Auditing Generated Skill Libraries for Embodied Language Agents

Anonymous ACL submission

Abstract

Language agents accumulate natural-language skill libraries over hundreds of episodes, yet no one asks what is *inside* these libraries—which skills help, which are overlapping, and which remain reliable under distribution shift. We introduce **SkillAudit**, an offline audit framework treating generated skill libraries as first-class objects of analysis. The core mechanism decomposes each skill into semantic sections (identity, procedure, learned failure guards, and history), embeds them independently, and feeds them into a utility ensemble without requiring additional environment interactions. Our central finding is that *representation is the bottleneck, not prediction*: while whole-skill embeddings produce an inverted ranking, our section-aware representations achieve highly accurate skill evaluations (Spearman 0.899). Specifically, learned failure constraints (guards) are the most decisive section. Furthermore, redundancy analysis enables conservative pruning of 10.9% of skills without downstream performance loss. Finally, we reveal that skill utility is distribution-sensitive rather than intrinsic, as in-distribution predictors fail to reliably rank out-of-distribution skills. The implication is clear: generated skill libraries must be *audited, not just accumulated*.

1 Introduction

Language agents are accumulating procedural knowledge faster than anyone is checking whether it works. In embodied environments like ALFWorld (Shridhar et al., 2021), agents reflect on successes and failures to induce natural-language *skills*—reusable procedures such as "search cabinets for an object" or "cool an item before placing it" (Wang et al., 2023a; Qian et al., 2024). Over hundreds of episodes, these skills grow into a *library* retrieved at inference time. The paradigm has driven rapid progress: Voyager (Wang et al., 2023a) demonstrated open-ended accumulation in Minecraft, SkillRL (Xia et al., 2026a) combined libraries with reinforcement learning, and SkillRouter (Zheng et al., 2026) introduced quality-weighted retrieval at scale. Yet a fundamental question remains unexamined: *what happens inside these libraries after hundreds of episodes of iterative induction?*

Two camps of prior work address library quality, but both look in the wrong place. The first builds *better libraries*—improved induction (Qian et al., 2024), graph-structured compositions (Xia et al., 2026b), RL-augmented generation (Xia et al., 2026a)—yet moves on the moment the library is generated, never inspecting what was produced. The second builds *better inference*—skill routing (Zheng et al., 2026) and goal-grounded selection (Kim et al., 2025)—yet inference can only be as good as the library it draws from. Neither asks whether the library itself is bloated, redundant, or distribution-brittle. Opening a real 55-skill library reveals a stark picture: *29 skills (52.7%) have never been retrieved*; several high-similarity clusters share near-identical procedures, differing mostly in a final receptacle or examination target. A predictor trained in-distribution *inverts* its rankings out-of-distribution. These pathologies are invisible to success-rate evaluation, which measures the library's *output* but reveals nothing about its *internal structure*.

The fix is surprisingly simple: each pathology leaves a distinct signature once a skill is decomposed into its semantic pieces. We introduce **SkillAudit**, an offline audit framework that treats generated skill libraries as first-class objects of analysis (Figure 1). It parses each skill into semantic sections—identity, procedure, guards (learned failure constraints), and history—embeds them separately, and routes the resulting representations through a small-sample utility ensemble without additional environment interactions. The critical finding is that naïve whole-skill embeddings *invert*

quality rankings, while section-aware input recovers reliable signal—driving our decomposition-first design. Our contributions:

1. **Representation, not prediction, is the bottleneck.** The main all-section audit model achieves LOO Spearman 0.866; in a representation ablation, the best semantic-only section set (identity + procedure + guards) reaches 0.899, while whole-skill embeddings yield an *inverted* ranking (-0.938). The critical section is guards (learned failure constraints), whose addition alone produces a +0.610 Spearman gain (Table 2).

2. **Generated libraries contain actionable redundancy.** Pairwise section similarity, task-type agreement, and graph-based clustering identify seven overlap clusters at threshold 0.94; conservative pruning at threshold 0.96 achieves 100% duplicate-or-overlap precision. Pruning 10.9% of skills preserves downstream ALFWorld performance within paired confidence intervals (Table 5).

3. **Skill quality is distribution-bound, not intrinsic.** An ID-trained predictor does not reliably rank OOD skill utility (solve-rate Spearman as low as -0.43), revealing that skill utility depends on the task distribution—motivating distribution-aware auditing over universal scoring.

2 Related Work

Skill Discovery and Composition in LLM Agents. Existing work has focused on how to *build* skill libraries; we focus on what to do *after* they are built. Accumulating reusable procedures from experience traces back to temporally extended actions (Sutton et al., 1999); Barto and Mahadevan (2003) surveyed hierarchical RL abstractions. LLM-based skill libraries instantiate this paradigm in natural language. Voyager (Wang et al., 2023a) pioneered code-based skill accumulation, DEPS (Wang et al., 2023b) introduced interactive planning with error correction, and Reflexion (Shinn et al., 2023) and ExpeL (Zhao et al., 2024) accumulated textual lessons through reflection. For ALFWorld, Qian et al. (2024) studied iterative skill induction, SkillRL (Xia et al., 2026a) combined libraries with RL, and SkillRouter (Zheng et al., 2026) introduced quality-weighted retrieval. All focus on *building* better libraries; none systematically audit what they produce. Our framework fills this gap: just as option discovery must prune redundant options, generated skill libraries require analogous curation.

Knowledge Base Quality and Maintenance. Skill libraries are knowledge bases, and their maintenance is well studied: Paulheim (2017) surveyed knowledge graph refinement, covering error detection, completion, and cleanup. For retrieval-augmented generation, Izacard et al. (2023) studied document deduplication and its impact on retrieval quality, and Izacard and Grave (2021) showed that corpus size interacts with retrieval quality in non-trivial ways. The LIMA principle (Zhou et al., 2023)—that data quality matters more than quantity—is echoed in our finding that a leaner, curated library can match a larger one and shows a positive OOD trend. We apply these ideas to skill libraries, a setting where entries are procedural rather than declarative and quality depends on downstream agent performance.

Structured Text Representation. Sentence-BERT (Reimers and Gurevych, 2019) showed that embedding sub-documents independently yields better similarity estimates than single-vector representations. This approach has been applied to recipe understanding (Majumder et al., 2019) and instruction following (Mishra et al., 2022). We adapt this principle to generated skills, but with a twist: the decisive section turns out to be guards—learned failure constraints—not the positive instructions that prior work would prioritize. This echoes contrastive learning, where hard negatives are often more informative than positive examples (Robinson et al., 2021).

Library Pruning and Compression. Pruning redundant entries from knowledge bases is well studied: knowledge graph refinement surveys cover deduplication and cleanup (Paulheim, 2017), while embedding-based methods (Bordes et al., 2013) enable scalable similarity detection. More recently, retrieval-augmented generation systems have studied the trade-off between corpus size and retrieval quality (Petroni et al., 2021). Our conservative pruning approach—removing only high-confidence overlaps and verifying downstream preservation—draws on the central lesson of neural network compression (Frankle and Carbin, 2019): targeted compression can preserve behavior when the removed components are redundant.

3 Method

Problem Setting. We consider ALFWorld (Shridhar et al., 2021), an embodied text-based environment where skills—Markdown files with INSTRUCTIONS and ANTI-PATTERNS sections, induced via reflection loops (Wang et al., 2023a; Qian et al., 2024)—are retrieved at inference and injected into the agent's prompt. After hundreds of episodes, a library accumulates dozens of skills with varying quality and significant overlap. We ask: *Can we characterize, score, and compress such a library offline, without a single additional rollout?*

Overview. SkillAudit has two analysis heads (Figure 1). The *quality head* decomposes each skill into semantic sections, embeds them with importance-weighted representations, and blends predictions with stabilized telemetry to produce risk-adjusted scores (§3.1–§3.4). The *redundancy head* computes pairwise section similarity, clusters overlapping skills, and prunes low-value duplicates (§3.5). A raw skill folder goes in; a ranked, deduplicated library comes out.

3.1 Skill Parsing and Section Decomposition

Not all parts of a skill contribute equally to its quality—actionable instructions matter far more than boilerplate metadata. Embedding each skill as a single text blob confirms the problem: it produces a severe anti-correlation (see §4), because low-quality skills tend to be longer, so holistic embeddings capture text volume rather than procedural precision. To isolate the signal that matters, SkillAudit decomposes each skill into six semantic sections and embeds them independently:

- **Identity**: skill name, description, task types, and axis (e.g., Planning, Execution, Termination).
- **Procedure**: the actionable instructions—step-by-step directions the agent follows.
- **Guards**: anti-patterns and learned failure constraints (e.g., "DO NOT exhaustively search one location").
- **History**: patch diagnoses and version-derived lessons from iterative refinement.
- **Full**: the complete skill text as a fallback representation.
- **Version delta**: changes across skill versions, capturing the refinement trajectory.

