Cross-Domain Contrastive Training of Embedding Models for Insight-Guided Agentic Reasoning

Tsz Ting Chung[1] **Mo Yu**[2†] **Jie Zhou**[2†] **Dit-Yan Yeung**[1]
[1]The Hong Kong University of Science and Technology
[2]WeChat AI, Tencent
ttchungac@connect.ust.hk

Abstract

Retrieving precise strategic insights at each decision-making step is critical for Large Language Model (LLM) agents. However, training effective retrieval models is often hindered by a scarcity of in-domain data and the inherent discrepancy between surface-level semantic similarity and functional relevance. In this work, we demonstrate that insight retrieval is fundamentally a procedural matching problem—the task of mapping concrete situations to abstract guiding rules—and show that this capability is transferable across domains. We propose **InsightEmb**, a contrastive training framework designed to learn abstract reasoning insights by training exclusively on mathematical reasoning data. Without exposure to in-domain training data, InsightEmb significantly outperforms base embedding models on diverse tasks, including the ALFWorld embodied environment, WebShop online shopping interactions, and SRA-bench agentic skill retrieval. Our results bridge the existing gap in standard embedding models by introducing a reasoning-aware embedding space.

1 Introduction

LLM-based agents in interactive environments—web navigation (Yao et al., 2022), embodied tasks (Shridhar et al., 2021), tool use (Schick et al., 2023)—must select actions under large action spaces where the optimal strategy depends on abstract reasoning rather than surface-level pattern matching. Providing agents with *insights*, abstract rules distilled from past experience (e.g., "check likely locations before exploring randomly"), has proven effective (Majumder et al., 2023; Wang et al., 2024), but the agent must dynamically retrieve the most pertinent insight at each step.

This dynamic retrieval poses a unique challenge that we call the *abstraction gap*: the query (the agent's current observation and action history) and the target (an abstract insight) live at fundamentally different levels of abstraction. Standard embedding models match on surface features—keywords like "mug" or "stoveburner"—rather than functional relevance such as "this situation requires systematic exploration." Consequently, off-the-shelf retrievers return topically related but strategically irrelevant insights.

Our key observation is that this *situation-to-insight* matching problem is not domain-specific. In mathematics, a probability problem about "Mary and James sitting in 7 chairs" must be matched to the insight "use complementary counting"—with zero keyword overlap. In an embodied task, "put a hot potato in fridge" must be matched to "find → transform → place." Both require the same structural capability: connecting a concrete situation to the abstract rule that resolves it.

We propose **InsightEmb**, a contrastive training framework that exploits this domain-agnostic structure. InsightEmb trains an embedding model in two stages, *entirely on mathematical reasoning data*: (1) *Situation-to-insight matching* teaches the model to bridge the abstraction gap by mapping math problems and partial solution trajectories to their relevant heuristic rules; (2) *Situation-to-experience matching* teaches structural similarity recognition between reasoning trajectories, reinforcing the model's ability to identify when different-looking situations require the same strategy. At inference time, the trained model retrieves insights for an LLM agent in ALFWorld (Shridhar et al., 2021)—an embodied household-task benchmark—*without any ALFWorld-specific training data*.

Our experiments yield three findings:

- **Large cross-domain gains.** InsightEmb improves ALFWorld success rates by +10.0% in-distribution and +2.2% out-of-distribution over the base Qwen3-Embedding-4B model, despite training exclusively on math data (§5).

†Co-corresponding authors.

- **Cross-domain > in-domain.** A model trained on math outperforms one trained on ALFWorld data by +7.1%, indicating that the diversity of mathematical reasoning provides a richer training signal than scarce in-domain data (§5).
- **Transfer to a second environment.** On WebShop, InsightEmb achieves the highest average task score among all embedding variants, confirming that the learned structural matching capability generalizes beyond embodied household tasks to web-based shopping (§5.5).
- **Complementary stages.** Stage 1 contributes the primary improvement; Stage 2 adds further gains by teaching structural problem-type recognition (§5.3).
- **Robust scaling and LLM-agnostic.** InsightEmb's advantage grows with multi-insight retrieval (top-5 Bundle achieves 66.43%) and transfers to GPT-5.2, where BUNDLE/InsightEmb reaches 67.85% (+2.85% over Base) and ATOMIC/InsightEmb reaches 67.14% (+7.85% over Base), confirming that the improvement stems from retrieval quality rather than LLM-specific interactions (§5.6, §5.7).

2 Related Work

Retrieval-Augmented LLM Agents. RAG (Lewis et al., 2020; Guu et al., 2020) has been extended to agentic settings for retrieving past experiences (Shinn et al., 2023; Majumder et al., 2023), tool documentation (Qin et al., 2024), and skill libraries (Wang et al., 2024). These systems retrieve *concrete* artifacts—specific examples, code, or API descriptions—where surface similarity is a reasonable proxy for relevance. In contrast, we retrieve *abstract insights* (general heuristic rules), where the query and target share no surface overlap, making standard retrieval ineffective. Moreover, prior agentic RAG systems use off-the-shelf or in-domain–trained retrievers; we show that a retriever trained on an unrelated domain can outperform both.

Experience-Based Agent Learning. Reflexion (Shinn et al., 2023) improves agents through verbal self-reflection; CLIN (Majumder et al., 2023) accumulates causal abstractions across episodes; Voyager (Wang et al., 2024) builds reusable skill libraries. These methods focus on *what* knowledge to accumulate. Our work addresses the orthogonal question of *how* to retrieve the right piece of accumulated knowledge at the right time, and is compatible with any knowledge-accumulation strategy.

Contrastive Training for Dense Retrieval. Dense passage retrieval (Karpukhin et al., 2020; Xiong et al., 2021) and instruction-aware embeddings (Wang et al., 2022; Su et al., 2023) have advanced general-purpose retrieval. However, these models are trained on query–document pairs where relevance correlates with lexical and topical overlap. Insight retrieval requires matching across abstraction levels—a capability not targeted by existing training objectives. Our contrastive curriculum explicitly trains for this abstraction-bridging property.

Cross-Domain Transfer. Prior cross-domain transfer work focuses on transferring the *reasoning capabilities* of LLMs themselves (Wei et al., 2022; Kojima et al., 2022). We transfer a different capability—the *retrieval geometry* of an embedding model—showing that the embedding-space structure learned from math (situations near their guiding rules) generalizes to embodied-task retrieval without any domain adaptation.

3 Method

3.1 Problem Setting

An LLM agent operates in an interactive environment where, at each step t, it observes a state s_t (task description, action history, current observation) and must select an action a_t. The agent has access to an insight corpus $\mathcal{I} = \{I_1, \ldots, I_N\}$, where each insight I_i is a natural-language description of abstract rules and strategies. We seek an embedding model f_θ that maps both states and insights into a shared space so that the most relevant insight can be retrieved via cosine similarity:

$$I^* = \arg\max_{I \in \mathcal{I}} \ \mathrm{sim}\big(f_\theta(s_t),\ f_\theta(I)\big). \qquad (1)$$

The core difficulty is the *abstraction gap*: s_t contains concrete details (specific objects, locations, actions) while I^* contains abstract rules (general strategies, heuristics). We address this gap through a two-stage contrastive curriculum trained entirely on mathematical reasoning data, exploiting the structural parallel between math insight retrieval and agentic insight retrieval.

3.2 Stage 1: Situation-to-Insight Matching

Stage 1 teaches the embedding model to bridge the abstraction gap by matching mathematical prob-

lems to their relevant heuristic rules.

Training data. Each training example is a triplet (q, I^+, I^-) where q is a query in one of three forms: (i) a raw math problem statement (*query-only*), (ii) a problem concatenated with its full chain-of-thought solution (*full trajectory*), or (iii) a problem with a truncated solution (*partial trajectory*). I^+ is a positive insight set containing rules helpful for solving the problem; I^- is a negative (irrelevant) insight set.

Multi-granularity invariance. A critical design choice is that all three query forms for the same base problem share *identical* positive and negative insight sets. This forces the model to learn that the same insight should be retrievable regardless of how much progress has been made—a property directly needed for dynamic agentic retrieval, where the query grows with each step.

Training objective. We use the InfoNCE contrastive loss with in-batch negatives:

$$\mathcal{L}_1 = -\log \frac{\exp\big(\text{sim}(f_\theta(q), f_\theta(I^+))/\tau\big)}{\sum_{I \in \{I^+\} \cup \mathcal{N}} \exp\big(\text{sim}(f_\theta(q), f_\theta(I))/\tau\big)} \quad (2)$$

where $\tau = 0.01$ and $\mathcal{N}$ includes in-batch negatives (training group size 11). The data spans three math domains—counting & probability, number theory, and geometry—totaling 4,943 examples.

