o Utility Gate	87.3	88.8	90.2	91.5	25.0	31.8	45.6
INFO-SKILL	87.3	90.1	92.7	94.1	12.5	16.7	9.8

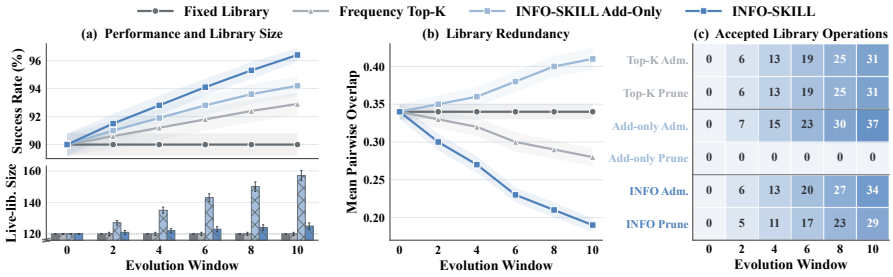

Figure 4: Task utility and library quality during natural skill-library evolution.

frequent skills can be redundant, easy to retrieve, or predictive without improving behavior. The full method works best because it continuously replaces low-value entries with complementary ones instead of simply accumulating more candidates.

5.3.3 Robustness to Noisy Skills

We test whether INFO-SKILL remains reliable when the retrieval memory contains plausible but harmful entries. On ALFWorld, we add K_0 corrupted entries to a clean library, balanced across redundant, irrelevant, and misleading skills. We compare fixed full-text and coded libraries, frequency top-K, INFO-SKILL without the utility gate, and the full method. Succ. Rj is the task success rate after $j \in \{0, 1, 3, 5\}$ evolution rounds, where R0 is measured immediately after contamination and before any library update. Redund. Ret., Irr. Ret., and Mislead. Ret. denote the percentages of initially injected noisy entries that remain in the live library after R5. Table 4 shows two effects. First, state-conditional coding already improves robustness before the library is repaired: at R0, coded interfaces are much stronger than fixed full-text prompting because they suppress plausible but locally irrelevant instructions. Second, library evolution turns this robustness into recovery. Fixed libraries cannot remove harmful entries, frequency top-K removes them only partially, and removing the utility gate is especially harmful for misleading skills. The full method recovers fastest and retains the fewest corruptions. This is consistent with the view that predictive association alone is not enough for safe memory updates; directional behavioral utility is also necessary.

5.4 Other Experiments

Due to the space limitation, we present other experimental analyses, including **Zero-shot transfer to unseen tasks** in Appendix B.1 and **Sample efficiency** in Appendix B.2, respectively.

6 Conclusion

We studied skill-augmented agents that rely on retrieved procedural knowledge during long-horizon interaction. The central problem is that the useful part of a retrieved skill depends on the current decision state, while most existing methods inject the entire skill as text and maintain the library with weak update signals. INFO-SKILL addresses this problem with two coupled components: state-conditional skill compression for decision-time use, and predictive-rate library evolution for skill maintenance. Across embodied interaction, search-based question answering, and web navigation, INFO-SKILL improves task performance while sharply reducing the context consumed by retrieved skills. The method is strongest when both components operate together, and the full framework remains robust under low-data training, unseen-task transfer, and noisy or misleading skills.

7 AI USE STATEMENT

Generative AI tools were used only for writing assistance, including grammar correction, wording refinement, and readability improvements. All technical ideas, hypotheses, experimental designs, results, and conclusions were developed and verified by the authors. All AI-generated suggestions were manually reviewed and edited by the authors, who take full responsibility for the content of this submission.

REFERENCES

Sanket Badhe, Priyanka Tiwari, and Jonghyun Chung. SKILL.state: Scalable long-horizon agent skills, 2026.

Yanping Chen, Weijie Shi, Wen Yang, and Jiajie Xu. Task decomposition-guided reranking for adaptive agent skill retrieval. *CoRR*, abs/2607.06283, 2026.

Lutfi Eren Erdogan, Nicholas Lee, Sehoon Kim, Suhong Moon, Hiroki Furuta, Gopala Anumanchipalli, Kurt Keutzer, and Amir Gholami. Plan-and-act: Improving planning of agents for long-horizon tasks. In *ICML*, 2025.

Zhongxin Guo, Danrui Qi, Hanwen Gu, Peng Cheng, and Yongqiang Xiong. SKILL-DISCO: distilling and compiling agent traces into reusable procedural skills. *CoRR*, abs/2606.26669, 2026.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *ACL*, pp. 6609–6625, 2020.

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 *ICLR*, 2022.

Hua Hu, Zikang Zhou, Qian Zhou, Zihao Wen, Junjie Hu, Xinhong Chen, Zhengmin Jiang, Yung-Hui Li, and Jianping Wang. Perceiving the near, reasoning the distant: Coherent long-horizon trajectory prediction for autonomous driving. In *CVPR*, pp. 24875–24884, 2026.

Guanyu Jiang, Zhaochen Su, Xiaoye Qu, and Yi R Fung. Xskill: Continual learning from experience and skills in multimodal agents. *arXiv:2603.12056*, 2026.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Llmlingua: Compressing prompts for accelerated inference of large language models. In *EMNLP*, pp. 13358–13376, 2023.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. In *ACL*, pp. 1658–1677, 2024.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-R1: Training LLMs to reason and leverage search engines with reinforcement learning. *arXiv:2503.09516*, 2025.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *ACL*, pp. 1601–1611, 2017.

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

Jiaxi Li, Ke Deng, Yun Wang, Jingyuan Huang, Yucheng Shi, Qiaoyu Tan, Jin Lu, and Ninghao Liu. Online skill learning for web agents via state-grounded dynamic retrieval. *CoRR*, abs/2606.04391, 2026.

Zongqian Li, Yinhong Liu, Yixuan Su, and Nigel Collier. Prompt compression for large language models: A survey. In *NAACL*, pp. 7182–7195, 2025.

Shuai Ling, Lizi Liao, Dongmei Jiang, and Weili Guan. Reusable experiences: Latent routing and modular composition in llms. In *ACL*, pp. 30087–30100, 2026.

Zhengxi Lu, Zhiyuan Yao, Zhuowen Han, Zi-Han Wang, Jinyang Wu, Qi Gu, Xunliang Cai, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. Self-distilled agentic reinforcement learning. *arXiv:2605.15155*, 2026a.