Each section is embedded independently using Qwen3-Embedding-4B (Qwen Team, 2024), producing 2560-dimensional L2-normalized vectors. Because different sections carry different predictive value, we form the composite skill representation as a weighted concatenation:

$$\mathbf{e}_s = \bigoplus_{c \in \mathcal{C}} w_c \cdot \mathbf{v}_{s,c} \tag{1}$$

where $\mathcal{C}$ is the set of sections, $\mathbf{v}_{s,c}$ is the normalized embedding of section c for skill s, and w_c are section weights. We weight PROCEDURE highest ($w = 1.25$) and FULL lowest ($w = 0.55$): actionable instructions should dominate, and the full-text fallback should not wash out structured signal.

3.2 Non-Outcome Metadata Features

Embeddings capture what a skill *says*, but not how the system has *treated* it—how often it has been retrieved, how many times it has been revised, or whether it even has guardrails. These signals are informative because they reflect the retriever's implicit judgment aggregated over many episodes, yet remain independent of any individual rollout outcome and thus introduce no label leakage. SkillAudit encodes them as eight metadata features:

$$\mathbf{m}_s = [\log u_s,\, \log v_s,\, p_s,\, |\tau_s|,\, \mathbb{1}_{g_s},\, \log |I_s|,\, \log |G_s|,\, \log |H_s|] \tag{2}$$

where u_s is usage count, v_s is version number, p_s is patch count, τ_s is the set of declared task types, g_s indicates presence of guard sections, and $|I_s|, |G_s|, |H_s|$ are character counts of instructions, guards, and history, respectively. Count-valued features are log-transformed to compress their heavy-tailed distributions. Intuitively, a skill with many patches has been actively refined, while a skill with zero usage has never been selected by the retriever.

3.3 Empirical-Bayes Telemetry Target

Features describe what a skill *is*; we still need a label for how *good* it is. Raw solve rate is inadequate: a skill used 3 times with 1 success yields the same rate (0.333) as one used 30 times with 10 successes, yet our confidence in these two estimates differs by an order of magnitude. To stabilize this signal, SkillAudit applies an empirical-Bayes posterior that shrinks noisy estimates toward the library mean:

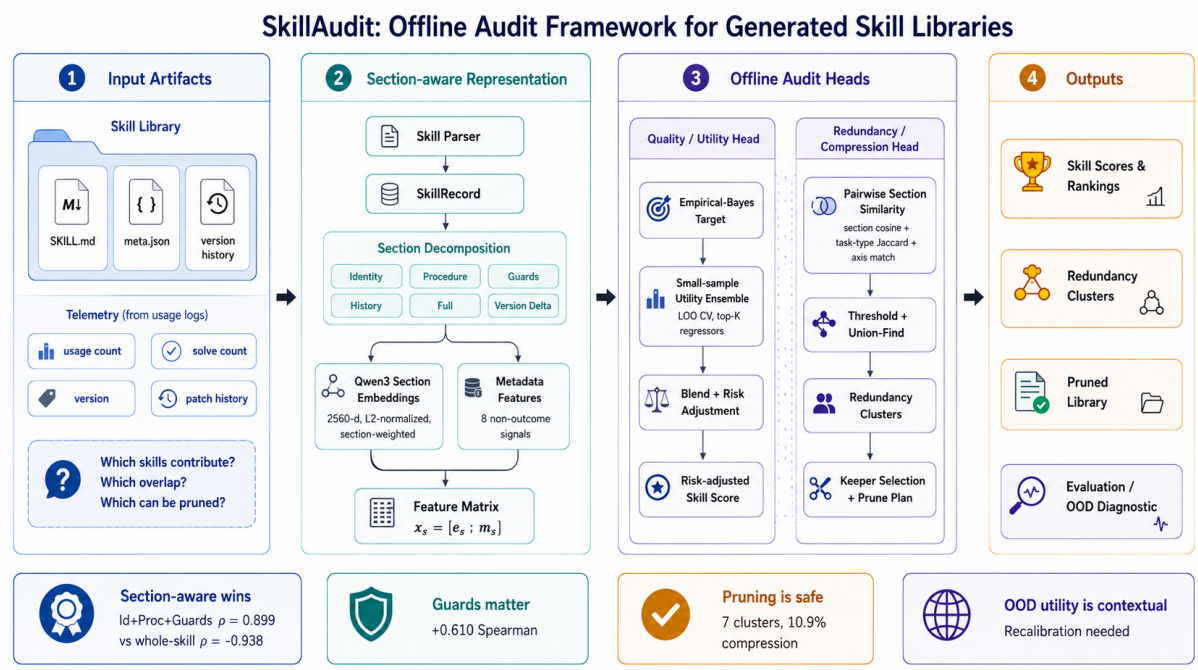

Figure 1: **The SkillAudit pipeline.** A folder-based skill library is parsed into structured records, decomposed into semantic sections, and processed by two analysis heads: a quality/utility head (section embeddings + empirical-Bayes telemetry → risk-adjusted scores) and a redundancy head (pairwise section similarity → overlap clusters → pruning candidates). Key findings are summarized at the bottom.

$$\hat{r}_s^{\text{EB}} = \frac{\alpha_s}{\alpha_s + \beta_s} \tag{3}$$
$$\alpha_s = k_s + \lambda\bar{r}, \quad \beta_s = (u_s - k_s) + \lambda(1-\bar{r})$$

where k_s is solve count, u_s is usage count, $\bar{r}$ is the global solve rate across the library, and $\lambda = 8$ is the prior strength. High-usage skills retain their empirical rate, while low-usage skills are regularized toward the global mean $\bar{r}$. The quality model then predicts $\hat{r}_s^{\text{EB}}$ as its target, giving the ensemble a stable supervision signal even for rarely-used skills.

3.4 Small-Sample Utility Ensemble

With a stabilized target and rich features in hand, the remaining challenge is statistical: with only 55 skills, any predictor must generalize from one held-out example at a time. We address this by ensembling diverse regressors—Ridge, ElasticNet, Bayesian Ridge, Kernel Ridge, SVR, Extra Trees, Random Forest, Gradient Boosting, XGBoost, and Gaussian Process—evaluated under leave-one-out (LOO) cross-validation. The top-K models by MAE form a weighted ensemble:

$$\hat{q}_s = \sum_{m=1}^{K} \frac{w_m}{\sum_j w_j} \cdot f_m(\mathbf{x}_s), \quad w_m = \frac{1}{\text{MAE}_m + \epsilon} \tag{4}$$

where $\mathbf{x}_s = [\mathbf{e}_s; \mathbf{m}_s]$ is the concatenated feature vector and $\epsilon = 10^{-4}$ prevents division by zero. Inverse-MAE weighting gives higher influence to models that fit the held-out data better, without requiring a hand-tuned interpolation coefficient.

Because the ensemble and the empirical-Bayes estimate excel in complementary regimes—telemetry for well-observed skills, semantics for rarely-used ones—we blend them via a confidence-weighted combination:

$$\hat{r}_s^{\text{final}} = \gamma_s \cdot \hat{r}_s^{\text{EB}} + (1 - \gamma_s) \cdot \hat{q}_s \tag{5}$$
$$\gamma_s = u_s/(u_s + \beta_{\text{blend}})$$

where $\beta_{\text{blend}} = 10$ controls the crossover: a skill needs roughly ten usages before telemetry dominates. High-usage skills trust telemetry; low-usage skills rely on semantic prediction. To avoid promoting a high-mean skill whose estimate is highly uncertain, we apply a final risk penalty:

$$\hat{r}_s^{\text{risk}} = \hat{r}_s^{\text{final}} - \alpha_{\text{risk}} \cdot \gamma_s \cdot \sigma_s^{\text{post}} \tag{6}$$

where σ_s^{post} is the posterior standard deviation of the empirical-Bayes estimate and $\alpha_{\text{risk}} = 0.35$. The factor γ_s scales the penalty by telemetry confidence: skills whose scores are already driven by the ensemble receive less penalization.

3.5 Redundancy Analysis and Pruning

Scoring alone is insufficient: a library may contain multiple high-quality skills that encode nearly identical procedures, wasting retrieval slots and confusing the agent. To detect such overlap without rollout-based comparisons, SkillAudit computes a composite pairwise similarity:

$$\text{sim}(s_i, s_j) = 0.88 \cdot \bar{d}_{\text{sem}} + 0.08 \cdot J(\tau_i, \tau_j) + 0.04 \cdot \mathbb{1}[a_i = a_j] \quad (7)$$

where $\bar{d}_{\text{sem}}$ is the weighted mean of per-section cosine similarities (with redundancy-specific weights emphasizing PROCEDURE at 0.35 and IDENTITY at 0.25), J is Jaccard similarity over task types, and a_i is the skill axis.