3.3 Stage 2: Situation-to-Experience Matching

Stage 1 teaches the model to match situations to *abstract rules*; Stage 2 complements this by teaching it to match situations to *structurally similar solved problems*. This reinforces the model's ability to recognize when two different-looking situations require the same reasoning approach.

The training data consists of triplets (q, T^+, T^-) where q is a raw problem or a full trajectory, T^+ is a structurally similar solved problem, and T^- is a dissimilar one. The loss is identical to Eq. 2 with trajectories replacing insights. This stage uses 2,896 examples across the same three math domains.

3.4 Inference: Dynamic Insight Retrieval

At inference time, the trained embedding model is deployed for dynamic insight retrieval in the target environment (ALFWorld) *without any domain-specific fine-tuning*. At each step t: (1) the agent's current state s_t is encoded with a task-specific instruction prefix;[1] (2) the top-k most similar insights are retrieved from the pre-encoded corpus; (3) the retrieved insights are prepended to the LLM's prompt for action generation.

4 Experimental Setup

4.1 Training

We fine-tune Qwen3-Embedding-4B (Qwen Team, 2025) using DeepSpeed ZeRO-3 on 8 GPUs. Both stages use learning rate 1×10^{-5} with cosine scheduling, per-device batch size 8 with 2 gradient accumulation steps, temperature τ=0.01, training group size 11, maximum passage length 1,536 tokens, 6 epochs, and warmup ratio 0.1. Stage 2 initializes from the Stage 1 checkpoint. Full hyperparameters are in Appendix E.

4.2 Evaluation Environment

We evaluate on ALFWorld (Shridhar et al., 2021), a text-based embodied environment where agents perform household tasks (e.g., "put a hot potato in fridge," "clean a mug and place it in coffeemachine"). Each task requires a sequence of navigation and interaction actions. We use Qwen3-8B as the action-generating LLM with greedy decoding, a history window of 3 steps, maximum 50 steps per episode, and top-1 insight retrieval. We evaluate on both in-distribution (140 games) and out-of-distribution (134 games) splits (seed 42).

4.3 Baselines and Variants

Embedding models. We compare three embedding configurations: (i) **Base**: the unmodified Qwen3-Embedding-4B; (ii) **In-domain**: the base model fine-tuned on ALFWorld-specific retrieval data; (iii) **InsightEmb** (ours): the base model trained with our two-stage math curriculum.

Insight corpora. We evaluate two insight corpus granularities per environment to disentangle the effect of the embedding model from the insight content. BUNDLE insights are multi-rule summaries, each containing several rules with full chain-of-thought reasoning. ATOMIC splits each bundle into individual rules with chain-of-thought

[1] Instruction: "*Given a query or an experience, retrieve relevant content that implicitly reflects the insights for solving the query, addresses issues in the provided experience, or matches the insights behind the experience (which is more than just similarity).*"

removed. Table 1 summarises the corpus statistics for both environments. Additional corpus variants (e.g., forward/reverse generation orders) are reported in Appendix **??**.

4.4 WebShop Evaluation

To validate cross-domain transfer beyond embodied tasks, we additionally evaluate on WebShop (Yao et al., 2022), a web-based shopping environment where agents navigate product pages to purchase items matching natural-language instructions. WebShop tasks require understanding product attributes, comparing options, and executing multi-step purchase workflows—a structurally different challenge from ALFWorld's spatial navigation. We use the same Qwen3-8B action-generating LLM with greedy decoding, a history window of 3 steps, maximum 50 steps per episode, top-1 dynamic insight retrieval, and 4 parallel environments. We evaluate on 500 test games (seed 42). We compare the same embedding model variants: Base, InsightEmb (Stage 1 only), and InsightEmb (Stage 1 + Stage 2).

4.5 SRA-Bench Retrieval Evaluation

To isolate retrieval quality from downstream action generation, we additionally evaluate on SRA-Bench, a static skill-retrieval benchmark spanning theorem proving, logical reasoning, tool use, contest math, medical calculation, and code-generation tasks. Each query is formed from the task information, and each candidate is represented by the full skill content, including the skill name, description, and body. We compare the original Qwen3-Embedding-4B checkpoint (Base) with our cross-domain trained checkpoint (InsightEmb) using recall at k (R@k) and normalized discounted cumulative gain at k (N@k) for $k \in \{1, 3, 5, 7, 10\}$. Following the benchmark setting, we report macro averages across datasets so that each task family contributes equally regardless of its number of examples.

5 Results

5.1 Main Results

Table 2 compares embedding models across insight corpus variants.

Three findings emerge from Table 2:

Cross-domain training yields large gains. InsightEmb with ATOMIC insights achieves 65.00% in-distribution and 57.46% OOD, improving over the base model (same corpus) by **+10.0%** and **+2.2%** respectively. Since InsightEmb is trained exclusively on mathematical reasoning data, these gains are entirely from cross-domain transfer—no ALFWorld data is used for training.

Cross-domain training outperforms in-domain training. InsightEmb with ATOMIC (65.00%) surpasses the in-domain–trained retriever (57.86%) by **+7.1%** in-distribution, despite never seeing any ALFWorld data. We attribute this to two factors: (1) math training covers three diverse reasoning domains, providing richer contrastive signal than the limited ALFWorld training data; (2) mathematical insights span a wider range of abstraction levels, from concrete formulas to meta-strategies, yielding better abstraction-gap training.

Corpus granularity matters. For the base model, splitting insights into individual rules (ATOMIC) improves OOD performance by +8.95% over multi-rule bundles (BUNDLE): 55.22% vs. 46.27%. This suggests that surface-matching models struggle to identify the relevant rule within a multi-rule bundle, and that finer-grained corpora are beneficial for retrieval. We analyze this interaction between corpus granularity and embedding quality in §**??**.

5.2 Static Skill Retrieval on SRA-Bench

Table 3 reports macro-averaged SRA-Bench retrieval performance across six task families. InsightEmb improves over Base on every cutoff, with small gains at R@1 (+1.04) and substantially larger gains at higher cutoffs (+8.90 R@10 and +5.78 N@10). This pattern indicates that cross-domain contrastive training primarily improves the ranking of relevant skills beyond the first position, which is especially useful when an agent can inspect or condition on multiple retrieved skills.

The only clear negative outlier is MedCalcBench at small cutoffs (Appendix A). A plausible explanation is that medical calculation skills are highly template-like, lexically specialized, and domain-specific: correct retrieval often depends on exact medical score names, disease terms, biomarker or variable names, units, and other clinical terminology. The Base embedding model appears to preserve these fine-grained biomedical and entity-level cues, whereas InsightEmb's cross-domain training encourages more abstract structural matching and may therefore downweight or smooth over such terminology at rank 1. This interpretation is also con-

Environment	Corpus	# Insights	Avg Words	Range
ALFWorld	BUNDLE	2,419	494	90–1,114
	ATOMIC	21,916	25	1–365
WebShop	BUNDLE	501	549	176–1,064
	ATOMIC	5,212	24	1–371

Table 1: Insight corpus statistics. BUNDLE = multi-rule summaries with chain-of-thought; ATOMIC = individual rules extracted from bundles with chain-of-thought removed. Range = min–max words per insight.

Insight	Embedding	In-D	OOD
Baseline (no fine-tuning)			
{}	—	52.86	55.22
BUNDLE	Base	54.29	46.27
ATOMIC	Base	55.00	55.22
In-domain fine-tuning			
BUNDLE	In-domain	57.86	59.70
Cross-domain (ours)			
BUNDLE	InsightEmb	**67.14**	**62.69**
ATOMIC	InsightEmb	**65.00**	**57.46**

Table 2: ALFWorld success rates (%). In-D = in-distribution (140 games), OOD = out-of-distribution (134 games). Since InsightEmb is trained exclusively on math data, every row under "Cross-domain" is a zero-shot transfer result. The In-domain retriever uses its own best-performing corpus variant. All use top-1 retrieval, 50 max steps, history length 3. Additional corpus variants in Appendix **??**.

sistent with the original SRA-Bench results, where MedCalcBench already achieves about 90% R@1 and over 90% R@10 after reranking the BM25 top-50 candidates with different reranker models, indicating that lexical candidate generation is less of a bottleneck for this task. The gap shrinks as k increases and disappears in recall by R@7/R@10, suggesting that InsightEmb still places the correct medical-calculation skill within the short candidate list but often not at the very top. In contrast, task families such as theorem proving, tool use, contest math, and code generation benefit from abstraction-aware ranking because the query and useful skill can differ substantially in wording while sharing a reasoning or procedural structure. This distinction is important in practical retrieval-augmented skill selection pipelines: an LLM reranker can only inspect a limited candidate set, such as the BM25 top-50 skills, due to context-window constraints. Thus, task families for which lexical retrieval fails to place the correct skill within this window constitute the more important bottleneck. From this perspective, the strong results on LogicBench and CHAMP are especially encouraging, because they require more semantic or reasoning-oriented matching between task descriptions and skill content, where surface lexical overlap is weak. InsightEmb therefore improves the recall of relevant skills before reranking, expanding the effective coverage of downstream LLM-based selection.