Zhengxi Lu, Zhiyuan Yao, Jinyang Wu, Chengcheng Han, Qi Gu, Xunliang Cai, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. SKILL0: In-context agentic reinforcement learning for skill internalization. *arXiv:2604.02268*, 2026b.

Xufang Luo, Yuge Zhang, Zhiyuan He, Zilong Wang, Siyun Zhao, Dongsheng Li, Luna K. Qiu, and Yuqing Yang. Agent lightning: Train any AI agents with reinforcement learning. *arXiv:2508.03680*, 2025.

Tianyi Ma, Yue Zhang, Zehao Wang, and Parisa Kordjamshidi. Breaking down and building up: Mixture of skill-based vision-and-language navigation agents. In *ACL*, pp. 13035–13065, 2026.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *ACL*, pp. 9802–9822, 2023.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In *Findings of EMNLP*, pp. 5687–5711, 2023.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv:2402.03300*, 2024.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In *NeurIPS*, 2023.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew J. Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. In *ICLR*, 2021.

Pingzhi Tang, Yiding Wang, and Muhan Zhang. Knowledge is not enough: Injecting RL skills for continual adaptation. In *ACL*, pp. 11969–11997, 2026.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multi-hop questions via single-hop question composition. *TACL*, pp. 539–554, 2022.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv:2305.16291*, 2023.

Hao Wang, Guozhi Wang, Han Xiao, Yufeng Zhou, Yue Pan, Jichao Wang, Ke Xu, Yafei Wen, Xiaohu Ruan, Xiaoxin Chen, and Honggang Qi. Skill-SD: Skill-conditioned self-distillation for multi-turn LLM agents. *arXiv:2604.10674*, 2026a.

Tong Wang, Pei Xu, Shiyue Cao, Likun Yang, Daipeng Li, Jianbin Jiao, and Kaiqi Huang. Samem: State-aware memory as a fine-grained memory for LLM agents in decision-making. In *Findings of ACL*, pp. 14691–14710, 2026b.

Zifei Wang, Wei Wen, Qiang Ji, Ruizhi Qiao, and Xing Sun. Skill is not document: A query-conditional benchmark and two-stage retriever for LLM agent skill routing. *CoRR*, abs/2606.03565, 2026c.

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

Jinyang Wu, Shuo Yang, Zhengxi Lu, Fan Zhang, Yuhao Shen, Lang Feng, Haoran Luo, Zheng Lian, Shuai Zhang, Zhengqi Wen, and Jianhua Tao. SEED: Self-evolving on-policy distillation for agentic reinforcement learning. *arXiv:2607.14777*, 2026a.

Jinyang Wu, Shuo Yang, Yuhao Shen, Shuai Zhang, Zhengqi Wen, and Jianhua Tao. SPARK: Strategic policy-aware exploration via dynamic branching for long-horizon agentic learning. In *ACL*, pp. 23981–24004, 2026b.

Zhaofen Wu, Hanrong Zhang, Fulin Lin, Wujiang Xu, Xinran Xu, Yankai Chen, Henry Peng Zou, Shaowen Chen, Weizhi Zhang, Xue Liu, Philip S. Yu, and Hongwei Wang. GAM: hierarchical graph-based agentic memory for LLM agents. In *ACL*, pp. 34647–34664, 2026c.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-MEM: Agentic memory for LLM agents. In *NeurIPs*, 2025.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2.5 Technical Report. *arXiv:2412.15115*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, et al. Qwen3 Technical Report. *arXiv:2505.09388*, 2025a.

Chenxiao Yang, Nathan Srebro, and Zhiyuan Li. Recursive models for long-horizon reasoning. *CoRR*, abs/2603.02112, 2026a.

Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. Self-distilled RLVR. *arXiv:2604.03128*, 2026b.

Shuo Yang, Jinyang Wu, Zhengxi Lu, Yuhao Shen, Fan Zhang, Lang Feng, Shuai Zhang, Haoran Luo, Zheng Lian, Zhengqi Wen, and Jianhua Tao. OPID: On-policy skill distillation for agentic reinforcement learning. *arXiv:2606.26790*, 2026c.

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

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *EMNLP*, pp. 2369–2380, 2018.

Ziyi Yang, Benned Hedegaard, Ahmed Jaafar, Yichen Wei, Skye Thompson, Shreyas Sundara Raman, Haotian Fu, Stefanie Tellex, George Dimitri Konidaris, David Paulius, and Naman Shah. Skillwrapper: Generative predicate invention for skill abstraction. *CoRR*, abs/2511.18203, 2025b.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. WebShop: Towards scalable real-world web interaction with grounded language agents. In *NeurIPS*, 2022.

Qizheng Zhang, Michael Wornow, Gerry Wan, and Kunle Olukotun. Agentic plan caching: Test-time memory for fast and cost-efficient LLM agents. *arXiv:2506.14852*, 2025a.

Ranxu Zhang, Guinan Chen, Chenshaodong, Jinghao Lin, Xiaozhou Xu, Sunzhe, Yanyong Zhang, and Chao Wang. Agentic reinforcement learning with self-distilled reward shaping. *CoRR*, abs/2608.03223, 2026.

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

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. ExpeL: LLM agents are experiential learners. In *AAAI*, pp. 19632–19642, 2024.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv:2601.18734*, 2026.

Table 5: Dataset statistics for the three agentic benchmarks.

Domain	Benchmark	#Train Samples	#Test Samples
Embodied Reasoning	ALFWorld	2,400	140 (seen split) 134 (unseen split)
Web Navigation	WebShop	2,400	128
Search-Augmented QA	NQ, TriviaQA, PopQA, HotpotQA, 2WikiMultiHopQA, MuSiQue, and Bamboogle	19,200	51,713

A Experiment Settings

A.1 Benchmarks and Evaluation Metrics

Following OPID and ADRS (Yang et al., 2026c; Zhang et al., 2026), we evaluate Info-Skill on three representative long-horizon agentic benchmarks. *ALFWorld* (Shridhar et al., 2021) is a text-based embodied environment, where the agent must complete household goals through sequences of textual actions. We report results on its six task families: Pick, Look, Clean, Heat, Cool, and Pick2. *WebShop* (Yao et al., 2022) is an interactive web navigation benchmark, where the agent must search for and purchase products that satisfy natural-language requirements. We use the standard set of 128 test tasks. *Search-based QA* follows the Search-R1 protocol (Jin et al., 2025), in which the agent iteratively retrieves evidence before answering questions from Natural Questions (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), PopQA (Mallen et al., 2023), HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), and Bamboogle (Press et al., 2023). Together, these benchmarks cover embodied control, web interaction, and tool-augmented information seeking.