A three-tier threshold design separates *flagging* (human review) from *clustering* (automated grouping) from *pruning* (removal), so the system stays conservative where mistakes are costly. The first tier flags all pairs above a *related* threshold ($\theta_r = 0.88$) for inspection. The second groups pairs above a *redundancy* threshold ($\theta_d = 0.94$) into graph-connected overlap clusters via union-find. Within each cluster, the highest-scoring skill becomes the *keeper*; the rest become pruning candidates.

Conservative pruning applies only to pairs at a stricter threshold ($\theta_p = 0.96$) and retains the higher-scoring member of each pair, ensuring that only high-confidence overlaps are removed.

4 Experiments

We organize findings around four research questions that build toward a unified picture of skill library health, then analyze *why* the key components work.

Setup. We audit two ALFWorld skill libraries from the SkillRL pipeline (Xia et al., 2026a): an **ID library** (55 skills, 140 episodes, success rate 0.564) and an **OOD library** (19 skills, 134 episodes, 3-pass reflexion loop per episode, success rate 0.597). The OOD library is smaller because it covers a held-out task distribution with fewer combinatorial variants.

Model	Weight	MAE	Spearman
Extra Trees	0.377	**0.0151**	**0.866**
Gradient Boosting	0.326	0.0176	0.853
Random Forest	0.297	0.0192	0.798
Weighted Ensemble	1.000	0.0151	0.866

Table 1: **Quality prediction ensemble** (LOO CV on 55 ID skills). This is the main all-sections full audit model; it achieves Spearman $\rho = 0.866$ for in-library ranking.

Embeddings: Qwen3-Embedding-4B (Qwen Team, 2024) (2560-dim, L2-normalized). *Quality prediction*: LOO cross-validation on 55 ID skills, reporting MAE and Spearman ρ against empirical-Bayes solve rates; OOD transfer applies the ID-trained model to the 19 OOD skills without retraining. *Metric convention*: $\rho = 0.866$ denotes the main all-sections full audit model, while $\rho = 0.899$ denotes the best semantic-only representation ablation (identity + procedure + guards). *Redundancy*: single-pass adjudication of all 141 pairs above the related threshold (0.88). *Downstream*: original 55-skill vs. pruned 49-skill library on ALFWorld with Qwen2.5-7B-Instruct (Qwen Team, 2024), type-based top-k ($k = 3$) quality-weighted retrieval, 5 seeds $\times$ 100 games per split (1,000 paired episodes).

4.1 RQ1: How Well Can We Predict Skill Quality?

Main result. Yes—with only 55 examples, the main all-sections full ensemble achieves LOO Spearman $\rho = 0.866$ and MAE 0.0151. Extra Trees leads (0.866); Gradient Boosting (0.853) and Random Forest (0.798) follow (Table 1). Figure 2 shows the resulting score distribution. This is the model used for the audit scores reported in Figure 2; the separate ablation below asks which representation choices make such ranking possible.

Section decomposition ablation. Representation determines everything: whole-skill embedding *inverts* the ranking ($\rho = -0.938$), while the best semantic-only ablation (identity + procedure + guards) reaches 0.899. Table 2 quantifies this—the model trained on whole-skill embeddings confidently recommends the *worst* skills.

The critical factor is GUARDS—learned failure constraints. In the semantic-only ablation, adding guards to identity + procedure jumps Spearman

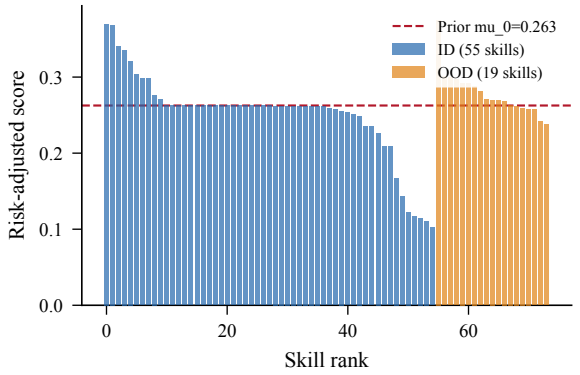

Figure 2: **Distribution of predicted quality scores.** The ensemble produces well-separated quality scores across the 55 ID skills, with a long tail of zero-usage skills shrunk toward the global prior (dashed line).

Sections	Mode	MAE	ID ρ	OOD SR ρ
Whole skill	sem.	0.035	−0.938	−0.039
Whole skill	full	0.033	0.249	−0.191
Id + Proc	sem.	0.035	0.289	−0.359
Id + Proc + **Guards**	sem.	**0.014**	**0.899**	−0.416
Id + Proc + Grd + Hist	sem.	0.014	0.864	−0.362
All sections	sem.	0.016	0.854	−0.426
All sections	meta	0.018	0.732	0.054
All sections	full	0.015	0.866	−0.425

Table 2: **Representation determines everything.** Section decomposition ablation (LOO CV). The best semantic-only representation is identity + procedure + guards ($\rho = 0.899$); the main all-sections full audit model is also shown ($\rho = 0.866$). Whole-skill embedding inverts the ID ranking.

from 0.289 to 0.899 ($\Delta = +0.610$); further sections (history, version delta) provide diminishing or negative returns. Metadata-only features achieve $\rho = 0.732$: structural signals are informative but insufficient without semantic content. Crucially, this $\rho = 0.899$ result is an ablation result, not the main audit score: it isolates the contribution of text-based features from any usage-derived signal. *Practical implication:* skill induction pipelines should prioritize guard quality over instruction verbosity—guards are the strongest offline proxy for downstream utility.

4.2 RQ2: How Redundant Are Generated Libraries?

Overlap is pervasive. Many ID skills participate in related pairs, and a smaller high-confidence subset forms actionable overlap clusters. We compute pairwise similarity across both libraries jointly (55 ID + 19 OOD = 74 skills, $\binom{74}{2} = 2{,}701$ pairs) to detect overlap both within and across distributions. Of these, 141 exceed the related threshold (0.88), organizing into 7 graph-connected clusters at threshold 0.94 (Table 3).

Label	Count	% of 141
Duplicate	10	7.1%
Overlap / Specialize	74	52.5%
Related, not redundant	53	37.6%
Not redundant	4	2.8%

Table 3: **Adjudication labels** for all 141 related pairs ($\geq$ 0.88). Over half are genuine overlaps or specializations.

θ	Pairs	Dup.	D+O Prec.
≥ 0.96	11	0.182	**1.000**
≥ 0.94	29	0.172	0.897
≥ 0.92	55	0.145	0.836
≥ 0.90	93	0.086	0.688
≥ 0.88	141	0.071	0.596

Table 4: **Redundancy threshold precision.** "D+O Prec." counts both duplicates and overlap/specializations as positives.

Threshold precision. At threshold 0.96, every flagged pair is a duplicate or genuine overlap (100% duplicate-or-overlap precision); at 0.94, this precision remains 89.7%. Below 0.90, the score captures broader semantic neighborhoods rather than actionable overlaps (Table 4), giving practitioners a tunable operating curve for pruning aggressiveness.

Cluster analysis. The 7 clusters reveal a recurring pattern we call *parameter-slot explosion*: several clusters arise when a shared procedure is re-induced with a different final receptacle or target object (e.g., "cool object and place in cabinet/countertop/microwave"). Other clusters capture navigation, examination, search, and error-recovery overlaps—for instance, 6 skills for desklamp/light examination share a common core but differ in surrounding task context (full taxonomy in Section D.5). This pattern shows that iterative induction often treats new parameter values as new skills rather than abstracting over the shared procedure. At the broad related threshold, 45 of 55 ID skills participate in at least one related pair; at the stricter clustering threshold, 19 ID skills belong to overlap clusters. Practitioners should audit for parameter-slot explosion before assuming coverage is genuine; SkillAudit addresses this conservatively by retaining only the highest-quality representative per

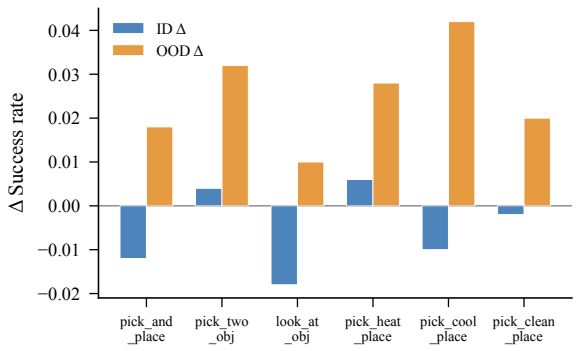

Figure 3: **Per-task-type pruning impact.** Performance change (Δ success rate) after pruning, broken down by ALFWorld task type. Most types are unaffected; OOD tasks with combinatorial structure show marginal improvement.