5.3 Ablation: Training Stage Contributions

Table 4 isolates the contribution of each training stage. The combined two-stage training yields +10.00% in-distribution and +2.24% OOD. Stage-level ablation on alternative corpus variants (Appendix **??**) shows that Stage 1 provides the primary improvement by directly training the abstraction-bridging capability, while Stage 2 contributes additional OOD gains by teaching the model to recognize *structural problem types* (e.g., "find-and-place" vs. "clean-and-place"), a capability particularly valuable when specific objects and locations are unfamiliar.

5.4 Per-Task-Type Breakdown

Table 5 breaks down in-distribution success rates by ALFWorld task type. InsightEmb leads in 4 of 6 task types, with the largest gains on *clean* tasks (+36% over Base, +27% over In-domain) and *heat* tasks (+16% over Base). The improvement is not concentrated in a single task type, confirming that the cross-domain training teaches a general structural matching capability rather than a task-specific shortcut.

Notably, *clean* tasks show the most dramatic improvement: InsightEmb achieves 59% vs. 23% for Base—nearly a 3× improvement. Clean tasks require a multi-step workflow (find object → go to sink → clean → place), and the retrieved insights provide explicit spatial logic chains (e.g., "kitchenware near sinks") that guide the agent through this workflow. The In-domain model leads only on *examine* tasks (76% vs. 65%), where ALFWorld-specific patterns (lamp locations) may provide an advantage.

Embedding	R@1	R@3	R@5	R@7	R@10	N@1	N@3	N@5	N@7	N@10
Base	30.33	40.84	46.40	49.83	54.13	33.82	37.62	39.96	41.23	42.64
InsightEmb	**31.37**	**46.39**	**54.25**	**58.56**	**63.03**	**36.29**	**42.00**	**45.29**	**46.92**	**48.42**
Δ	+1.04	+5.55	+7.85	+8.73	+8.90	+2.47	+4.38	+5.33	+5.69	+5.78

Table 3: SRA-Bench macro-average retrieval results across six task families. Queries use task information and candidates use the full skill content. R@k is recall at k and N@k is normalized discounted cumulative gain at k. Per-task details for @1 and @10 are provided in Appendix A.

Configuration	In-D (%)	OOD (%)
Base model	55.00	55.22
InsightEmb (Stage 1 + Stage 2)	**65.00**	**57.46**
Total Δ	+10.00	+2.24

Table 4: Overall training effect on the ATOMIC insight corpus. S→I = situation-to-insight; S→E = situation-to-experience. Stage-level ablation with per-stage breakdowns is in Appendix **??**.

Task Type	Base	In-dom.	Ours
clean (22)	23	32	**59**
cool (26)	65	65	**77**
heat (19)	37	42	**53**
put (48)	73	73	**79**
examine (17)	59	**76**	65
find_two (8)	**25**	12	**25**
Overall	54.3	57.9	**67.1**

Table 5: In-distribution success rates (%) by task type. Base = ATOMIC/Base, In-dom. = In-domain (best corpus), Ours = ATOMIC/InsightEmb. Numbers in parentheses indicate total games per type.

5.5 WebShop Results

Table 6 presents results on WebShop, a second cross-domain evaluation environment. Unlike ALFWorld's binary success metric, WebShop uses a continuous task score (0–1) reflecting partial credit for attribute matching, making average task score the primary metric.

Three findings emerge from the WebShop evaluation:

InsightEmb transfers to web shopping. All InsightEmb variants outperform the Base embedding model on both success rate and average task score. The best average task score (0.329) is achieved by InsightEmb (Stage 1) with BUNDLE insights, representing a **+0.148** improvement over Base (0.181) on the same corpus—an 82% relative gain. This confirms that the structural matching capability learned from math transfers to a second, structurally distinct interactive environment.

Insight	Embedding	Succ.	SR (%)	Avg Score
{}	—	28	5.6	0.309
BUNDLE	Base	5	1.0	0.181
BUNDLE	InsightEmb	**12**	**2.4**	**0.328**
ATOMIC	Base	9	1.8	0.283
ATOMIC	InsightEmb	**20**	**4.0**	**0.290**

Table 6: WebShop results (500 test games, dynamic retrieval). SR = success rate, Avg Score = average task score (0–1). All models trained exclusively on math data. InsightEmb variants consistently outperform Base across both insight corpora.

Stage 2 improves success rate. With ATOMIC insights, InsightEmb (Stage 1+2) achieves the highest success rate (4.0%, 20/500) compared to Base (1.8%, 9/500), a **2.2**× improvement. This suggests that Stage 2's situation-to-experience matching helps the agent complete full purchase workflows, not just partial attribute matching.

WebShop remains challenging. Overall success rates are low across all configurations (1–4%), reflecting WebShop's inherent difficulty: the agent must navigate complex product pages, compare multiple options, and execute precise purchase sequences. However, the consistent advantage of InsightEmb over Base demonstrates that even in this challenging setting, cross-domain–trained retrieval provides meaningful benefits.

Qualitative patterns. Win/loss analysis reveals that InsightEmb (Stage 1+2) with BUNDLE wins 8 games that Base loses, while Base wins only 1 game that InsightEmb loses—an 8:1 dominance ratio. InsightEmb also reduces zero-score games from 361 to 247 (a 31.6% reduction), indicating that it helps the agent at least partially complete tasks that Base completely fails on. Manual inspection of the divergent games reveals three recurring patterns (detailed examples in Appendix H):

1. **Variant selection awareness.** InsightEmb's insights guide the agent to explicitly select product variants (color, size) before purchasing, while

Insight	Embedding	Top-k	SR (%)
Bundle insights			
BUNDLE	InsightEmb	3	60.00
BUNDLE	Base	3	50.00
BUNDLE	InsightEmb	5	**66.43**
BUNDLE	Base	5	48.57
BUNDLE	InsightEmb	7	56.43
BUNDLE	Base	7	43.57
Atomic insights			
ATOMIC	InsightEmb	3	63.57
ATOMIC	Base	3	62.14
ATOMIC	InsightEmb	5	60.71
ATOMIC	Base	5	63.57
ATOMIC	InsightEmb	7	**65.72**
ATOMIC	Base	7	60.00

Table 7: ALFWorld success rates (%) with varying numbers of retrieved insights (top-k). All experiments use Qwen3-8B, max 50 steps, history length 3. For reference, top-1 results: ATOMIC/InsightEmb = 65.00%, ATOMIC/Base = 55.00% (in-distribution).

Base frequently skips this step, resulting in partial scores instead of perfect scores.

2. **Loop prevention.** Base gets stuck in search–browse–back loops for 21–50 steps; InsightEmb's insights about session management and error recovery help the agent break out of unproductive cycles.
3. **Procedural sequencing.** InsightEmb retrieves insights that encode a sequential workflow (search → verify → select variants → buy), while Base retrieves topically relevant but procedurally vague insights.

These patterns mirror the ALFWorld finding (§5.8.5): InsightEmb performs *procedural matching* rather than *topical matching*, retrieving insights that address the agent's current bottleneck.

5.6 Scaling the Number of Retrieved Insights

All previous experiments use top-1 retrieval. We now investigate how performance scales when the agent retrieves multiple insights ($k \in \{3, 5, 7\}$) per step, evaluating on ALFWorld with Qwen3-8B. Table 7 presents the results.

Three findings emerge from the scaling analysis:

InsightEmb benefits more from multi-insight retrieval with Bundle. With BUNDLE insights, InsightEmb shows a clear advantage across all k values, with the gap widening at k=5 (+17.86% over Base: 66.43% vs. 48.57%). This is the highest absolute success rate achieved by any BUNDLE configuration, surpassing even the top-1 ATOMIC/InsightEmb result (65.00%). In contrast, Base performance *degrades* as k increases (50.00% → 48.57% → 43.57%), suggesting that additional retrieved insights introduce noise when the embedding model cannot reliably rank them.