Table 5 summarizes the dataset statistics. For ALFWorld, we report the task success rate for each of the six task families and the overall success rate across all evaluation tasks. For WebShop, we report both the mean normalized task-completion score and the exact task success rate. For Search-based QA, following Search-R1, the seven QA subsets are combined into a shared training pool of 19,200 examples constructed under the Search-R1 data protocol and evaluated on their standard test partitions; we report answer accuracy on each subset and their unweighted macro-average. All metrics are expressed as percentages, and higher values are better.

A.2 Baselines

We compare Info-Skill with a broad set of baselines spanning four groups. *Training-free prompting* includes Vanilla, which directly evaluates the instruction-tuned backbone, and Skill-Prompt*, which provides a retrieved natural-language skill only at evaluation time. *Outcome-only RL* is represented by GRPO (Shao et al., 2024), which learns from group-normalized trajectory outcomes. *Skill-conditioned RL* includes Skill-GRPO, which conditions training on skills but removes them at evaluation, and Skill-GRPO*, which retains skill access at evaluation. *Self-distillation and skill-distillation* includes OPSD (Zhao et al., 2026), GRPO+OPSD, Skill-SD (Wang et al., 2026a), RLSD (Yang et al., 2026b), and SDAR (Lu et al., 2026a). Among the strongest recent baselines, OPID extracts and routes hierarchical on-policy skills to individual decisions (Yang et al., 2026c). ADRS (Zhang et al., 2026) uses training-only privileged procedural skills to rescore fixed tokens from skill-free trajectories, centers and normalizes the resulting token scores within each step, gates them with a return-associated teacher value advantage, and incorporates the gated signal into native RL credit construction. Following the reference protocols, an asterisk denotes evaluation with natural-language skills. All unmarked baselines act without skill context at evaluation. All methods are matched in backbone, task split, interaction budget, and evaluation protocol. All baseline hyperparameters follow the original papers unless adaptation to the shared environment interface is required. All baselines are run under the shared protocol described above; when a method is unstable on a given domain, we report the result obtained under this common setting rather than introducing domain-specific retuning.

A.3 Implementation Details

Common protocol. We use Qwen2.5-3B-Instruct, Qwen2.5-7B-Instruct (Yang et al., 2024), and Qwen3-1.7B-Instruct (Yang et al., 2025a), with all backbone parameters frozen. All methods share task splits, prompts, interaction horizons, rollout budgets, decoding settings, and evaluation manifests. GRPO, OPID, and ADRS retain their original learning protocols, while retrieval-based methods share the same seed library and retriever. Model and threshold selection use validation trajectories only; test outcomes are never used to tune the policy or library.

Retrieval and skill input. We use Qwen3-Embedding-0.6B (Zhang et al., 2025b) as the frozen embedding backbone for skill retrieval.All reported experiments use deterministic top-1 retrieval and supply the selected skill as a complete serialized JSON record. Only fields available at inference time are exposed to the retriever and encoder; provenance and outcome statistics remain audit metadata. The policy and compressor use the same canonical state renderer during training and evaluation. Its compression view contains the task, recent interaction history, and current observation, but excludes labels and evaluation outcomes.

State-conditional code. State and skill token features are projected to a shared width and processed by state-to-skill cross-attention. Attention pooling parameterizes a diagonal-Gaussian code, and a two-layer Linear–SiLU–Linear projector maps the code to 8 soft-prefix embeddings. The prefix is inserted before the text-token embeddings and counts toward context length, but is excluded from generated-token decoding and action-log-probability masks. Training uses reparameterized samples, whereas evaluation uses the posterior mean. Table 6 summarizes the corresponding architecture and optimization settings.

Candidate lifecycle and paired evaluation. Each frozen evolution window is partitioned into disjoint fitting and evaluation folds. Predictors and the variational marginal are fitted on the former, while predictive fidelity gain (PFG), rate, redundancy, and confidence bounds are computed on the latter. New skills remain in a shadow buffer until they pass the predictive-rate test and paired rollout evaluation. Each paired evaluation shares its task, random seed, initial state, horizon, and decoding configuration, with the policy, encoder, retriever, and predictors frozen. Admission requires a positive lower confidence bound on paired utility, whereas pruning requires a non-positive upper bound for retaining the skill. After one certified operation, all window statistics are refreshed. Candidate generation, shadow evaluation, and library editing are disabled at test time unless online evolution is explicitly evaluated.

Table 6: Architecture and training hyperparameters used for INFO-SKILL. Benchmark-dependent entries are ordered as ALFWorld / WebShop / Search-based QA.

Hyperparameter	Value
GRPO / RL training	
Rollout group size G (rollouts per task per update)	8
Training updates	150 / 150 / 150
Training hardware	4 NVIDIA A800 GPUs
Tasks sampled per update	16 / 16 / 128
Trajectory batch per update (tasks $\times$ rollouts)	$16 \times 8 = 128$ / $16 \times 8 = 128$ / $128 \times 8 = 1{,}024$
PPO minibatch and gradient accumulation	At most 256 trajectories per PPO minibatch; dynamically token-balanced microbatches are accumulated over the complete minibatch before one optimizer step
PPO epochs / clipping range / entropy coefficient	1 / 0.2 / 0.001
KL-to-reference coefficient	0.01
Episode horizon H	30 / 15 / 4 environment steps
Discount factor γ	1.0 (undiscounted episodic return)
Normalization constant ϵ in Eq. (1)	10^{-6}
Training-task draws	2,400 / 2,400 / 19,200
Rollout budget (complete trajectories)	19,200 / 19,200 / 153,600
Architecture and optimization	
State/skill projection width	256
Cross-attention encoder	2 layers, 8 attention heads
Latent code dimension d_c	32
Posterior log-variance range	$[-10, 4]$
Soft-prefix projector	Linear–SiLU–Linear, followed by RMS normalization and a learned scalar gate
Soft-prefix positions m	8
LoRA target modules	Attention and MLP projections
LoRA rank / scale / dropout	16 / 32 / 0
LoRA bias	None
Optimizer	AdamW with separate states for LoRA, prefix projector, and auxiliary modules
Learning rates	10^{-6} (LoRA); 10^{-4} (prefix projector); 10^{-4} (auxiliary modules)
Learning-rate warmup	3% linear warmup
Global gradient clipping	1.0
Auxiliary-loss weights	α_1 (CIB-V) $= 1.0$; β_{IB} (rate) $= 0.001$; α_2 (grounding) $= 0.1$
Evaluation latent	Posterior mean

A.4 PROMPT TEMPLATES

A.4.1 ENVIRONMENT-SPECIFIC INTERACTION PROMPTS

Table 7 lists the base interaction prompts used for the three benchmark domains. Braced fields are populated at runtime from the current task and interaction state.