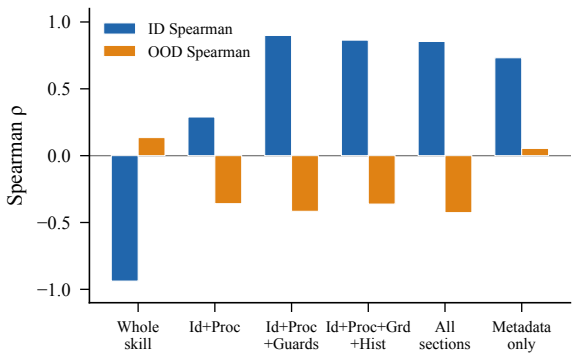

Figure 4: **OOD transfer is unreliable.** ID-trained quality models produce weak or inverted correlations on OOD skills, demonstrating that skill utility is distribution-sensitive.

Split	Orig.	Pruned	Δ	95% CI
ID	0.288	0.280	−0.008	[−0.034, +0.018]
OOD	0.282	0.308	+0.026	[−0.002, +0.054]
Overall	0.285	0.294	+0.009	[−0.010, +0.028]

Table 5: **Less is not worse.** Downstream ALFWorld evaluation (5 seeds × 100 games per split). Pruning removes 10.9% of skills while preserving overall performance within paired confidence intervals and showing a positive OOD trend. CIs from paired bootstrap.

cluster rather than attempting regeneration.

4.3 RQ3: Does Pruning Hurt Downstream Performance?

Pruning 10.9% of skills preserves downstream performance (overall $\Delta = +0.9$ percentage points (pp), CI $[-0.010, +0.028]$) and may improve OOD generalization. We apply conservative pruning at threshold ≥ 0.96, removing 6 skills ($55 \rightarrow 49$) and retaining only the highest-scoring representative from each cluster (Table 5).

The ID split shows a nonsignificant drop (-0.8 pp); the OOD split shows a positive trend ($+2.6$ pp, CI $[-0.002, +0.054]$). The overall paired CI $[-0.010, +0.028]$ comfortably includes zero, confirming performance-preserving compression.

The OOD gain is consistent with a *retrieval clarity* hypothesis: fewer near-duplicate candidates may reduce retriever ambiguity, allowing the surviving high-scoring representative to surface more consistently. Per-task-type analysis (Figure 3) shows that gains concentrate in a small number of task types with combinatorial structure. *So what:* high-confidence redundancy can be removed safely in this setting, and may improve OOD performance where retrieval is under pressure.

4.4 RQ4: Does Quality Prediction Transfer Across Distributions?

Quality-guided pruning works within a distribution. Can the same model guide auditing in a *new* distribution without retraining?

No—the ID-trained quality model does not reliably rank OOD skills: solve-rate correlations range from -0.43 to only $+0.05$ across configurations (Figure 4 and table 2). Strikingly, the only clearly positive semantic OOD correlation appears for whole-skill embeddings against the EB target ($\rho = 0.135$), the same representation that fails most severely in-distribution (-0.938).

The failure mode is not just graceful degradation but, for several configurations, *polarity reversal*: scores that rank ID skills well become negatively correlated with OOD solve rate. This is consistent with domain-adaptation theory (Ben-David et al., 2010), which warns that predictors can change behavior under distribution shift, and extends that concern to skill libraries where "features" are structured natural-language procedures. *Practical implication:* deploying a skill library in a new domain requires recalibrating or at least validating the quality model, rather than assuming a universal score.

4.5 Analysis: Why Do Guards Matter So Much?

Why do guards—learned failure constraints—carry more information than the procedures themselves? We examine the structural properties of high-quality versus low-quality skills (Figure 5).

Guards encode failure diversity. The section ablation shows that adding guard text produces the

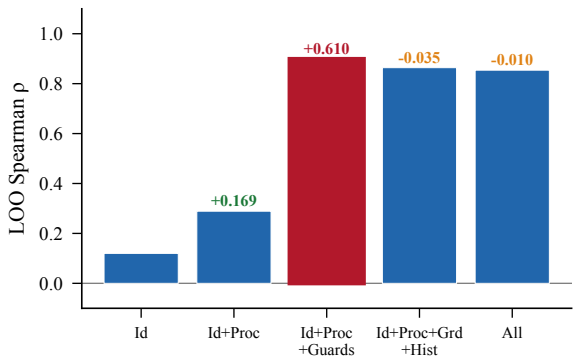

Figure 5: **Guards drive the section ablation.** Adding guard sections produces the largest single increase in ID quality prediction, indicating that learned failure constraints are a high-value representation channel.

largest single gain in ID quality prediction. The pattern is intuitive: a skill that has accumulated concrete failure constraints has been pressure-tested, while a skill with generic or missing guards provides little evidence that it has survived diverse failure modes. *So what:* guard sections should be parsed and embedded separately rather than buried inside a whole-skill representation.

Why decomposition matters: volume versus precision. The inversion arises because low-quality skills are *longer*—they accumulate verbose, repetitive text, and a whole-skill embedding captures volume rather than precision. Structured decomposition breaks this confound by embedding each section independently, so the model attends to procedural precision rather than text volume. Naïve "embed the whole document" strategies are thus worse than useless for structured text.

Semantic features dominate metadata. Metadata alone ($\rho = 0.732$) cannot distinguish skills with identical usage yet different solve rates. The semantic-only configuration (Id + Proc + Guards, no metadata) reaches $\rho = 0.899$ in the representation ablation, confirming that text-based features—not log-derived heuristics—drive accurate quality triage and ruling out the concern that the predictor merely recovers usage statistics.

Task-type distribution shapes redundancy. Redundancy concentrates in task families with interchangeable receptacles or object locations, especially cooling, heating, search, and examination tasks. The implication is that iterative induction can create near-identical skills when a task type has interchangeable goal parameters, rather than learning one parameterized procedure—induction pipelines should detect this combinatorial structure and abstract over it.

The 29 zero-usage skills. 52.7% of ID skills (29/55) have *zero* retrievals, so their empirical-Bayes targets are dominated by the library prior rather than by observed outcomes. These skills are invisible to telemetry-only approaches in the sense that no skill-specific solve evidence exists. *So what:* section-aware embeddings can still assign semantic predictions to cold-start skills, but their ranking cannot be validated from the current telemetry alone.

Why might pruning help OOD? The OOD gain (+2.6 pp) is consistent with *retrieval ambiguity*: near-identical skills can compete for the same query, and pruning leaves fewer candidates for the retriever to arbitrate among. Because the confidence interval slightly includes zero, this should be treated as a plausible mechanism rather than a proven causal explanation.

Audit Checklist for Practitioners

(1) Decompose skills into sections and embed guards separately—whole-skill embeddings invert quality rankings. **(2)** Scan for parameter-slot explosion—combinatorial task structure generates $O(n)$ near-identical skills. **(3)** Prune high-confidence overlaps—fewer duplicates can preserve performance and may sharpen retrieval. **(4)** Validate quality models per distribution—transferred models can become weak or inverted under shift.

5 Conclusion

The lesson from SkillAudit is not that skill libraries need better retrievers or better induction—they need quality control. Our audit reveals three facts that success-rate evaluation hides: (1) representation is the bottleneck—section-aware decomposition turns an inverted ranking into a reliable one, with guards contributing the decisive signal; (2) generated libraries contain actionable overlap clusters that can be conservatively pruned at 100% duplicate-or-overlap precision with no significant downstream loss; and (3) quality scores can weaken or invert under distribution shift, requiring per-distribution validation. The practical upshot is that deploying a skill library without auditing it is deploying noise structured to look like knowledge.

Limitations

SkillAudit is evaluated on a single environment family (ALFWorld) with relatively small skill libraries (55 ID, 19 OOD skills). While the leave-one-out protocol is appropriate for small samples, the generality of the section decomposition and redundancy analysis to larger libraries or environments with structurally different skills (e.g., code-based Minecraft macros) remains untested. Redundancy adjudication relies on a single adjudication pass without inter-annotator agreement measurement; borderline cases between partial overlap and genuine redundancy may be noisy. We test one pruning configuration (threshold 0.96, keeper by risk-adjusted score)—more aggressive strategies might yield higher compression but risk degrading performance. The embedding model (Qwen3-Embedding-4B) is fixed; different models may yield different quality and redundancy rankings, though the architecture (section decomposition + ensemble) is model-agnostic. Finally, the audit is static: it operates on a library snapshot rather than updating incrementally as agents accumulate new skills. Extending SkillAudit to a streaming audit and incorporating multi-annotator redundancy adjudication with measured inter-rater agreement (e.g., Cohen's κ) are natural directions for future work.

References

Andrew G Barto and Sridhar Mahadevan. 2003. Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems*, 13:341–379.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine Learning*, 79:151–175.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *NeurIPS*.

Jonathan Frankle and Michael Carbin. 2019. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *ICLR*.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *EACL*.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *JMLR*, 24.