Insight	Embedding	Top-k	Succ.	SR (%)	Avg Score
Bundle insights					
BUNDLE	InsightEmb	3	**12**	**2.4**	**0.332**
BUNDLE	Base	3	9	1.8	0.186
BUNDLE	InsightEmb	5	**12**	**2.4**	**0.335**
BUNDLE	Base	5	8	1.6	0.196
BUNDLE	InsightEmb	7	10	2.0	**0.326**
BUNDLE	Base	7	5	1.0	0.190
Atomic insights					
ATOMIC	InsightEmb	3	12	2.4	0.245
ATOMIC	Base	3	11	2.2	**0.293**
ATOMIC	InsightEmb	5	11	2.2	0.245
ATOMIC	Base	5	**13**	**2.6**	**0.288**
ATOMIC	InsightEmb	7	12	2.4	0.262
ATOMIC	Base	7	**13**	**2.6**	**0.287**

Table 8: WebShop results with varying numbers of retrieved insights (top-k). SR = success rate, Avg Score = average task score (0–1). With BUNDLE insights, InsightEmb consistently outperforms Base across all k values. With ATOMIC insights, Base performs comparably or slightly better.

Atomic insights show diminishing returns from scaling. With ATOMIC insights, the gap between InsightEmb and Base is smaller and less consistent. At k=3 and k=5, the two models perform comparably (63.57 vs. 62.14; 60.71 vs. 63.57), while at k=7 InsightEmb pulls ahead (65.72% vs. 60.00%). This suggests that when insights are already fine-grained (single rules), the marginal value of additional retrievals is lower, and InsightEmb's advantage manifests primarily at higher k where ranking quality becomes critical.

Optimal k depends on corpus granularity. For BUNDLE/InsightEmb, k=5 is optimal; for ATOMIC/InsightEmb, k=7 achieves the best result. This aligns with the intuition that multi-rule bundles are information-dense (each bundle contains several rules), so fewer retrievals suffice, while single-rule atomic insights require more retrievals to cover the relevant strategy space.

We additionally evaluate multi-insight retrieval on WebShop to test whether the scaling patterns transfer across environments. Table 8 presents the results.

Three findings emerge from the WebShop scaling analysis, the first two mirroring the ALFWorld patterns and the third revealing a granularity-dependent reversal.

Bundle scaling transfers to WebShop. With BUNDLE insights, InsightEmb maintains a large

and stable advantage over Base across all k values, with avg-score gaps of $+0.146$, $+0.139$, and $+0.136$ at k=3, 5, 7 respectively. The absolute gain (~$+0.14$ avg score; up to $+82\%$ relative) is substantially larger than what top-1 retrieval alone delivers, confirming that the ALFWorld finding—better embeddings unlock larger, information-dense insight units—generalizes to a second environment.

Atomic-corpus reversal: Base edges ahead on WebShop. Unlike ALFWorld, where ATOMIC/InsightEmb remains competitive with or ahead of Base at every k, on WebShop Base slightly outperforms InsightEmb under the ATOMIC corpus at every k (0.293 vs. 0.245 at k=3; 0.288 vs. 0.245 at k=5; 0.287 vs. 0.262 at k=7). We attribute this to the interaction between environment dynamics and corpus granularity: WebShop trajectories are short and goal-driven (a handful of variant/price decisions), so single-rule atomic insights are already near-sufficient, leaving little headroom for a stronger ranker; at the same time, InsightEmb was optimized to discriminate *bundle-sized* situational patterns, so when the corpus is pre-decomposed into rules the structural advantage is neutralized. This is consistent with our ablation finding that the gains from InsightEmb scale with insight-unit density rather than with simple retrieval recall.

Bundle is the dominant configuration for InsightEmb across environments. Combining these results with Table 7, the overall picture is robust: BUNDLE/InsightEmb is the best configuration on both ALFWorld (SR 66.43% at k=5) and WebShop (avg score 0.335 at k=5), whereas switching to ATOMIC is only preferable for the weaker Base retriever. In deployment, this argues for pairing InsightEmb with bundle-level insight corpora whenever chain-of-thought or multi-rule summaries are available.

5.7 Cross-LLM Generalization

To verify that InsightEmb's advantage is not specific to Qwen3-8B, we evaluate with GPT-5.2 as the action-generating LLM using top-1 retrieval across all insight configurations. Table 9 shows the results.

To further validate cross-LLM generalization, we also evaluate with DeepSeek-R1 as the action-generating LLM using top-1 retrieval. Table 10 shows the results.

Insight	Embedding	SR (%)
{}	—	61.43
BUNDLE	InsightEmb	**67.85**
BUNDLE	Base	65.00
ATOMIC	InsightEmb	67.14
ATOMIC	Base	59.29

Table 9: ALFWorld success rates (%) with GPT-5.2 as the action-generating LLM and top-1 retrieval. {} denotes the no-insight baseline. InsightEmb achieves the highest success rate (67.85%) and consistently outperforms Base across both insight granularities.

Insight	Embedding	SR (%)
BUNDLE	InsightEmb	**71.43**
BUNDLE	Base	61.43
ATOMIC	InsightEmb	**71.43**
ATOMIC	Base	66.43

Table 10: ALFWorld success rates (%) with DeepSeek-R1 as the action-generating LLM and top-1 retrieval. InsightEmb consistently outperforms Base across both insight granularities, with +10.0% for BUNDLE and +5.0% for ATOMIC.

We also evaluate cross-LLM generalization on WebShop with DeepSeek-R1 as the action-generating LLM using top-1 retrieval. Table 11 shows the results.

Several findings emerge. First, the no-insight baseline already reaches 61.43%, reflecting GPT-5.2's strong zero-shot capability. Nevertheless, insight-augmented retrieval with InsightEmb pushes performance substantially higher: BUNDLE/InsightEmb achieves **67.85%** (+6.42% over no-insight) and ATOMIC/InsightEmb reaches 67.14% (+5.71%). Second, InsightEmb consistently outperforms Base for both granularities—by +2.85% for BUNDLE and +7.85% for ATOMIC—confirming that the retrieval improvement transfers across LLMs. Third, the ATOMIC/InsightEmb advantage (+7.85% over Base) is even larger than observed with Qwen3-8B (+10.0% in-distribution), suggesting that InsightEmb's precise retrieval is especially valuable when the action-generating LLM is already strong and can better exploit well-matched insights.

5.8 Analysis

5.8.1 Why Does Cross-Domain Transfer Work?

The success of transfer rests on a structural parallel between math and embodied tasks: both are se-

Insight	Embedding	Succ.	SR (%)	Avg Score
BUNDLE	InsightEmb	101	20.2	0.239
BUNDLE	Base	64	12.8	0.189
ATOMIC	InsightEmb	117	23.4	0.290
ATOMIC	Base	**145**	**29.0**	**0.353**

Table 11: WebShop results (500 test games) with DeepSeek-R1 as the action-generating LLM and top-1 retrieval. SR = success rate, Avg Score = average task score (0–1). With BUNDLE insights, InsightEmb outperforms Base on both success rate (+7.4%) and average score (+0.050). With ATOMIC insights, Base achieves higher performance.

Component	Math	ALFWorld
Situation	Problem statement	Task + observation
Trajectory	Solution steps	Action history
Insight	Solving heuristic	Execution strategy
Abstraction gap	"chairs" ↔ "complementary counting"	"stoveburner" ↔ "find-transform-place"

Table 12: Structural analogy between math reasoning and embodied task execution that enables cross-domain transfer.

quential decision-making problems where abstract rules guide concrete actions. Table 12 makes this parallel explicit.

Contrastive training on math teaches a *domain-agnostic geometric property*: situations should be embedded near the abstract rules that resolve them. Because the query instruction prefix provides a domain-agnostic framing (§3.4), this geometric property activates for ALFWorld queries without domain adaptation.

Embedding space analysis (Appendix D) reveals that InsightEmb does not cluster insights by topic better than the Base model—in fact, Base achieves higher Silhouette scores (0.053 vs. 0.036) and nearest-neighbor label purity. Instead, InsightEmb compresses the space (mean pairwise similarity 0.82 vs. 0.57), forcing the retrieval decision to depend on *fine-grained structural cues* rather than coarse topical separation. The downstream result is that InsightEmb retrieves the procedurally correct insight (§5.8.5), even though its embedding space looks less cleanly clustered by topic.

5.8.2 Multi-Granularity Query Invariance

Our Stage 1 design enforces that query-only, partial-trajectory, and full-trajectory versions of the same problem all retrieve the same insight. This directly maps to the agentic setting: at step 0

Task: "heat some tomato and put it in fridge"

Base model retrieves (ATOMIC/Base):
"Verify Object State Before Interaction — Ensure the object meets required conditions."
→ Agent goes to `cabinet 1`, then `cabinet 2`, `cabinet 3`...
→ **Timeout at 50 steps** (never checks fridge for tomato)

InsightEmb retrieves (ATOMIC/InsightEmb):
*"Establish Container Priority Hierarchy — Check FOOD-related containers (**fridge**, diningtable) first for perishables."*
→ Agent goes to `fridge 1` → finds tomato → take → microwave → heat → fridge → put
→ **Success in 16 steps**

Figure 1: Qualitative comparison on Game 124 (in-distribution). The base model's generic insight leads to exhaustive cabinet search; InsightEmb retrieves a domain-specific container hierarchy that directs the agent to the fridge first.

the agent has only the task description (analogous to query-only), at mid-game it has partial action history (partial trajectory), and at late steps it has extensive history (full trajectory). The multi-granularity training ensures that the correct insight is retrievable at every stage of task execution, not just at the beginning.