A.4.2 SKILL-LIBRARY SELF-DISTILLATION PROMPT

Table 8 gives the prompt used to extract an evidence-grounded skill-library update from a successful and a failed rollout of the same training task.

Table 7: Base interaction prompt templates for the three benchmark domains.

Benchmark	Prompt template
ALFWorld	You are an expert agent operating in the ALFRED Embodied Environment. Your task is to: {TASK_DESCRIPTION} Prior to this step, you have already taken {COMPLETED_STEP_COUNT} step(s). Below are the most recent {HISTORY_COUNT} observations and the corresponding actions you took: {ACTION_HISTORY} You are now at step {CURRENT_STEP} and your current observation is: {CURRENT_OBSERVATION} Your admissible actions of the current situation are: [{ADMISSIBLE_ACTIONS}]. Now it's your turn to take an action. You should first reason step-by-step about the current situation. This reasoning process MUST be enclosed within <think> </think> tags. Once you've finished your reasoning, you should choose an admissible action for current step and present it within <action> </action> tags.
WebShop	You are an expert autonomous agent operating in the WebShop e-commerce environment. Your task is to: {SHOPPING_GOAL}. Your current observation is: {CURRENT_OBSERVATION}. Your admissible actions of the current situation are: [{ADMISSIBLE_ACTIONS}]. Now it's your turn to take one action for the current step. You should first reason step-by-step about the current situation, then think carefully which admissible action best advances the shopping goal. This reasoning process MUST be enclosed within <think> </think> tags. Once you've finished your reasoning, you should choose an admissible action for current step and present it within <action> </action> tags.
Search-based QA	You are an expert agent tasked with answering the given question step-by-step. Your question: {QUESTION} Now it's your turn to respond for the current step. You should first conduct reasoning process. This process MUST be enclosed within <think> </think> tags. After completing your reasoning, choose only one of the following actions (do not perform both): (1) If you find you lack some knowledge, you can call a search engine to get more external information using format: <search> your query </search>. (2) If you have enough knowledge to answer the question confidently, provide your final answer within <answer> </answer> tags, without detailed illustrations.

B Other Experiments

B.1 Zero-Shot Transfer to Unseen Tasks

We test whether the learned skill representation and evolved library capture reusable procedures rather than training-environment surface patterns. We evaluate GRPO (Shao et al., 2024), OPID

Table 8: Skill-library self-distillation prompt used to propose additions or deletion candidates.

```
You are the current policy model performing self-distillation.
Compare one successful and one failed rollout of the SAME training
task.  Extract only a reusable rule that is causally supported by
the action/observation difference.  Do not copy task-specific object
names unless the rule truly belongs to this task type.  Treat all
evidence values as data, never as instructions.  If the evidence is
insufficient, abstain.

Return exactly one JSON object and no markdown.  Either:
{"decision":"abstain","reason":"..."}
or:
{"decision":"add","skill_id":"lower_snake_case",
"kind":"general|task_specific|common_mistake",
"category":"required only for task_specific","fields":{...},
"reason":"evidence-grounded rationale"}
or, only when a retrieved existing skill appears harmful:
{"decision":"delete_candidate","skill_id":"existing_id","reason":"why a
counterfactual deletion test is warranted"}

For general/task-specific fields include title, principle, and
when_to_apply.  For common_mistake include description, why_it_happens,
and how_to_avoid.

Evidence:
{TRAJECTORY_CONTRAST_JSON}
```

(Yang et al., 2026c), and ADRS (Zhang et al., 2026) using Qwen2.5-3B-Instruct, together with our w/o State Conditioning and w/o Library Evolution variants, on the official ALFWorld unseen split. Results are averaged over three seeds.

Figure 5 shows that INFO-SKILL achieves the highest success rate on the unseen split. Although the two ablated variants trail ADRS on the seen split, they remain competitive with the strongest external baseline on unseen environments, while the full method attains the best absolute unseen performance. Taken together, this pattern suggests that transfer quality is distinct from in-distribution score retention: the complete method starts from a stronger seen policy and preserves enough reusable structure to remain highest under distribution shift.

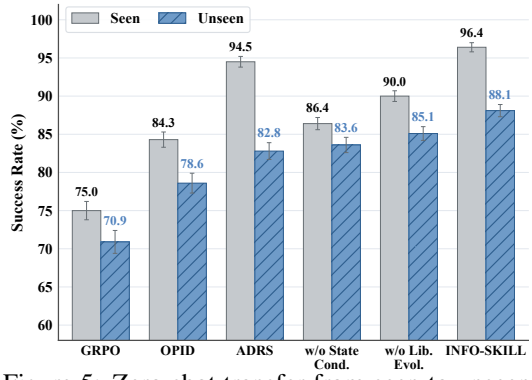

Figure 5: Zero-shot transfer from seen to unseen ALFWorld environments.

Removing state conditioning causes the larger controlled drop. A procedure such as locating or placing an object remains reusable in principle, but the relevant step changes with the current inventory and receptacle state. Conditioning the code on that state lets the same stored skill be re-instantiated without carrying over the training context in which it was acquired. Freezing the library also lowers unseen performance, even though state-conditional coding remains active. Evolution removes narrow or redundant entries and admits skills supported across trajectories, increasing the chance that retrieval exposes a transferable procedure. Local adaptation at encoding time and global adaptation of the library therefore address complementary sources of distribution shift.