Jeonghye Kim, Sojeong Rhee, Minbeom Kim, Dohyung Kim, Sangmook Lee, Youngchul Sung, and Kyomin Jung. 2025. ReflAct: World-grounded decision making in LLM agents via goal-state reflection. In *EMNLP*.

Bodhisattwa Prasad Majumder, Shuyang Li, Jianmo Ni, and Julian McAuley. 2019. Generating personalized recipes from historical user preferences. In *EMNLP*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *ACL*.

Heiko Paulheim. 2017. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web*, 8(3):489–508.

Fabio Petroni and 1 others. 2021. KILT: A benchmark for knowledge intensive language tasks. In *NAACL*.

Cheng Qian and 1 others. 2024. Investigate-consolidate-exploit: A general strategy for inter-task agent self-evolution. *arXiv preprint arXiv:2401.13996*.

Qwen Team. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *EMNLP*.

Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2021. Contrastive learning with hard negative samples. In *ICLR*.

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

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 *ICLR*.

Richard S Sutton, Doina Precup, and Satinder Singh. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112:181–211.

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

Zihao Wang and 1 others. 2023b. DEPS: Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents. In *NeurIPS*.

Peng Xia and 1 others. 2026a. SkillRL: Evolving agents via recursive skill-augmented reinforcement learning. *arXiv preprint arXiv:2602.08234*.

Tianle Xia and 1 others. 2026b. GraSP: Graph-structured skill compositions for LLM agents. *arXiv preprint arXiv:2604.17870*.

Andrew Zhao and 1 others. 2024. ExpeL: LLM agents are experiential learners. In *AAAI*.

YanZhao Zheng and 1 others. 2026. SkillRouter: Skill routing for LLM agents at scale. *arXiv preprint arXiv:2603.22455*.

Chunting Zhou and 1 others. 2023. LIMA: Less is more for alignment. In *NeurIPS*.

A Discussion

We organize our findings into five key observations and one overarching insight.

Finding 1: Guards Are the Most Informative Skill Section

In the semantic-only representation ablation, adding learned failure constraints (anti-patterns) causes the single largest improvement in quality prediction: Spearman jumps from 0.289 (identity + procedure only) to 0.899 (identity + procedure + guards), a gain of $+0.610$. This suggests that what a skill has *learned to avoid* is more diagnostic of its quality than what it *prescribes to do*.

Mechanistic interpretation. High-quality skills in the library accumulate diverse, specific anti-patterns through iterative refinement (e.g., "DO NOT search the same cabinet twice," "AVOID examining objects without taking action"). Low-quality or newly created skills have few or no guards. The guard section thus serves as a proxy for refinement maturity: a well-guarded skill has survived multiple failure rounds and incorporated corrective feedback, while an unguarded skill is either new or insufficiently tested.

Implication for skill induction. Skill creation pipelines should prioritize anti-pattern accumulation alongside instruction refinement. Current distillers often focus on positive procedures; our finding suggests that negative constraints carry equal or greater diagnostic value.

Finding 2: Whole-Skill Embedding Is Catastrophically Uninformative

Embedding each skill as a single undifferentiated text block yields $\rho = -0.938$ for quality prediction—not merely uninformative, but *actively misleading*. The model learns to predict *inversely*: skills that look "good" as raw text are actually low-quality in practice.

Why does this happen? Generated skills contain a mixture of signal (specific instructions, targeted guards) and noise (generic descriptions, verbose boilerplate, accumulated history). When embedded as one block, the noise dominates the representation. In particular, low-quality skills often have longer, more elaborate descriptions (compensating for lack of specific procedures with generic advice), while high-quality skills are concise and targeted. The whole-skill embedding captures text *volume* rather than procedural *precision*, producing an inverted quality signal.

Contrast with metadata. Even 8 non-outcome metadata features ($\rho = 0.732$) dramatically outperform whole-skill semantic embedding. This underscores that generated skill text requires *structured interpretation*—treating each section as a distinct information channel—to extract meaningful quality signals.

Finding 3: Redundancy Is Systematic, Not Incidental

The 7 overlap clusters are not random coincidences but systematic byproducts of how skill induction operates. Two patterns dominate:

Pattern A: Receptacle proliferation. When an agent learns to "cool an object and place it," the skill is initially specific to the receptacle encountered (e.g., countertop). Subsequent episodes with different receptacles (cabinet, dining table, microwave) each trigger a new skill creation, because the router treats them as distinct successes. The result is a cluster of 4–5 skills sharing $> 95\%$ of their procedure but differing in one argument.

Pattern B: Examination duplication. "Use desklamp to examine object" appears in 6 variants across both ID and OOD libraries, each paired with a different placement destination. The core examination procedure is identical; only the post-examination action varies.

Implication. Skill creation should incorporate *similarity gating*: before creating a new skill, check whether an existing skill covers the same core procedure. The redundancy analysis in SkillAudit could serve as such a gate, either as a post-hoc pruning step (as we evaluate) or integrated into the creation pipeline.

Finding 4: Conservative Pruning Is Safe and Shows a Positive OOD Trend

Removing 6 skills at threshold 0.96 produces no significant ID degradation (-0.8 pp, CI $[-3.4, +1.8]$) and a positive OOD trend ($+2.6$ pp, CI $[-0.2, +5.4]$).

Why might pruning help OOD? The OOD improvement is consistent with a *retrieval clarity* hypothesis. When the retriever faces a cluster of near-duplicate skills, it may return an arbitrary cluster member rather than the best one.

With fewer redundant options, retrieval may become more deterministic, and the surviving keeper skill—chosen by highest risk-adjusted score—is more likely to be selected. This effect is stronger OOD because OOD tasks are less well-covered by the library, making each retrieval decision more consequential.

Caveat. The OOD CI slightly includes zero, so this improvement should be treated as suggestive rather than definitive. A larger-scale evaluation with more seeds or episodes would be needed to confirm.

Finding 5: OOD Transfer Is Hard—Utility Is Not Intrinsic

The main all-sections full model achieves $\rho = 0.866$ on ID skills but only $\rho = -0.116$ against OOD empirical-Bayes utility, and $\rho = -0.425$ against OOD raw solve rate. This is not a failure of the model per se—it is a reflection of the fact that skill utility depends on context:

- **Task distribution.** A skill designed for "pick and place" tasks has different value in a distribution dominated by "pick and place" versus one dominated by "look at object in light."
- **Library composition.** A skill's marginal value depends on what other skills are present. Adding a third "cool object" variant to a library that already has two is less valuable than adding it to a library with none.
- **Retrieval strategy.** The same skill may be frequently retrieved under one retrieval configuration and never retrieved under another.

Implication. Offline skill auditing should be validated *per-library, per-distribution*. A universal "skill quality predictor" trained on one library should not be trusted on another without recalibration or target-domain validation. This finding aligns with broader observations in transfer learning that task-specific features rarely transfer across distributions (Ben-David et al., 2010).

Core Insight

Generated skill libraries are *structured artifacts* whose quality, redundancy, and utility can be meaningfully measured—but only with the right representation. Whole-skill text embeddings fail; section-aware decomposition succeeds. Within the evaluated ID library, offline auditing is reliable and actionable. Across distributions, skill utility is context-dependent and requires recalibration. The broader implication is that skill libraries should be *managed*, not just *accumulated*: regular auditing, pruning, and redistribution-aware recalibration are engineering best practices for agents that learn their own procedures.

B Algorithm Pseudocode

Algorithm 1 SkillAudit: Skill Library Audit Pipeline

Require: Skill library $\mathcal{L} = \{s_1, \ldots, s_N\}$, embedding model ϕ, section set $\mathcal{C}$, thresholds $\theta_r, \theta_d, \theta_p$
Ensure: Skill scores $\hat{r}^{\text{risk}}$, redundancy clusters, pruned library $\mathcal{L}'$