5.8.3 Corpus Granularity: Multi-Rule vs. Single-Rule Insights

An interesting pattern emerges from comparing BUNDLE (multi-rule bundles with chain-of-thought) and ATOMIC (individual rules without chain-of-thought). For the base model, splitting into single rules improves OOD performance substantially (+8.95%), because surface-matching models struggle to identify the relevant rule within a multi-rule bundle. This indicates that finer-grained insight corpora are generally beneficial for retrieval, as they reduce the noise from irrelevant co-located rules and allow the embedding model to match against a single, focused strategy.

5.8.4 Qualitative Example: Insight-Guided Search

Figure 1 illustrates how InsightEmb's retrieved insights lead to better action selection on a representative task ("heat some tomato and put it in fridge").

The base model's insight ("Verify Object State") is topically relevant but strategically vacuous—it does not indicate *where* to find a tomato. InsightEmb retrieves a structurally matched insight that encodes a concrete search priority (fridge first for perishables), directly determining the agent's

Rule Category	Per-Game Avg.		Aggregate	
	Base	Ours	Base	Ours
WHERE (search)	4.3	**7.8**	17	**31**
HOW (state change)	**3.3**	0.5	13	2
PLACE (destination)	0.8	0.3	3	1
OTHER	1.8	1.5	7	6
Total	10	10	40	40

Table 13: Rule category distribution in top-10 retrieved rules from the ATOMIC corpus (5,251 individual rules). Aggregated across 4 case study games. InsightEmb overwhelmingly retrieves WHERE rules (77.5% vs. 42.5%), while Base over-retrieves HOW rules.

Category	Base			InsightEmb		
	0–4	5–14	15+	0–4	5–14	15+
Search & Locate	27.5	18.5	22.6	**65.6**	**51.9**	**51.1**
Verification	9.8	16.4	14.5	17.4	30.5	32.9
State Transform	20.5	25.2	24.3	2.6	6.3	7.5
Navigation	20.7	17.6	17.3	5.1	1.2	1.3
Task Planning	10.3	10.6	11.7	5.5	7.9	5.6
Placement	11.1	11.6	9.5	2.5	1.1	1.0

Table 14: Retrieved insight category distribution (%) across task phases (step ranges). Base retrieves a roughly uniform mix across all phases; InsightEmb concentrates on Search & Locate early (65.6%) and shifts toward Verification later (32.9%), showing phase-aware retrieval.

first action. This pattern—generic vs. structurally specific retrieval—recurs across the 13 games that InsightEmb wins uniquely (see Appendix C for additional examples).

5.8.5 Topical vs. Procedural Retrieval: Rule-Level Analysis

A natural question is whether InsightEmb's improvement comes from better *topical matching* (retrieving insights that share the task's topic) or genuine *procedural understanding* (retrieving insights that address the task's current bottleneck). To answer this, we leverage the ATOMIC insight corpus, which contains 5,251 *individual* rules (avg. 194 chars each) rather than multi-rule bundles. Because each insight is a single rule, we can examine exactly which *type* of rule each embedding model prioritizes for the same query. We classify each retrieved rule into functional categories—WHERE (search strategy), HOW (state-change procedures), PLACE (destination logic), and VERIFY (state checks)—and compare the top-10 retrieved rules for each case study game.

Table 13 reveals a striking pattern: InsightEmb retrieves WHERE rules 77.5% of the time (31/40), while Base retrieves them only 42.5% (17/40). Conversely, Base retrieves HOW rules 32.5% of the time (13/40) vs. only 5% (2/40) for InsightEmb.

This difference is most dramatic in Game 124 ("heat some tomato and put it in fridge"). The Base model retrieves 7/10 rules about *how to heat*—"Thermal Management Protocol," "Appliance Utilization for State Modification," "Validate Heating Requirements"—which describe the heating *procedure* but not *where to find the tomato*. InsightEmb retrieves 10/10 rules about *where to search*—"Prioritize High-Probability Containers First," "Check food-related containers (fridge, countertops)," "Contextual Container Prioritization"—which address the actual bottleneck: the agent's first action must be to *locate* the tomato.

This reveals a key difference in how the two models understand task structure:

- **Base** matches the query to rules that share the task's *topic* (heating → heating rules). This is *topical matching*.
- **InsightEmb** matches the query to rules that address the task's *current bottleneck* (the agent hasn't found the object yet → search rules). This is *procedural matching*.

The procedural matching is correct: in ALFWorld, the agent must first *find* the target object before it can apply any state transformation. InsightEmb has learned this sequential dependency—not from ALFWorld data, but from the structural parallel in math, where a problem must first be *understood* (matched to the right strategy) before it can be *solved* (executed step by step).

5.8.6 Step-Conditioned Retrieval Dynamics

The rule-level analysis above examines *which* rules are retrieved; we now examine *when* they are retrieved. If InsightEmb truly understands procedural structure, its retrieval should *change* as the task progresses: early steps should retrieve search rules (the agent hasn't found the object yet), while later steps should shift toward state-transformation or verification rules.

We extract the retrieved insight from every step of all 140 in-distribution games (dynamic retrieval with ATOMIC corpus), classify each insight using GPT-5.2 into the same 8 situation labels, and aggregate by task phase. Table 14 shows the results (see Appendix F for per-step visualizations).

Two key patterns emerge. First, **InsightEmb shows phase-aware retrieval**: at steps 0–4, it retrieves Search & Locate rules **65.6%** of the time;

as the task progresses, Verification rules rise from 17.4% to 32.9% and State Transform from 2.6% to 7.5%. In contrast, **Base retrieval is phase-invariant**: Search & Locate varies only from 18.5% to 27.5%, and State Transform stays between 20.5% and 25.2% across all phases.

Search-dominant retrieval improves all task types. A natural concern is that InsightEmb's heavy bias toward Search & Locate rules (51–66%) might only help search-heavy tasks. However, per-task-type analysis reveals the opposite: InsightEmb's largest gains over Base are on *state-transformation* tasks—clean (+22.7%, from 36.4% to 59.1%), cool (+15.3%, from 46.2% to 61.5%), and put (+13.0%, from 67.4% to 80.4%)—precisely the tasks where Base over-retrieves topically matched but procedurally premature rules (e.g., heating rules before the object is found). This confirms that Search & Locate rules are *universally useful*: every ALFWorld task begins with locating the target object, and retrieving the right search strategy at the right time is the single most impactful procedural decision.

5.8.7 Synthetic Procedural Probes

The step-conditioned analysis above shows that InsightEmb's retrieval changes over time, but the agent's observation also changes—so the shift could reflect surface-level observation changes rather than genuine procedural understanding. To isolate procedural sensitivity, we construct *synthetic probe pairs*: topically identical queries that differ only in procedural phase (SEARCH: object not yet found; ACTION: object in hand). A surface-matching retriever should return similar results for both; a procedurally-aware retriever should return different rule types.

We test 4 probe pairs (clean, cool, examine, put), retrieving the top-1 insight from the ATOMIC corpus and classifying it with GPT-5.2 (see Appendix G for full details). InsightEmb achieves **75%** accuracy (6/8) vs. Base's 50% (4/8), with **perfect SEARCH-phase accuracy** (4/4 vs. 3/4). The clearest example is "examine book": InsightEmb retrieves SEARCH & LOCATE for the SEARCH query and NAVIGATION for the ACTION query, while Base retrieves SEARCH & LOCATE for *both*—failing to recognize that the bottleneck has shifted from finding to navigating. Both models struggle with ACTION-phase probes (Base: 1/4, InsightEmb: 2/4), suggesting that fine-grained procedural distinctions remain challenging while coarse phase distinctions (Search vs. non-Search) are well captured.

6 Discussion

Why does math outperform in-domain data? The in-domain ALFWorld retriever (57.86%) underperforms InsightEmb (65.00%) despite being trained on task-relevant data. Two factors explain this gap. First, *data diversity*: math training covers three structurally distinct reasoning domains, each with its own abstraction patterns, while ALFWorld training data is limited to a single environment with repetitive task structures. Second, *abstraction range*: mathematical insights span from concrete formulas to high-level meta-strategies, providing a richer gradient of abstraction levels for the contrastive loss to exploit. These factors suggest that the quality of contrastive training for insight retrieval depends more on the *structural diversity* of the training domain than on its topical proximity to the target domain.