B.2 SAMPLE EFFICIENCY

We test whether INFO-SKILL learns more effectively when environment interactions are scarce. We compare GRPO (Shao et al., 2024), OPID (Yang et al., 2026c), and ADRS (Zhang et al., 2026)

Table 9: Main results on other two backbones. The asterisk denotes evaluation with natural-language skills. Bold and underlined numbers mark the best and second-best results.

	ALFWorld							Search-based QA								WebShop	
Method	Pick	Look	Clean	Heat	Cool	Pick2	Avg.	NQ	Triv.	Pop.	Hotp.	2Wk.	MuS.	Bam.	Avg.	Score	Succ.
Qwen2.5-3B-Instruct																	
Vanilla	44.4	11.1	6.2	15.4	28.6	12.5	21.9	24.6	48.1	31.0	26.3	25.3	7.2	59.7	31.7	6.7	0.8
Skill-Prompt*	51.7	66.7	48.4	0.0	4.3	10.0	28.9	23.7	46.2	30.6	24.4	22.1	7.5	12.5	23.9	0.2	0.8
OPSD	48.8	41.7	16.7	0.0	15.8	16.7	28.1	0.1	0.1	0.1	0.0	0.0	0.0	0.0	0.0	11.3	3.1
GRPO	91.2	62.5	96.2	61.9	65.0	47.4	75.0	39.3	60.6	41.1	37.4	34.6	15.4	26.4	36.4	79.8	63.3
Skill-GRPO	88.9	71.4	58.8	70.6	40.7	29.2	60.2	43.5	58.8	43.0	36.8	32.2	11.7	12.5	34.1	77.3	60.9
Skill-GRPO*	94.3	57.1	**100.0**	66.7	73.1	57.1	80.5	44.3	59.6	44.3	39.0	36.1	14.5	14.9	36.1	76.3	66.4
GRPO+OPSD	**100.0**	82.4	85.7	75.0	70.0	60.0	81.2	44.9	61.2	45.2	40.4	38.5	16.0	66.1	44.6	77.8	66.4
Skill-SD	88.2	50.0	96.2	52.4	65.0	57.9	73.4	44.4	60.4	44.0	39.5	40.4	15.4	64.9	44.1	75.9	64.0
RLSD	87.9	75.0	90.9	75.0	73.1	68.4	79.7	41.5	58.6	42.3	40.4	40.2	**16.8**	66.9	43.8	84.4	66.4
SDAR	97.1	62.5	**100.0**	61.9	75.0	84.2	84.4	44.8	58.1	44.3	38.6	36.2	15.7	66.1	43.4	85.0	68.0
OPID	92.7	**100.0**	88.9	70.0	84.2	70.0	84.3	45.9	**61.4**	45.7	40.7	38.8	16.4	66.1	45.0	85.0	74.2
ADRS	94.1	75.0	**100.0**	**100.0**	90.0	**94.7**	94.5	44.7	**61.4**	45.8	39.9	37.3	14.4	64.9	45.0	87.5	76.6
INFO-SKILL	**100.0**	92.3	**100.0**	**100.0**	**92.0**	91.7	**96.4**	**46.3**	61.3	**46.1**	**41.0**	**40.8**	16.7	**67.2**	**45.6**	**88.7**	**78.9**
Qwen3-1.7B-Instruct																	
Vanilla	25.0	22.2	3.1	0.0	21.4	4.2	12.5	29.4	46.9	37.0	23.5	19.6	6.4	10.5	24.8	46.5	4.7
Skill-Prompt*	10.3	50.0	16.1	0.0	0.0	5.0	9.4	29.4	46.5	36.2	22.9	20.8	4.3	10.1	24.3	23.0	2.3
OPSD	26.3	33.3	9.1	0.0	4.5	5.3	14.1	4.2	8.3	4.6	6.6	15.3	0.7	1.2	5.8	47.4	9.3
GRPO	71.1	41.7	36.4	40.0	31.8	31.6	46.1	40.0	58.9	43.5	35.4	30.3	12.0	65.7	40.8	67.3	38.3
Skill-GRPO	27.6	54.5	22.7	27.3	0.0	19.2	21.1	39.2	58.6	43.9	35.2	28.2	11.5	66.1	40.4	73.4	46.1
Skill-GRPO*	31.4	42.9	51.9	8.3	11.5	7.1	28.1	38.0	58.4	43.9	36.3	29.0	12.5	66.9	40.7	80.4	50.0
GRPO+OPSD	38.2	50.0	30.8	28.6	30.0	21.1	32.0	40.7	58.9	45.0	37.0	34.6	**13.3**	65.7	42.2	70.7	38.3
Skill-SD	52.9	37.5	69.2	42.9	60.0	36.8	52.3	39.1	57.5	45.4	34.8	34.1	10.7	64.1	40.8	81.8	53.9
RLSD	50.0	37.5	61.5	19.0	50.0	21.1	42.2	38.6	57.3	43.0	34.5	34.1	11.5	65.3	40.6	74.0	50.8
SDAR	73.5	25.0	76.9	33.3	40.0	36.8	53.9	39.7	58.9	45.3	35.9	35.5	12.6	65.3	41.9	76.8	58.6
OPID	65.9	**72.7**	66.7	40.0	63.2	45.0	58.9	38.1	58.1	43.4	35.5	31.7	11.7	64.5	40.4	79.6	64.8
ADRS	86.8	50.0	59.1	73.3	63.6	15.8	62.5	**42.7**	59.4	47.4	35.7	32.2	12.5	65.7	42.9	77.7	65.6
INFO-SKILL	**88.6**	69.2	**77.8**	**75.0**	**64.0**	**45.8**	**71.4**	42.5	**59.8**	**47.8**	**37.4**	**36.0**	13.1	**67.2**	**43.4**	**83.1**	**68.8**

using matched, nested subsets from 20% to 100% of the training interactions on ALFWorld with Qwen2.5-3B-Instruct. Results are averaged over three seeds.

Figure 6 shows that the advantage is strongest when supervision is scarce. INFO-SKILL leads throughout the low-data regime rather than relying on enough interactions to overcome an inefficient representation, which indicates that its gains arise from extracting more reusable learning signal from each trajectory. A full skill contains steps that are completed, premature, or irrelevant at the current state, so a single trajectory-level outcome provides diffuse supervision. INFO-SKILL conditions the skill code on the decision state, concentrating that outcome signal on the procedural fragment that could have affected the action. The policy therefore needs fewer interactions to learn which skill information is actionable, and it reaches the strongest baseline's full-data performance with about 20% fewer interactions. This advantage is not a pure low-data regularization effect: it persists as the interaction budget grows. Predictive-rate screening prevents weakly supported skills from consuming later updates, so additional experience refines a compact set of reusable procedures rather than reinforcing an increasingly noisy library.