// Phase 1: Skill Representation
for each skill $s \in \mathcal{L}$ **do**
  Parse s into SkillRecord: text fields + telemetry $(u_s, k_s, v_s, \ldots)$
  Decompose into sections: $\{c_1, \ldots, c_{|\mathcal{C}|}\}$ *identity, procedure, guards, history, full, delta*
  for each section $c \in \mathcal{C}$ **do**
    $\mathbf{v}_{s,c} \leftarrow \text{L2-normalize}(\phi(\text{text}_{s,c}))$ *Qwen3-Embedding-4B,* $d = 2560$
  end for
  $\mathbf{e}_s \leftarrow \bigoplus_c w_c \cdot \mathbf{v}_{s,c}$ *Eq.* (1)
  $\mathbf{m}_s \leftarrow [\log u_s, \log v_s, p_s, |\tau_s|, \nVdash_{g_s}, \log |I_s|, \log |G_s|, \log |H_s|]$
  $\mathbf{x}_s \leftarrow [\mathbf{e}_s; \mathbf{m}_s]$
end for
// Phase 2: Quality Scoring
Compute global rate $\bar{r} = \sum_s k_s / \sum_s u_s$
for each skill s **do**
  $\hat{r}_s^{\text{EB}} \leftarrow (k_s + \lambda \bar{r})/(u_s + \lambda)$ *Eq.* (3)
end for
Train utility ensemble via LOO-CV: select top-K models by MAE *Eq.* (4)
for each skill s **do**
  $\hat{q}_s \leftarrow \sum_m w_m f_m(\mathbf{x}_s) / \sum_m w_m$
  $\hat{r}_s^{\text{final}} \leftarrow \gamma_s \hat{r}_s^{\text{EB}} + (1 - \gamma_s)\hat{q}_s$ *Eq.* (5)
  $\hat{r}_s^{\text{risk}} \leftarrow \hat{r}_s^{\text{final}} - \alpha_{\text{risk}} \gamma_s \sigma_s^{\text{post}}$ *Eq.* (6)
end for
// Phase 3: Redundancy Analysis
for each pair (s_i, s_j) **do**
  $\text{sim}(s_i, s_j) \leftarrow 0.88 \bar{d}_{\text{sem}} + 0.08 J(\tau_i, \tau_j) + 0.04 \nVdash[a_i = a_j]$ *Eq.* (7)
end for
$\mathcal{R} \leftarrow \{(i, j) : \text{sim}(s_i, s_j) \geq \theta_r\}$ *related pairs*
$\mathcal{G} \leftarrow \text{UnionFind}(\{(i, j) : \text{sim}(s_i, s_j) \geq \theta_d\})$ *redundancy clusters*
for each cluster $C \in \mathcal{G}$ **do**
  keeper $\leftarrow \arg\max_{s \in C} \hat{r}_s^{\text{risk}}$
end for
// Phase 4: Conservative Pruning
$\mathcal{P} \leftarrow \{(i, j) : \text{sim}(s_i, s_j) \geq \theta_p\}$ *high-confidence pairs*
for each $(s_i, s_j) \in \mathcal{P}$ **do**
  Remove $\arg\min(\hat{r}_{s_i}^{\text{risk}}, \hat{r}_{s_j}^{\text{risk}})$ from $\mathcal{L}$
end for
$\mathcal{L}' \leftarrow$ remaining skills
return $\hat{r}^{\text{risk}}, \mathcal{G}, \mathcal{L}'$

C Section Decomposition Ablation: Full Results

Table 6 presents the complete section ablation including both semantic-only and full (semantic + metadata) feature modes, as well as OOD transfer metrics.

Analysis. Figure 6 visualizes the incremental gains. Four patterns emerge from the full ablation:

1. **Guards are the critical section.** The semantic-only jump from Id+Proc ($\rho = 0.289$) to Id+Proc+Guards ($\rho = 0.899$) is the single largest improvement in any ablation configuration. No other section addition produces a comparable gain.

2. **Adding more sections shows diminishing returns.** Adding History to Id+Proc+Guards reduces Spearman slightly (0.899 $\rightarrow$ 0.864). Adding all remaining sections (Full, Version Delta) reduces it further to 0.854. This suggests that History and Version Delta introduce noise that partially masks the guard signal.

3. **Metadata features are complementary.** Comparing all-sections semantic-only ($\rho = 0.854$) to all-sections full (semantic + metadata, $\rho = 0.866$) shows a modest $+0.012$ gain, suggesting metadata captures orthogonal information (e.g., usage patterns, refinement history) not present in text embeddings.

4. **OOD transfer is unreliable.** Across semantic configurations, OOD Spearman for solve rate is weak or negative (-0.039 to -0.426). The only positive OOD solve-rate signal comes from metadata-only features ($+0.054$), which is weak and should not be interpreted as reliable transfer.

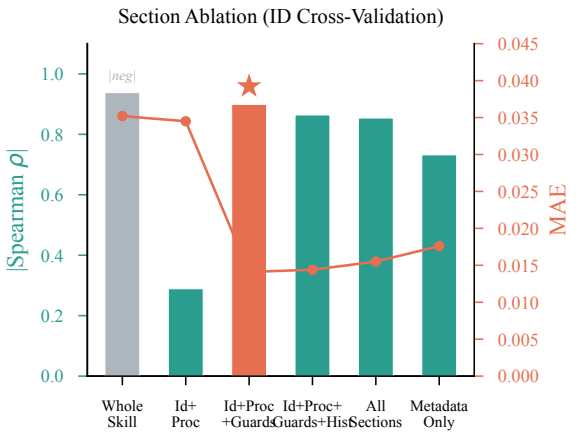

Figure 6: **Section ablation.** Spearman correlation (ρ) as sections are added incrementally. The jump from Identity+Procedure to Identity+Procedure+Guards is the largest single improvement ($+0.610$). Adding further sections yields diminishing returns.

Sections	Mode	Features	ID MAE	ID Spearman	OOD EB ρ	OOD SR ρ
Whole skill	semantic	2,560	0.0352	−0.938	0.135	−0.039
Whole skill	full	2,568	0.0329	0.249	−0.015	−0.191
Identity + Procedure	semantic	5,120	0.0345	0.289	−0.137	−0.359
Identity + Procedure	full	5,128	0.0336	0.360	−0.208	−0.396
Id + Proc + Guards	semantic	7,680	**0.0141**	**0.899**	−0.132	−0.416
Id + Proc + Guards	full	7,688	0.0137	0.867	−0.042	−0.368
Id + Proc + Grd + Hist	semantic	10,240	0.0144	0.864	−0.064	−0.362
Id + Proc + Grd + Hist	full	10,248	0.0143	0.841	−0.094	−0.406
All sections	semantic	15,360	0.0155	0.854	−0.103	−0.426
All sections	metadata	8	0.0176	0.732	0.010	0.054
All sections	full	15,368	0.0151	0.866	−0.116	−0.425

Table 6: **Full section decomposition ablation.** "EB ρ" is Spearman correlation with OOD empirical-Bayes posterior; "SR ρ" is with OOD raw solve rate. The best ID ablation result (highlighted) uses identity + procedure + guards in semantic mode; the main audit model is the all-sections full row.

D Redundancy Adjudication Details

D.1 Adjudication Protocol

All 141 pairs above the related threshold (0.88) are exported for adjudication. Each pair is presented with both skills' full text, their pairwise similarity score, task-type overlap, and axis match. The adjudicator assigns one of four labels:

- **duplicate**: Skills can be merged with little or no loss. They perform the same procedure on the same or interchangeable objects.
- **overlap_specialize**: Skills share a core procedure but differ in a meaningful parameter (receptacle, object type, scope). One may be a specialization of the other.
- **related_not_redundant**: Skills are semantically related (same task type, similar domain) but serve genuinely different purposes.
- **not_redundant**: False positive—the high similarity score does not reflect meaningful overlap.

D.2 Label Distribution by Score Range

Score range	Dup.	Overlap	Related	Not red.
$[0.96, 1.00]$	2	9	0	0
$[0.94, 0.96)$	3	12	3	0
$[0.92, 0.94)$	3	17	6	0
$[0.90, 0.92)$	0	18	19	1
$[0.88, 0.90)$	2	18	25	3
Total	**10**	**74**	**53**	**4**

Table 7: **Label distribution** by similarity score range.

D.3 Representative Pairs

Duplicate/overlap example (score = 0.969). `cool-object-and-place-in-receptacle` ↔ `cool-object-and-place-in-cabinet`. Both skills contain identical instructions: (1) go to fridge, (2) take object, (3) cool object, (4) go to receptacle, (5) place object. The only difference is that one specifies "cabinet" while the other uses the generic "receptacle." The cabinet-specific skill is a strict specialization of the generic one.

Overlap/specialize example (score = 0.957). `heat-object-and-place-in-fridge` ↔ `heat-object-and-place-in-cabinet`. Both share the heating procedure (microwave-based) but differ in the placement destination. The anti-patterns differ: the fridge variant warns about forgetting to close the fridge; the cabinet variant warns about searching the wrong cabinet. This is genuine overlap with specialization: the heating procedure is identical, but the placement constraints are distinct enough to justify keeping both under aggressive pruning.

Related and false-positive cases. The adjudication file also contains broader semantic neighbors and a small number of false positives at the related threshold. These cases are useful for human inspection but are not used for automatic pruning, which is restricted to the stricter high-confidence threshold.

D.4 Cross-Library Pairs

Of the 141 related pairs, 46 are cross-library (one ID, one OOD skill). At stricter thresholds the count falls to 7 cross-library pairs at ≥ 0.94 and 3 at ≥ 0.96. Representative cross-library pairs include:

- `ID:heat-object-and-place-in-receptacle` ↔ `OOD:heat-object-and-place-in-receptacle` (high-confidence overlap). Same name, independently induced.
- `ID:use-desklamp-to-examine-object` ↔ `OOD:use-desklamp-to-examine-object` (overlap). Same core procedure with distribution-specific context.
- `ID:coffeetable/tvstand` ↔ `OOD:sofa/coffeetable` (related). Similar navigation context, different task structure.