Implications for agentic RAG. Our results suggest a practical paradigm for retrieval in data-scarce agentic settings: rather than collecting expensive in-domain retrieval training data, practitioners can train on data-rich reasoning domains (such as math) and transfer the learned structural matching capability. The key requirement is that the training data exhibits the same *situation* $\rightarrow$ *abstract rule* structure as the target domain. This is broadly satisfied by any domain where concrete problems are solved by applying abstract principles.

Limitations and future work. Our evaluation covers two target environments (ALFWorld and WebShop) with two action-generating LLMs (Qwen3-8B and GPT-5.2) and multiple retrieval budgets ($k \in \{1, 3, 5, 7\}$). Validating the approach on additional environments (e.g., ScienceWorld) and with a wider range of LLMs would further strengthen the generality claim. We also lack direct retrieval quality metrics (e.g., Recall@k); the current evaluation measures only downstream task success, which conflates retrieval quality with LLM action-generation quality. Finally, we have not explored how the choice of math sub-domains or training data scale affects transfer, nor whether the approach extends to more complex multi-step reasoning tasks with larger insight corpora. Additionally, while our synthetic procedural probes (§5.8.7)

demonstrate phase-sensitive retrieval, the ACTION-phase accuracy (2/4) suggests room for improvement in fine-grained procedural distinctions.

7 Conclusion

We introduced InsightEmb, a contrastive training framework that learns to retrieve abstract reasoning insights for LLM agents by training exclusively on mathematical reasoning data. Our key finding is that insight retrieval is a structural matching problem: the ability to connect concrete situations to abstract guiding rules transfers across domains without any in-domain training data. InsightEmb improves ALFWorld success rates by +10.0% over the base embedding model and +7.1% over an in-domain–trained retriever, and further demonstrates consistent gains on WebShop (82% relative improvement in average task score over the base model), confirming that structurally diverse out-of-domain data can be more valuable than scarce in-domain data for training agentic retrieval models. These results point toward a general principle: when the retrieval task requires bridging an abstraction gap, training on any domain that exhibits the same situation-to-rule structure can yield effective cross-domain transfer.

Acknowledgments

[To be added.]

References

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: Retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*.

Bodhisattwa Prasad Majumder, Bhavana Dalvi Mishra, Peter Jansen, Oyvind Tafjord, Niket Tandon, Li Zhang, Chris Callison-Burch, and Peter Clark. 2023. CLIN: A continually learning language agent for rapid task adaptation and generalization. *arXiv preprint arXiv:2310.10134*.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, and 1 others. 2024. ToolLLM: Facilitating large language models to master 16000+ real-world APIs. In *Proceedings of the Twelfth International Conference on Learning Representations*.

Qwen Team. 2025. Qwen3 technical report. *arXiv preprint*.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2021. ALFWorld: Aligning text and embodied environments for interactive learning. In *Proceedings of the Ninth International Conference on Learning Representations*.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2023. One embedder, any task: Instruction-finetuned text embeddings. In *Findings of the Association for Computational Linguistics: ACL 2023*.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2024. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and

Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *Proceedings of the Ninth International Conference on Learning Representations*.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022. WebShop: Towards scalable real-world web interaction with grounded language agents. In *Advances in Neural Information Processing Systems*.

A SRA-Bench Per-Task Retrieval Details

Table 15 reports per-task SRA-Bench retrieval results at @1 and @10. The macro-average in Table 3 is computed by averaging these task-family results equally, regardless of the number of examples in each task family.

B Training Data Statistics

C Qualitative Case Studies

We present two additional examples where InsightEmb succeeds while both baselines fail.

Game 4: "put a clean soapbar in cabinet." All three models check sinkbasin and bathtubbasin (both empty). The base model then goes to `countertop 1` (generic guess); the in-domain model does the same. InsightEmb retrieves the insight "Adhere to Spatial Logic Chains—toiletries in bathrooms" and checks `garbagecan 1`, where the soapbar is found. InsightEmb completes the task in **9 steps**; both baselines timeout at 50.

Game 129: "put a cool pan in stoveburner." The base model searches cabinets sequentially (cabinet 1, 2, 3...). InsightEmb retrieves "Prioritize Task-Relevant Locations First—check destinations mentioned in the task" and goes directly to `stoveburner 1`, where the pan already is. InsightEmb completes in **29 steps**; both baselines timeout. This illustrates the counterintuitive but effective heuristic of checking the task's *destination* before searching storage—a strategy that InsightEmb's structurally-trained retrieval surfaces.

D Embedding Space Visualization

Figure 2 shows t-SNE projections of 500 randomly sampled individual rules from the ATOMIC corpus (5,251 single-rule insights), colored by GPT-classified situation labels. The Base model produces better-separated topical clusters, while InsightEmb compresses the space. Table 18 quantifies this: Base achieves higher Silhouette scores and nearest-neighbor label purity, yet InsightEmb achieves higher downstream success. This confirms that InsightEmb's advantage lies in *query-conditioned* retrieval rather than insight-insight clustering.

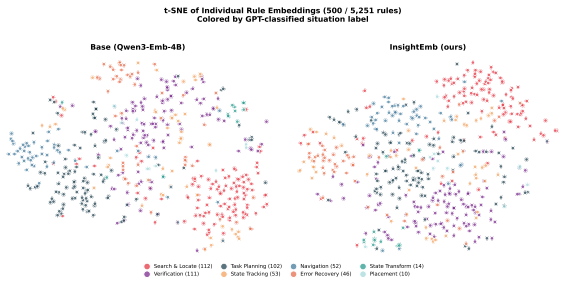

Figure 2: t-SNE of individual rule embeddings (ATOMIC corpus, 500 / 5,251 rules). Colors indicate GPT-classified situation labels. Base (left) shows better topical separation; InsightEmb (right) compresses the space but retrieves more procedurally relevant rules at inference time.

E Hyperparameters

F Step-Conditioned Retrieval Details

Figure 3 shows the grouped bar chart of retrieved insight category distributions across task phases for Base and InsightEmb. Figure 4 shows the per-step stacked area chart (smoothed with a 3-step rolling average).

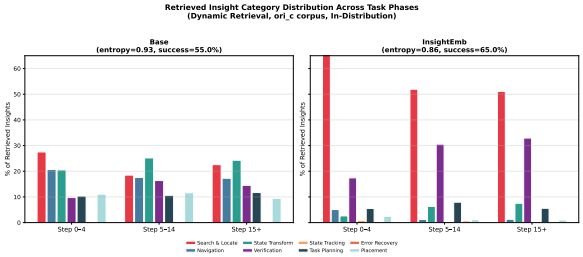

Figure 3: Retrieved insight category distribution across task phases (Step 0–4, 5–14, 15+). InsightEmb (right) shows clear phase-dependent retrieval: Search & Locate dominates early steps, then Verification rises as the agent progresses. Base (left) shows a roughly uniform distribution.

G Synthetic Procedural Probe Details

To isolate procedural sensitivity from surface-level observation changes, we construct *synthetic probe pairs*: topically identical queries that differ only in procedural phase. Each probe pair consists of two queries about the same task:

Task family	n	Base				InsightEmb			
		R@1	N@1	R@10	N@10	R@1	N@1	R@10	N@10
TheoremQA	747	48.19	48.19	74.43	60.63	**58.10**	**58.10**	**86.61**	**72.37**
LogicBench	760	1.58	1.58	**40.39**	19.53	**6.32**	**6.32**	39.34	**20.18**
ToolQA	1,430	16.22	16.22	44.83	28.84	**29.65**	**29.65**	**62.45**	**45.69**
CHAMP	223	13.86	19.73	38.34	26.97	**19.06**	**29.15**	**52.58**	**38.48**
MedCalcBench	1,100	**92.73**	**92.73**	**98.18**	**95.80**	63.27	63.27	**98.18**	81.52
BigCodeBench	1,140	9.42	24.47	28.62	24.09	**11.85**	**31.23**	**39.04**	**32.27**

Table 15: Per-task SRA-Bench retrieval results at @1 and @10. Queries use task information and candidates use full skill content. Bold marks the better value within each task family; ties are bolded for both embeddings.

Stage	Total	Query	Traj
Stage 1	4,943	∼1,650	∼3,293
Stage 2	2,896	∼2,329	∼567

Table 16: Training data statistics. "Query" = query-only samples, "Traj" = full + partial trajectory samples.

Math Domain	Stage 1	Stage 2
Counting & Probability	1,650	966
Number Theory	1,650	966
Geometry	1,643	964
Total	4,943	2,896

Table 17: Task distribution in training data.