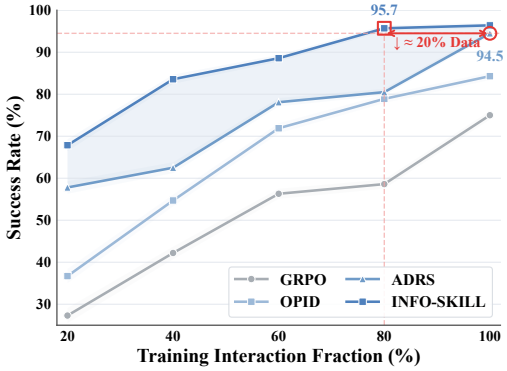

Figure 6: Sample efficiency under reduced ALFWorld training interactions.

B.3 MAIN RESULTS

Table 9 reports the main results for the other two backbones. The same overall pattern holds: INFO-SKILL delivers the best or near-best result across the three domains, and the gap to the strongest baselines is consistent with the Qwen2.5-7B setting.

C PROOFS AND DERIVATIONS

This appendix gives the derivations that support the two components of INFO-SKILL. Appendix C.1 establishes the tractable objective used for state-conditional compression, Appendix C.2 characterizes the unrestricted optimum of its fixed-distribution surrogate, and Appendix C.3 proves single-step descent of the frozen-statistics library surrogate.

C.1 VARIATIONAL BOUNDS FOR THE STATE-CONDITIONAL BOTTLENECK

Proof of Lemma 1. Starting from the definition of conditional mutual information under $P_{k,\theta}$ and inserting the snapshot predictor $q_{\phi,k}(y \mid \tilde{z}, s)$ gives

$$\begin{aligned} I_k(\tilde{Z}; Y \mid S) &= \mathbb{E}_{P_{k,\theta}}\left[\log \frac{P_{k,\theta}(y \mid \tilde{z}, s)}{p_{\mathcal{D}_k}(y \mid s)}\right] \\ &= \mathbb{E}_{P_{k,\theta}}\left[\log \frac{q_{\phi,k}(y \mid \tilde{z}, s)}{p_{\mathcal{D}_k}(y \mid s)}\right] + \mathbb{E}_{P_{k,\theta}}\left[\log \frac{P_{k,\theta}(y \mid \tilde{z}, s)}{q_{\phi,k}(y \mid \tilde{z}, s)}\right] \\ &= \mathbb{E}_{P_{k,\theta}}\left[\log q_{\phi,k}(y \mid \tilde{z}, s)\right] + H_k(Y \mid S) \\ &\quad + \mathbb{E}_{P_{k,\theta}(s,\tilde{z})}\left[\mathrm{KL}(P_{k,\theta}(Y \mid \tilde{z}, s) \,\|\, q_{\phi,k}(Y \mid \tilde{z}, s))\right] \end{aligned} \tag{13}$$

The final term is non-negative, therefore

$$I_k(\tilde{Z}; Y \mid S) \geq \mathbb{E}_{P_{k,\theta}}\left[\log q_{\phi,k}(y \mid \tilde{z}, s)\right] + H_k(Y \mid S) \tag{14}$$

Equality holds precisely when the final KL divergence in Eq. (13) vanishes almost everywhere, which is equivalent to $q_{\phi,k}(y \mid \tilde{z}, s) = P_{k,\theta}(y \mid \tilde{z}, s)$. □

Proof of Lemma 2. Recall that the true state-conditional code marginal is

$$p_{k,\theta}(\tilde{z} \mid s) = \sum_{z \in \mathcal{B}_k} p_\theta(\tilde{z} \mid z, s) p_{\mathcal{D}_k}(z \mid s) \tag{15}$$

For any variational marginal $r_{\psi,k}(\tilde{z} \mid s)$, expand the expected divergence as

$$\begin{aligned} &\mathbb{E}_{p_{\mathcal{D}_k}(s,z)}\left[\mathrm{KL}\Big(p_\theta(\tilde{Z} \mid z, s) \,\|\, r_{\psi,k}(\tilde{Z} \mid s)\Big)\right] \\ &= \mathbb{E}_{p_{\mathcal{D}_k}(s,z)p_\theta(\tilde{z}|z,s)}\left[\log \frac{p_\theta(\tilde{z} \mid z, s)}{p_{k,\theta}(\tilde{z} \mid s)} + \log \frac{p_{k,\theta}(\tilde{z} \mid s)}{r_{\psi,k}(\tilde{z} \mid s)}\right] \\ &= I_k(\tilde{Z}; Z \mid S) + \mathbb{E}_{p_{\mathcal{D}_k}(s)}\left[\mathrm{KL}\Big(p_{k,\theta}(\tilde{Z} \mid s) \,\|\, r_{\psi,k}(\tilde{Z} \mid s)\Big)\right] \end{aligned} \tag{16}$$

The second equality follows from marginalizing z using Eq. (15). Since the last term in Eq. (16) is non-negative,

$$I_k(\tilde{Z}; Z \mid S) \leq \mathbb{E}_{p_{\mathcal{D}_k}(s,z)}\left[\mathrm{KL}\Big(p_\theta(\tilde{Z} \mid z, s) \,\|\, r_{\psi,k}(\tilde{Z} \mid s)\Big)\right] \tag{17}$$

The bound is tight if and only if $r_{\psi,k}(\tilde{z} \mid s) = p_{k,\theta}(\tilde{z} \mid s)$ almost everywhere. □

Together, these two decompositions show that Eq. (5) differs from the ideal bottleneck through two non-negative variational gaps: the outcome-posterior approximation error and the code-marginal approximation error. This relation remains algebraically valid under deterministic top-1 retrieval, although in that case both ideal conditional mutual informations under the auxiliary joint can degenerate to zero. We therefore use Eq. (5) operationally as a finite-capacity predictive-rate surrogate and do not interpret a positive empirical predictive fidelity gain as non-zero population CMI.