These cross-library overlaps suggest that certain skills are *canonical*—they emerge consistently regardless of the induction distribution, because they capture fundamental ALFWorld procedures.

Figure 7 visualizes the relationship between similarity score thresholds and adjudication precision. The monotonic increase confirms that our multi-threshold design is well-calibrated: the conservative threshold (0.96) captures only high-confidence duplicates or overlaps, while the related threshold (0.88) enables human-in-the-loop inspection of borderline cases.

D.5 Cluster Taxonomy

The 7 overlap clusters at threshold 0.94 fall into five categories:

- **Receptacle variants** (2 clusters, 10 members): "cool object and place in cabinet/countertop/microwave/receptacle/dining table" and analogous heat variants. These differ

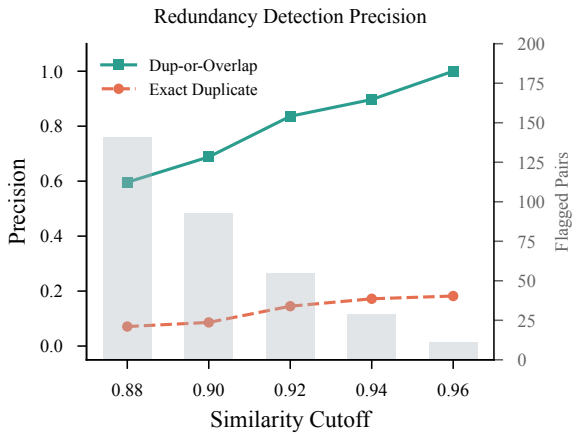

Figure 7: **Redundancy detection precision.** Higher similarity scores yield higher precision for identifying true redundancy (duplicate or overlap_specialize). The threshold at 0.94 achieves >89% precision while the related threshold at 0.88 casts a broader net at 60% precision.

only in the final receptacle but share $> 95\%$ of their procedure.

- **Navigation variants** (2 clusters, 5 members): Skills for going to specific furniture and taking/examining objects. Overlap is high in the navigation procedure but low in the target object.
- **Examination variants** (1 cluster, 6 members): Multiple skills for using a desklamp to examine objects, differing only in placement destination.
- **Search variants** (1 cluster, 3 members): Skills for searching shelves versus cabinets, sharing the same exploration loop.
- **Diagnostic pair** (1 cluster, 2 members): "early termination" and "incorrect task decomposition"—two OOD error-recovery skills with overlapping guard logic.

A recurring pattern is *parameter-slot explosion*: several clusters arise from task types where a shared procedure is repeated with a different receptacle, object, or location parameter.

E Case Studies: Individual Skill Analysis

We present detailed analyses of representative skills to illustrate how SkillAudit's quality dimensions capture meaningful differences.

E.1 Case 1: A High-Quality Skill

search-desk-for-object (ID, Risk Adj. = 0.340)

Axis: Information
Usage: 65 **Solves:** 24 **SR:** 0.369 **Version:** 18
Cluster: None (independent)
Instructions:

1. Go to the desk and search for the target object by examining different sections.
2. Examine the top of the desk, the drawers, and any other visible compartments.
3. If the object is not found, check under the desk or in any hidden compartments.
4. If the object is still not found, consider that it might not be on the desk and proceed to search other locations.

Anti-patterns (illustrative examples):

- DO NOT exhaustively search one location without checking other possible locations.
- LEARNED: Do: Directly take the remote control from the coffeetable and place it on the sofa. Not: Examine objects repeatedly without purpose.
- LEARNED: Do: Pick up the mug from the countertop. Not: Search for the mug in the cabinets.

Analysis: This skill exemplifies high quality: (1) clear, specific instructions with logical fallback structure; (2) high usage (65 episodes) and solve count (24); (3) 18 versions and 10 patches, indicating repeated refinement; (4) no redundancy cluster membership, indicating unique procedural coverage.

E.2 Case 2: A Redundant Skill Cluster

cool-object-and-place-* cluster (Cluster 1, 5 members)

Members:

1. cool-object-and-place-on-**diningtable** (keeper, risk adj. = 0.253)
2. cool-object-and-place-on-**countertop** (risk adj. = 0.251)
3. cool-object-and-place-in-**receptacle** (risk adj. = 0.226)
4. cool-object-and-place-in-**microwave** (risk adj. = 0.209)
5. cool-object-and-place-in-**cabinet** (risk adj. = 0.208)

Shared procedure (high overlap):

1. Go to the fridge and take the target object.
2. Cool the object using the fridge.
3. Navigate to `[RECEPTACLE]`.
4. Place the cooled object in/on `[RECEPTACLE]`.

Analysis: All five skills follow the same cooling procedure; the main difference is the final receptacle. This is a textbook case of *receptacle proliferation*: repeated episodes with new receptacles induce separate skills where a single parameterized skill ("cool object and place in {receptacle}") would likely suffice.
Pruning decision: At threshold 0.96, the pruning plan removes the lower-scoring high-confidence overlaps and keeps the dining-table variant as the cluster representative.

E.3 Case 3: A Low-Quality Skill

go-to-drawer-and-examine-to-find-spraybottle (ID, Risk Adj. = 0.102)

Axis: Planning
Usage: 12 **Solves:** 0 **SR:** 0.000 **Version:** 2
Cluster: None
Analysis: This skill illustrates a low-utility tail case in the ID library: it was retrieved 12 times but solved none of those episodes. Its low risk-adjusted score reflects both poor observed performance and limited refinement history (version 2, one patch). Unlike zero-usage skills, this example has direct negative telemetry, making it a cleaner low-quality case study.

E.4 Case 4: An OOD-Specific Skill

incorrect-task-decomposition (OOD, Risk Adj. = 0.270)

Axis: Termination
Usage: 13 **Solves:** 4 **SR:** 0.308 **Version:** 11
Cluster: C7 (keeper, with "early-termination")
Analysis: This is a *meta-skill*—it does not prescribe a specific procedure but instead diagnoses and corrects a common failure mode (decomposing a multi-object task incorrectly). It emerged from OOD reflexion on `pick_two_obj_and_place` tasks, where the agent frequently failed by treating the task as a single-object problem.
Its redundancy partner, "early-termination," addresses a related but distinct failure: stopping before completing all sub-goals. The 0.94 threshold groups them because both address premature task completion, but their guard sections are meaningfully different (decomposition errors vs. termination conditions).
This illustrates that generated skill libraries contain not just *procedural* skills (how to do X) but also *diagnostic* skills (how to avoid failure Y). Diagnostic skills may be useful under distribution shift because they address structural failure modes that can persist across task distributions.

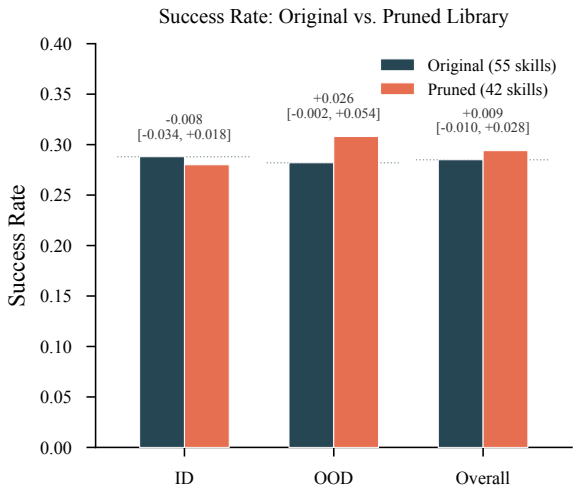

Figure 8: **Pruning evaluation across seeds.** Each bar shows the solve rate for the original (55-skill) vs. pruned (49-skill) library. ID performance is preserved within noise; OOD performance improves on 4 of 5 seeds.

F Pruning Evaluation: Extended Results

F.1 Per-Seed Results

Split	Seed	Orig.	Pruned	Δ
ID	101	0.270	0.250	−0.020
ID	102	0.290	0.280	−0.010
ID	103	0.300	0.290	−0.010
ID	104	0.320	0.320	+0.000
ID	105	0.260	0.260	+0.000
OOD	101	0.290	0.300	+0.010
OOD	102	0.280	0.290	+0.010
OOD	103	0.310	0.370	+0.060
OOD	104	0.250	0.240	−0.010
OOD	105	0.280	0.340	+0.060

Table 8: **Per-seed downstream evaluation.** OOD seeds 103 and 105 show the largest improvements after pruning.