Metric	Base	InsightEmb
Silhouette (cosine)	**0.053**	0.036
NN Purity (k=5)	**0.653**	0.592
NN Purity (k=10)	**0.615**	0.570
Intra-class sim	0.574	0.823
Inter-class sim	0.497	0.801
Sim gap (intra−inter)	**0.077**	0.022
Downstream success	54.3%	**67.1%**

Table 18: Clustering quality (500 rules, GPT-classified labels, cosine metric on raw 2,560-dim embeddings) vs. downstream task success. Base clusters insights by topic better; InsightEmb retrieves more useful insights for the agent.

Hyperparameter	Stage 1	Stage 2
Base model	Qwen3-Emb-4B	Stage 1 ckpt
Learning rate	1×10^{-5}	1×10^{-5}
LR scheduler	Cosine	Cosine
Warmup ratio	0.1	0.1
Batch size (per device)	8	8
Grad. accumulation	2	2
Training epochs	6	6
Temperature τ	0.01	0.01
Training group size	11	11
Max passage length	1,536	1,536
Precision	BF16	BF16
GPUs	8	8
DeepSpeed	ZeRO-3	ZeRO-3
Seed	3407	3407

Table 19: Training hyperparameters for both stages.

- **Query A (SEARCH phase)**: The agent has just started and needs to find the target object. Expected retrieval: SEARCH & LOCATE.
- **Query B (ACTION phase)**: The agent already holds the object and needs to act on it. Expected retrieval depends on the task type (e.g., STATE TRANSFORM, NAVIGATION, or PLACEMENT).

Both queries mention the same task and objects, so a surface-matching retriever should return similar results for both. A procedurally-aware retriever should return different rule types. We retrieve the top-1 insight from the ATOMIC corpus using each embedding model and classify it with GPT-5.2.

G.1 Probe Definitions

We define 4 probe pairs covering different ALFWorld task types. Each probe simulates a realistic agent observation at two different procedural phases.

Probe 1: Clean a soapbar. **Task**: "put a clean soapbar in cabinet."

- **Query A (SEARCH)**: Agent sees bathtubbasin, cabinets, countertop, drawer, garbagecan, sinkbasin, toilet. Object not yet found. *Expected: Search & Locate.*
- **Query B (ACTION)**: Agent has picked up soapbar 1 from countertop 1. Available actions include going to sinkbasin or cabinets. *Expected: State Transform* (need to clean at sinkbasin).

Probe 2: Cool a pan. **Task**: "put a cool pan in stoveburner."

- **Query A (SEARCH)**: Agent sees cabinets, countertop, fridge, microwave, sinkbasin, stoveburners. Object not yet found. *Expected: Search & Locate.*
- **Query B (ACTION)**: Agent has picked up pan 1

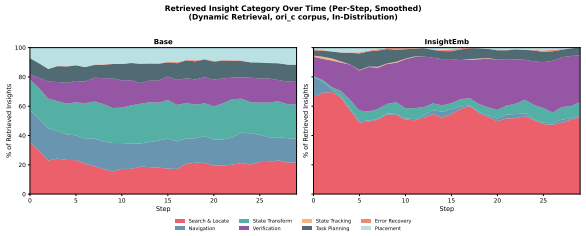

Figure 4: Per-step retrieved insight category distribution (smoothed). InsightEmb shows a clear Search → Verification transition, while Base remains roughly uniform throughout.

from stoveburner 1. Available actions include going to fridge. *Expected: State Transform* (need to cool in fridge).

Probe 3: Examine a book under desklamp. **Task**: "examine the book with the desklamp."

- **Query A (SEARCH)**: Agent sees bed, desk, drawers, garbagecan, shelves, sidetable. Object not yet found. *Expected: Search & Locate.*
- **Query B (ACTION)**: Agent has picked up book 1 from shelf 1. Available actions include going to desk or sidetable. *Expected: Navigation* (need to go to desklamp location).

Probe 4: Put a butterknife in drawer. **Task**: "put a clean butterknife in drawer."

- **Query A (SEARCH)**: Agent sees cabinet, countertop, drawers, fridge, microwave, sinkbasin, stoveburner. Object not yet found. *Expected: Search & Locate.*
- **Query B (ACTION)**: Agent has cleaned butterknife 1 using sinkbasin 1. Available actions include going to drawers. *Expected: Placement* (need to place in drawer).

G.2 Results Summary

Table 20 shows the top-1 retrieval results for all 8 queries (4 probes × 2 phases).

G.3 Detailed Retrieved Insights

Below we list the top-1 retrieved insight for each query under both models, including the classified label and cosine similarity score.

Probe 1: Clean a soapbar.

- **Query A (SEARCH)** — Expected: SEARCH & LOCATE
  - **Base** [✓]: "Container Proximity Hierarchy: When storing objects, first navigate to the nearest valid container type specified in the task (drawer/cabinet) before considering distant alternatives." (SEARCH & LOCATE, sim=0.760)
  - **InsightEmb** [✓]: "Maintain Task-Focused Exploration: Always prioritize locations directly relevant to the target object (e.g., sinkbasins and countertops for dishsponges) before considering other areas." (SEARCH & LOCATE, sim=0.690)
- **Query B (ACTION)** — Expected: STATE TRANSFORM
  - **Base** [✓]: "Apply Contextual Cleaning Protocol: When 'clean' is specified in the task, automatically prioritize sinkbasin interaction after obtaining the target object, before any placement attempts." (STATE TRANSFORM, sim=0.781)
  - **InsightEmb** [✓]: "Locate object → Clean at sinkbasin (if dirty) → Re-examine cleanliness → Place" (STATE TRANSFORM, sim=0.675)

Probe 2: Cool a pan.

- **Query A (SEARCH)** — Expected: SEARCH & LOCATE
  - **Base** [×]: "When heating is required, first locate then navigate directly to the nearest appropriate appliance (microwave/stove) after obtaining the target object" (STATE TRANSFORM, sim=0.805). *Base performs topical matching: "cool a pan" → thermal/heating rules.*

Task	Phase	Expected	Base	Ours
Clean soapbar	Search	Search	✓	✓
	Action	Transform	✓	✓
Cool pan	Search	Search	×	✓
	Action	Transform	×	×†
Examine book	Search	Search	✓	✓
	Action	Navigation	×	✓
Put butterknife	Search	Search	✓	✓
	Action	Placement	×	×‡
Accuracy			4/8	**6/8**
SEARCH phase			3/4	**4/4**
ACTION phase			1/4	**2/4**

✓ = matches expected. †Retrieved Verification. ‡Retrieved Search & Locate.

Table 20: Full synthetic procedural probe results. InsightEmb achieves 75% accuracy (6/8) vs. Base's 50% (4/8), with perfect SEARCH-phase accuracy (4/4).

  - **InsightEmb** [✓]: "Prioritize Object Proximity First: Check locations most contextually relevant to the task (sink for cleaning, countertops/stove for pans) before exhaustive cabinet searches." (SEARCH & LOCATE, sim=0.677)

- **Query B (ACTION)** — Expected: STATE TRANSFORM
  - **Base** [×]: "Verify Inventory Before Placement: Explicitly confirm possession of the target object using the 'inventory' command before attempting to place it on a target appliance like a stoveburner." (VERIFICATION, sim=0.799)
  - **InsightEmb** [×†]: "Ensure Cooling Completion Before Placement: Verify the cooling action (e.g., fridge interaction) is fully executed and acknowledged by the environment before attempting to move the object to its final destination." (VERIFICATION, sim=0.702). *A reasonable near-miss: verification of cooling is closely related to the cooling procedure itself.*

Probe 3: Examine a book under desklamp.

- **Query A (SEARCH)** — Expected: SEARCH & LOCATE
  - **Base** [✓]: "Strategic Location Prioritization: Always prioritize movement towards locations mentioned in the task objective (e.g., 'desklamp' implies proximity to desk/sidetable) before exploring unrelated areas." (SEARCH & LOCATE, sim=0.798)
  - **InsightEmb** [✓]: Same insight retrieved. (SEARCH & LOCATE, sim=0.668)

- **Query B (ACTION)** — Expected: NAVIGATION
  - **Base** [×]: Same insight as Query A: "Strategic Location Prioritization…" (SEARCH & LOCATE, sim=0.740). *Base retrieves the same rule for both phases—it does not recognize the procedural shift.*
  - **InsightEmb** [✓]: "Prioritize Proximity First: Always navigate directly to the most relevant object/location mentioned in the goal before interacting with secondary objects (e.g., go to the desk before focusing on the desklamp)." (NAVIGATION, sim=0.692). *InsightEmb correctly switches from Search to Navigation.*

Probe 4: Put a butterknife in drawer.