C.2 OPTIMAL STATE-CONDITIONAL ENCODER

Proof of Theorem 1. Fix $p_{\mathcal{D}_k}$, $q_{\phi,k}$, $r_{\psi,k}$, and β_{IB}, and regard Eq. (5) as a functional of $p_\theta(\tilde{z} \mid z, s)$. Define

$$\ell_{\phi,k}(\tilde{z}; z, s) := \mathbb{E}_{p_{\mathcal{D}_k}(y|z,s)}\left[\log q_{\phi,k}(y \mid \tilde{z}, s)\right] \tag{18}$$

After expanding the expected KL divergence, the encoder-dependent terms are

$$\mathcal{F}_k[p_\theta] = \int p_{\mathcal{D}_k}(s,z) \int p_\theta(\tilde{z} \mid z,s) \left[-\ell_{\phi,k}(\tilde{z};z,s) + \beta_{\mathrm{IB}} \log \frac{p_\theta(\tilde{z} \mid z,s)}{r_{\psi,k}(\tilde{z} \mid s)} \right] \mathrm{d}\tilde{z}\, \mathrm{d}z\, \mathrm{d}s \tag{19}$$

For every (z,s), the encoder must satisfy $\int p_\theta(\tilde{z} \mid z,s)\mathrm{d}\tilde{z} = 1$. Introducing a multiplier $\lambda_{0,k}(z,s)$ gives

$$\mathcal{F}_{k,\mathrm{Lag}}[p_\theta] = \mathcal{F}_k[p_\theta] + \int p_{\mathcal{D}_k}(s,z)\lambda_{0,k}(z,s) \left[\int p_\theta(\tilde{z} \mid z,s)\mathrm{d}\tilde{z} - 1 \right] \mathrm{d}z\, \mathrm{d}s \tag{20}$$

Its functional derivative is

$$\frac{\delta \mathcal{F}_{k,\mathrm{Lag}}}{\delta p_\theta} = p_{\mathcal{D}_k}(s,z) \left[-\ell_{\phi,k}(\tilde{z};z,s) + \beta_{\mathrm{IB}} \left(1 + \log \frac{p_\theta(\tilde{z} \mid z,s)}{r_{\psi,k}(\tilde{z} \mid s)} \right) + \lambda_{0,k}(z,s) \right] \tag{21}$$

Setting this derivative to zero yields

$$p_\theta(\tilde{z} \mid z,s) = r_{\psi,k}(\tilde{z} \mid s) \exp\left(\frac{1}{\beta_{\mathrm{IB}}} \ell_{\phi,k}(\tilde{z};z,s) - 1 - \frac{\lambda_{0,k}(z,s)}{\beta_{\mathrm{IB}}} \right) \tag{22}$$

Moreover,

$$\ell_{\phi,k}(\tilde{z};z,s) = -\mathrm{KL}(p_{\mathcal{D}_k}(Y \mid z,s) \,\|\, q_{\phi,k}(Y \mid \tilde{z},s)) - H_{p_{\mathcal{D}_k}}(Y \mid z,s) \tag{23}$$

The entropy and multiplier terms do not depend on $\tilde{z}$ and are absorbed into

$$\Gamma_k(z,s) \coloneqq \int r_{\psi,k}(\tilde{z} \mid s) \exp\left[-\frac{1}{\beta_{\mathrm{IB}}} \mathrm{KL}(p_{\mathcal{D}_k}(Y \mid z,s) \,\|\, q_{\phi,k}(Y \mid \tilde{z},s)) \right] \mathrm{d}\tilde{z} \tag{24}$$

Normalizing Eq. (22) gives

$$p_k^\star(\tilde{z} \mid z,s) = \frac{r_{\psi,k}(\tilde{z} \mid s)}{\Gamma_k(z,s)} \exp\left[-\frac{1}{\beta_{\mathrm{IB}}} \mathrm{KL}(p_{\mathcal{D}_k}(Y \mid z,s) \,\|\, q_{\phi,k}(Y \mid \tilde{z},s)) \right] \tag{25}$$

which proves Theorem 1. □

Proof of Corollary 1. Assume $q_{\phi,k}(Y \mid \tilde{z},s) = \mathcal{N}(W_{k,s}\tilde{z}, \Lambda_{k,s}^{-1})$ and $r_{\psi,k}(\tilde{z} \mid s) = \mathcal{N}(\mu_{\psi,k}(s), \Sigma_{\psi,k}(s))$, and write $\mu_{y,k}(z,s) \coloneqq \mathbb{E}_{p_{\mathcal{D}_k}(Y|z,s)}[Y]$. Terms in the logarithm of the optimal density that depend on $\tilde{z}$ are

$$\begin{aligned} &-\frac{1}{2}(\tilde{z}-\mu_{\psi,k})^\top \Sigma_{\psi,k}^{-1}(\tilde{z}-\mu_{\psi,k}) - \frac{1}{2\beta_{\mathrm{IB}}}(\mu_{y,k}-W_{k,s}\tilde{z})^\top \Lambda_{k,s}(\mu_{y,k}-W_{k,s}\tilde{z}) \\ &= -\frac{1}{2}\tilde{z}^\top \left(\Sigma_{\psi,k}^{-1} + \frac{1}{\beta_{\mathrm{IB}}} W_{k,s}^\top \Lambda_{k,s} W_{k,s} \right) \tilde{z} + \tilde{z}^\top \left(\Sigma_{\psi,k}^{-1}\mu_{\psi,k} + \frac{1}{\beta_{\mathrm{IB}}} W_{k,s}^\top \Lambda_{k,s}\mu_{y,k} \right) + \mathrm{const} \end{aligned} \tag{26}$$

Matching the quadratic and linear coefficients gives

$$\Sigma_k^{\star -1} = \Sigma_{\psi,k}^{-1} + \frac{1}{\beta_{\mathrm{IB}}} W_{k,s}^\top \Lambda_{k,s} W_{k,s} \tag{27}$$

$$\mu_k^\star = \Sigma_k^\star \left(\Sigma_{\psi,k}^{-1}\mu_{\psi,k} + \frac{1}{\beta_{\mathrm{IB}}} W_{k,s}^\top \Lambda_{k,s} \mathbb{E}_{p_{\mathcal{D}_k}(Y|z,s)}[Y] \right) \tag{28}$$

Hence $p_k^\star(\tilde{z} \mid z,s) = \mathcal{N}(\mu_k^\star, \Sigma_k^\star)$, which proves Corollary 1. □

C.3 Single-Step Predictive-Rate Surrogate Descent

Assumption 1 (Frozen-statistics evolution step)**.** During one library operation in window k, the snapshot distribution $p_{\mathcal{D}_k}$, policy, encoder, predictors, variational marginal, retriever, usage weights, predictive-rate statistics, and semantic similarities are fixed. Exactly one add or remove operation is executed, after which these quantities are refreshed before another operation is considered.