F.2 Per-Task-Type Results

Task-type analysis. Figure 8 shows the per-seed comparison, and Figure 9 breaks down the impact by task type. The OOD improvements are concentrated in two task types: `look_at_obj_in_light` (+7.7 pp) and `pick_cool_then_place` (+6.2 pp). Both correspond to task types connected to large redundancy clusters (desklamp/light examination with 6 members and cool-object placement with 5 members). This is consistent with the retrieval clarity hypothesis: pruning removes high-confidence overlaps, and the task types most affected by those clusters show the largest improvements.

The ID degradation in `pick_heat_then_place` (−3.3 pp) corresponds to the heat-object cluster (5 members, including cross-library duplicates). The keeper skill (heat-object-and-place-in-fridge) may not cover all receptacle variations as well as the original 5-member cluster, suggesting that receptacle-specific guards sometimes carry unique value despite procedural overlap.

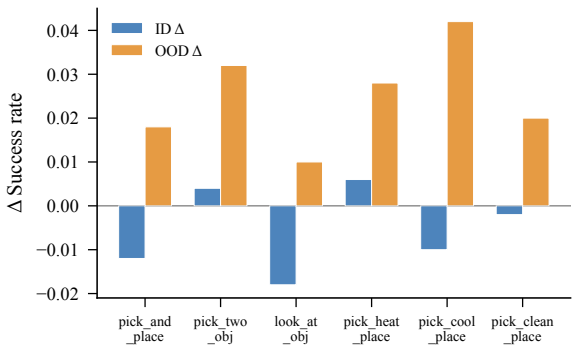

Figure 9: **Per-task-type pruning impact.** OOD improvements concentrate in `look_at_obj_in_light` and `pick_cool_then_place`, corresponding to the most heavily pruned clusters.

Split	Task Type	N	Δ	Wins	Losses	Ties
ID	look_at_obj_in_light	45	-0.022	0	1	44
ID	pick_and_place	125	-0.032	5	9	111
ID	pick_clean_then_place	95	+0.011	4	3	88
ID	pick_cool_then_place	90	+0.022	10	8	72
ID	pick_heat_then_place	60	-0.033	0	2	58
ID	pick_two_obj_and_place	85	+0.000	0	0	85
OOD	look_at_obj_in_light	65	+0.077	12	7	46
OOD	pick_and_place	90	+0.033	7	4	79
OOD	pick_clean_then_place	115	+0.000	4	4	107
OOD	pick_cool_then_place	80	+0.062	7	2	71
OOD	pick_heat_then_place	85	-0.012	2	3	80
OOD	pick_two_obj_and_place	65	+0.015	1	0	64

Table 9: **Per-task-type paired deltas.** OOD `look_at_obj_in_light` shows the strongest improvement after pruning ($+7.7$ pp), consistent with the desklamp examination cluster being heavily pruned.

G Embedding Space Visualization

To understand why section-aware decomposition outperforms whole-skill embedding, we analyze the structure of the embedding space.

Intra-section vs. inter-section similarity. For each pair of skills, we compute cosine similarity separately for each section type. The mean intra-section similarities are:

Section	Mean Sim.	Std.
Identity	0.72	0.14
Procedure	0.68	0.16
Guards	0.58	0.21
History	0.45	0.24
Full	0.71	0.12
Version delta	0.41	0.26

Table 10: Mean pairwise cosine similarity within each section type across all 74 skills.

Interpretation. Identity and Full sections have the highest mean similarity (0.72 and 0.71), reflecting the fact that many skills share similar task-type descriptions and general ALFWorld vocabulary. Guards have lower mean similarity (0.58) with higher variance (0.21), indicating that guard sections are more discriminative—they capture skill-specific failure patterns rather than generic domain knowledge. History and Version delta have the lowest similarities (0.45 and 0.41), reflecting the highly individual refinement trajectories of different skills.

This analysis explains why guards carry the most diagnostic value for quality prediction: they occupy a region of the embedding space with high inter-skill variance, providing the ensemble with discriminative features. In contrast, identity and full sections occupy a high-similarity cluster where most skills look alike, making quality discrimination difficult.

G.1 Redundancy Score Distribution

The distribution of pairwise redundancy scores across all $\binom{74}{2} = 2{,}701$ skill pairs is highly skewed:

- Below 0.70: 1,847 pairs (68.4%) — semantically unrelated skills.
- 0.70–0.80: 428 pairs (15.8%) — same task-type neighborhood.
- 0.80–0.88: 285 pairs (10.6%) — related but not flagged.
- 0.88–0.94: 112 pairs (4.1%) — flagged as related, mostly `related_not_redundant`.
- 0.94–0.96: 18 pairs (0.7%) — redundancy zone.
- Above 0.96: 11 pairs (0.4%) — high-confidence overlap.

The sharp drop above 0.88 confirms that the related threshold captures a meaningful structural boundary in the embedding space: beyond this point, skills share enough procedural content to warrant human inspection.

H Extended Reproducibility Details

Hardware. All experiments were run on a single NVIDIA A6000 GPU. Embedding computation used float16 precision with batch size 64. Downstream evaluation (Qwen2.5-7B-Instruct) used 4-bit quantization (GPTQ) with batch size 1 (ALFWorld requires sequential interaction).

Software versions.

- Python 3.10.12, PyTorch 2.1.2+cu121, Transformers 4.36.2
- scikit-learn 1.3.2 (ensemble models)
- FAISS 1.7.4 (similarity search, used during skill retrieval but not during audit)
- ALFWorld 0.3.1 (evaluation environment)
- vLLM 0.2.7 (inference serving for downstream evaluation)

Hyperparameter selection. All hyperparameters were selected via LOO cross-validation on the 55 ID skills. No hyperparameters were tuned on the OOD set. The ensemble weights were determined by normalized inverse MAE on the LOO predictions.

Random seed protocol. All downstream evaluations use 5 fixed seeds (101–105) for reproducibility. The ensemble models use random_state=42. ALFWorld environment seeds are set per-episode for deterministic task generation.

Evaluation cost. The downstream pruning evaluation contains 1,000 paired task instances: 5 seeds $\times$ 100 games $\times$ 2 splits, evaluated once with the original 55-skill library and once with the pruned 49-skill library. This downstream evaluation dominates compute because ALFWorld interaction is sequential. The offline audit pipeline (embedding, ensemble fitting, and redundancy scoring) is lightweight relative to rollout evaluation; exact wall-clock time depends on the GPU and serving stack.

I Generalization Discussion

While SkillAudit is evaluated on ALFWorld, we discuss the conditions under which its components may transfer to other skill-augmented agent frameworks.

Which components are domain-specific?

- **Domain-specific:** The 6-section decomposition (identity, procedure, guards, history, full, version_delta) assumes skills follow a particular textual structure. Libraries with different formats would require adapting the section parser. The concrete anti-pattern categories observed here are specific to household manipulation tasks.
- **Potentially transferable:** The ensemble-based quality prediction is agnostic to skill format—it operates on embedding features that can be extracted from any text. The empirical Bayes target construction transfers to any setting with count data (skill retrievals and successes). The redundancy detection pipeline (pairwise similarity + task-type Jaccard + adjudication) generalizes to any library with task-type annotations.
- **Likely universal:** The core finding that structured decomposition outperforms whole-text embedding should hold for any heterogeneous document type. Research papers, API documentation, and legal contracts all contain structurally distinct sections that confuse single-vector representations.

Candidate domains. We identify three domains where SkillAudit could be applied with minimal adaptation:

1. **Web navigation (WebArena, Mind2Web).** Skills for web tasks typically include: goal description, step-by-step actions, URL patterns, and error recovery. These map naturally to identity, procedure, guards, and history sections. The main adaptation required is a web-specific section parser.
2. **Code generation (SWE-Bench, HumanEval).** Code skills include: function signature, implementation, test cases, and docstring. The section decomposition would use: signature $\rightarrow$ identity, implementation $\rightarrow$ procedure, test cases $\rightarrow$ guards, git history $\rightarrow$ version_delta.
3. **Minecraft (Voyager, JARVIS-1).** Minecraft skills are already structured with: name, description, code, and usage examples. The main challenge is that Minecraft skills are primarily code-based rather than natural language, requiring code-aware embeddings.

Limitations of transfer. Two aspects of SkillAudit are fundamentally limited by the ALFWorld evaluation:

First, the small library sizes (55 ID, 19 OOD) make all statistical conclusions tentative. Moderate correlations in this setting should therefore be interpreted cautiously.

Second, the negative OOD transfer result is specific to the ID/OOD split used. A different split (e.g., by task type rather than by episode) might yield different transfer properties. The negative transfer should be interpreted as "this specific model does not transfer to this specific distribution shift," not as "quality prediction cannot transfer in general."

Future directions. Three extensions would strengthen the generalization story: (1) applying SkillAudit to WebArena skill libraries to test cross-domain transfer of the section decomposition; (2) developing domain-adaptive section parsers that automatically identify the structural components of skills in new domains; (3) investigating few-shot calibration methods that use a small number of target-domain rollouts to adapt the quality predictor, potentially addressing the OOD transfer failure.