- **Query A (SEARCH)** — Expected: SEARCH & LOCATE
  - **Base** [✓]: "Container Proximity Hierarchy: When storing objects, first navigate to the nearest valid container type specified in the task (drawer/cabinet) before considering distant alternatives." (SEARCH & LOCATE, sim=0.782)
  - **InsightEmb** [✓]: "Prioritize Direct Object Source Checks First: Immediately inspect surfaces like countertops and tables before cabinets/drawers when searching for items." (SEARCH & LOCATE, sim=0.677)

- **Query B (ACTION)** — Expected: PLACEMENT
  - **Base** [×]: "Verify Object Placement: Before finalizing placement in drawers, confirm both target items are in inventory and ensure drawer is open through explicit interaction." (VERIFICATION, sim=0.781)
  - **InsightEmb** [×‡]: "Maintain Task-Focused Exploration: Always prioritize locations directly relevant to the target object before considering other areas." (SEARCH & LOCATE, sim=0.672)

G.4 Key Observations

InsightEmb achieves higher probe accuracy. InsightEmb correctly retrieves the expected rule category in 6/8 cases (75%) vs. Base's 4/8 (50%). The gap is most pronounced in the SEARCH phase: InsightEmb achieves **4/4** (perfect), while Base achieves 3/4. Base fails on "cool a pan" by retrieving a State Transform rule about heating procedures instead of a Search rule, demonstrating topical matching (cooling → thermal rules) rather than procedural matching.

InsightEmb shows phase-sensitive retrieval. The "examine book" probe is the clearest example: InsightEmb retrieves SEARCH & LOCATE for Query A (need to find the book) and NAVIGATION for Query B (have the book, need to go to the desklamp)—correctly adapting to the procedural phase. Base retrieves SEARCH & LOCATE for *both* queries, failing to recognize that the agent's bottleneck has shifted from finding to navigating.

ACTION phase remains challenging. Both models struggle with ACTION-phase probes (Base:

1/4, InsightEmb: 2/4). For "cool a pan," InsightEmb retrieves VERIFICATION ("Ensure Cooling Completion Before Placement") instead of STATE TRANSFORM—a reasonable but not exact match, since verification of cooling is closely related to the cooling procedure itself. This suggests that fine-grained procedural distinctions (Transform vs. Verify) remain difficult, while coarse phase distinctions (Search vs. non-Search) are well captured.

H WebShop Qualitative Examples

We present illustrative examples from the WebShop evaluation where InsightEmb (Stage 1+2) with BUNDLE insights succeeds while Base fails, or achieves substantially higher task scores. These examples demonstrate the three qualitative patterns identified in §5.5: variant selection awareness, loop prevention, and procedural sequencing.

H.1 Example 1: Office Chair (Game 82)

Task: *"Find me height adjustable, high density, easy install, easy assemble home office chairs for living room with color: type 2-grey fabric, and price lower than 120.00 dollars."*

InsightEmb: WON in 6 steps (score = 1.0). **Base**: LOST in 21 steps (score = 0.857).

InsightEmb insight (rules):

> 1. *Precision Filtering* — Always include exact measurements, pack sizes, and price ceilings in search queries.
> 5. *Installation Requirement Check* — For products requiring assembly, confirm "easy installation" claims through description keywords.
> 7. *Color Match Verification* — Use exact color terminology from the task description and confirm against product descriptions, accounting for potential variations like "grey-grey" vs "charcoal."

Base insight (rules):

> 1. *Precision Search Formulation* — Always include all critical attributes in search queries using natural language syntax.
> 6. *Multi-Page Exploration* — Always click next to review subsequent search result pages when first-page results don't fully match requirements.

Action comparison.

- **InsightEmb**: search → click product (`b08p8lrfz4`) → check features → go back → **select "type 2-grey fabric"** → buy now ✓
- **Base**: search → click next → click next → inspect wrong product (`b08fqtmcnm`) → back to search → re-search → click next → inspect same wrong product again → … (21 steps, never selects color variant)

Analysis. InsightEmb's "Color Match Verification" rule explicitly instructs the agent to select the exact color variant before purchasing. Base's generic "Precision Search Formulation" does not emphasize variant selection, causing the agent to skip this critical step and purchase without selecting the correct color, resulting in a partial score.

H.2 Example 2: Men's Outdoor Shoes (Game 274)

Task: *"Find me non slip men's outdoor shoes with rubber sole, comfortable fit, unique design with size: 13.5, and price lower than 140.00 dollars."*

InsightEmb: WON in 6 steps (score = 1.0). **Base**: LOST in 5 steps (score = 0.833).

Action comparison. Both models find the same product (`b09hh6p68w`), but:

- **InsightEmb**: click product → check features → go back → **select size 13.5** → buy now ✓
- **Base**: click back to search → re-search → click same product → **buy now without selecting size** ×

Analysis. InsightEmb's insight includes a "Specification Cross-Check" rule: *"When product descriptions partially match requirements, check secondary attributes before committing to purchase."* This guides the agent to verify and select the size variant before purchasing. Base's insight mentions "Attribute Cross-Verification" but the agent skips the size selection step in practice.

H.3 Example 3: Women's Sweater (Game 368)

Task: *"Find me women's relaxed fit long sleeve sweaters with color: lilac, and size: x-large, and price lower than 60.00 dollars."*

InsightEmb: WON in 5 steps (score = 1.0). **Base**: LOST in 3 steps (score = 0.600).

Action comparison. Both find the same product (`b07dkgjr74`), but:

- **InsightEmb**: search → click product → **select lilac** → **select x-large** → buy now ✓
- **Base**: search → click product → **buy now immediately** (without selecting color or size) ×

Analysis. This is the clearest example of the *variant selection awareness* pattern. InsightEmb's insight includes a "Buy Decision Threshold" rule: *"Initiate purchase only when a product matches ALL stated requirements simultaneously, avoiding partial matches."* This prevents premature purchasing.

Base's insight lacks this procedural constraint, and the agent buys immediately without selecting the required color and size variants.

H.4 Example 4: Men's Loafers (Game 206)

Task: *"Find me men's loafers & slip-ons with rubber outsole, rubber sole with color: blue, and size: 10.5, and price lower than 70.00 dollars."*

InsightEmb: WON in 18 steps (score = 1.0). **Base**: LOST in 50 steps (score = 0.000).

This is the most dramatic example ($\Delta = +1.000$).

InsightEmb insight (key rules):

> 7. *Size Interpretation Rule* — For clothing/apparel, include both numerical sizes and categorical descriptors in search queries to account for variant labeling.
> 10. *Session Management Rule* — After 3 unsuccessful search refinement cycles, reset by returning to initial search and testing alternative phrasing rather than continuing deep navigation.

Analysis. InsightEmb's "Session Management Rule" helps the agent break out of unproductive search loops and eventually find the correct product. Base gets stuck in an infinite loop of searching and browsing for all 50 steps, never finding a matching product, scoring zero. This illustrates the *loop prevention* pattern: InsightEmb's structurally-trained retrieval surfaces error-recovery strategies that prevent the agent from exhausting its step budget on fruitless exploration.

H.5 Example 5: Men's Shorts (Game 183, Atomic)

Task: *"Find me officially licensed, machine wash men's shorts with drawstring closure, elastic waistband with color: heather grey, and size: large, and price lower than 70.00 dollars."*

InsightEmb: WON in 5 steps (score = 1.0). **Base**: LOST in 28 steps (score = 0.857).

Action comparison.

- **InsightEmb**: search → click product → **select heather grey** → **select large** → buy now ✓ (5 steps)
- **Base**: search → click product → back to search → re-search → click same product → back to search → re-search → back to search → … (28 steps, never selects variants)

Analysis. InsightEmb's atomic insight (*"Precisely encode key attributes in search queries"*) is concise and actionable. Base's atomic insight is a generic rule template that does not guide the agent through the variant selection workflow, causing it to loop between searching and product pages without ever selecting the required color and size.

H.6 Summary of WebShop Qualitative Patterns

Across all divergent games, three consistent patterns emerge:

Metric	InsightEmb	Base
Wins (other loses)	**8**	1
Zero-score games	**247**	361
Perfect-score games	**12**	5
Avg task score	**0.328**	0.181

Table 21: Head-to-head comparison between InsightEmb (Stage 1+2) and Base with BUNDLE insights on 500 WebShop test games.

1. **Variant selection awareness** (Games 82, 274, 368, 183): InsightEmb consistently guides the agent to select product variants (color, size) before purchasing. Base frequently skips this step, resulting in partial scores (0.600–0.857) instead of perfect scores. This is the single most impactful behavioral difference.
2. **Loop prevention** (Games 206, 279): Base gets stuck in search–browse–back loops for 21–50 steps. InsightEmb's insights about session management and error recovery help the agent break out of unproductive cycles.
3. **Procedural sequencing**: InsightEmb retrieves insights that encode a sequential workflow (search → verify → select variants → buy), while Base retrieves topically relevant but procedurally vague insights. This mirrors the ALFWorld finding where InsightEmb performs *procedural matching* rather than *topical matching*.