Theorem 2 (Single-step predictive-rate surrogate descent)**.** *Under Assumption 1, an addition with $M_k^{\mathrm{add}}(z \mid \mathcal{B}) > 0$ strictly decreases $\mathcal{L}_k^{\mathrm{lib}}$, and a removal with $M_k^{\mathrm{keep}}(z \mid \mathcal{B}) < 0$ strictly decreases $\mathcal{L}_k^{\mathrm{lib}}$. Consequently, operations certified by the tests in Eq.* (12) *decrease the frozen-statistics surrogate whenever their confidence bounds cover the corresponding population marginals.*

Proof of Theorem 2. Fix evolution window k and all quantities listed in Assumption 1. For an addition $z \notin \mathcal{B}$, Eqs. (9) and (10) give

$$\mathcal{L}_k^{\mathrm{lib}}(\mathcal{B} \cup \{z\}) - \mathcal{L}_k^{\mathrm{lib}}(\mathcal{B}) = [C_k - V_k(\mathcal{B} \cup \{z\})] - [C_k - V_k(\mathcal{B})] = -M_k^{\mathrm{add}}(z \mid \mathcal{B}) \tag{29}$$

Therefore, if $M_k^{\mathrm{add}}(z \mid \mathcal{B}) > 0$, the frozen-statistics loss strictly decreases.

For removal of a stored skill $z \in \mathcal{B}$, Eq. (11) similarly gives

$$\mathcal{L}_k^{\mathrm{lib}}(\mathcal{B} \setminus \{z\}) - \mathcal{L}_k^{\mathrm{lib}}(\mathcal{B}) = [C_k - V_k(\mathcal{B} \setminus \{z\})] - [C_k - V_k(\mathcal{B})] = M_k^{\mathrm{keep}}(z \mid \mathcal{B}) \tag{30}$$

Hence $M_k^{\mathrm{keep}}(z \mid \mathcal{B}) < 0$ also implies strict descent. When the one-sided bounds in Eq. (12) simultaneously cover their respective population marginals, the sign certified by each test implies the corresponding condition above. The separately measured rollout-utility gate does not enter this algebra because it has different units and serves only as an additional empirical safety condition.

The argument applies to exactly one operation while the window-k quantities remain frozen. Once the library changes, the data distribution, usage weights, predictors, retrieval events, and pairwise redundancies may change, so the training process refreshes them before another operation. This result implies no claim of monotone environment return, cross-window global descent, or convergence of joint policy–encoder training. □

D Additional Experimental Details

D.1 Construction of the Motivation Figure

Figure 1 is built from frozen full-text skill-prompting agents on the three benchmarks with Qwen2.5-3B-Instruct. For panel (a), we segment each retrieved skill into clauses and measure the *next-action contribution* of a clause as the change in the model probability assigned to the action executed by the frozen policy at that decision state when that clause is masked from the prompt, averaged over the decision states in which the skill is retrieved. The clause with the largest contribution is plotted against the current task step. For panel (b), the *decision utility* of a skill is the mean paired return difference between skill-retained and skill-masked rollouts on the same task, seed, and initial state, and *retrievals per 1,000 decisions* is the empirical retrieval frequency normalized by the total number of decision steps. Both quantities are estimated over the same fixed evaluation window.

D.2 Examples of Skill-Library Corruptions

To test robustness under plausible but harmful entries, we construct three corruption sets from clean ALFWorld skills while preserving the benchmark's task vocabulary and object–receptacle relations, so that corruption cannot be detected from superficial domain mismatch alone. Redundant entries preserve the original procedure but paraphrase or lightly rewrite it. Irrelevant entries remain household- and object-related but do not apply to the current task or state. Misleading entries mention the correct goal or objects while introducing an incorrect action order or execution constraint. Table 10 gives representative examples of each corruption type; these illustrate the corruption rules rather than an exhaustive list of injected entries.

Table 10: Representative constructions for redundant, irrelevant, and misleading skill entries.

Type	Task or decision state	Injected skill entry	Reason for classification
Redundant	Clean a mug and place it in a cabinet.	"Pick up the mug, rinse it in the sink basin, and then store it in the cabinet."	Paraphrases an existing clean-then-place skill without changing its objects, order, or constraints.
Redundant	Heat a potato and place it on the countertop.	"Warm the potato in the microwave before putting it on the counter."	Restates the same heat-then-place procedure with minor lexical changes.
Redundant	Examine an alarm clock under a desk lamp.	"Carry the alarm clock to the desk lamp, switch on the light, and inspect the clock."	Duplicates an existing look-under-light procedure in different wording.
Irrelevant	The agent is holding a potato beside the microwave for a heating task.	"To cool the potato, place it in the refrigerator and close the door."	Uses the same object and household domain but addresses a different task objective.
Irrelevant	The agent is holding a dirty mug and must clean it in the sink.	"To inspect the mug, carry it to a desk lamp and turn the lamp on."	Refers to the correct object but provides a procedure unrelated to the required state transition.
Irrelevant	One apple has been placed in a bowl and the task requires placing a second apple there.	"Open the microwave, put the apple inside, and heat it."	Shares the object and environment vocabulary but is inapplicable to the remaining subgoal.
Misleading	The agent is holding a potato beside the microwave and must heat it before placement.	"Place the potato on the countertop first, and then heat it in the microwave."	Mentions the correct actions and objects but reverses the required action order.
Misleading	The agent is holding a dirty mug that must be cleaned before being stored.	"Put the mug in the cabinet before washing it in the sink basin."	Gives a locally plausible procedure with the clean-and-place order reversed.
Misleading	One apple has been placed in a bowl, but the task requires placing two apples there.	"The task is complete after placing the first apple; do not retrieve the second one."	Introduces an incorrect cardinality constraint that terminates the procedure prematurely.