SKILLCRAFT: SKILL-SET RETRIEVAL BEYOND RANKING FOR LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model (LLM) agents increasingly rely on external procedural skills to solve complex tasks, yet existing skill retrieval predominantly follows a relevance ranking formulation. We show that this formulation overlooks a critical gap: *candidate discovery does not imply set recoverability*. In multi-skill queries, target skills are frequently interleaved with irrelevant distractors, destroying rank-prefix recoverability. On SKILLRET, while top-50 retrieval achieves 89.6% candidate completeness, 46.5% of complete queries remain source-rank inseparable, admitting a median of seven unavoidable intruders under contiguous prefixes. To resolve this bottleneck, we propose SKILLCRAFT, which reformulates post-retrieval selection as a dedicated *set-resolution* problem. Rather than treating the upstream ranking as an immutable boundary, SKILLCRAFT constructs contextualized list-relative evidence and estimates non-monotonic membership probabilities, dynamically skipping high-ranked distractors to recover non-contiguous skill combinations. Candidate-level membership learning is further coupled with set-level calibration to govern variable-cardinality admission. On the 6,660-skill SKILLRET benchmark, SKILLCRAFT achieves the strongest skill-set recovery results, leading in both ExactSet and Set F1, and outperforming recent specialized skill retrieval systems (e.g., SKILLROUTER, SKILLREASON, and R3-SKILL). Beyond set recovery, its selected skill contexts yield higher downstream task accuracy across three external benchmarks under controlled execution settings. Moreover, SKILLCRAFT achieves the strongest candidate coverage among 17 competitive baselines across sparse, encoder-only, decoder-only, and two-stage retrieve-and-rerank pipelines.

1 INTRODUCTION

Modern large language model (LLM) agents increasingly augment foundation models with external tools, APIs, and procedural skills to support complex planning and execution beyond parametric knowledge (Luo et al., 2026; Xu & Yan, 2026). These skills encapsulate reusable capabilities such as executable scripts, API workflows, and invocation constraints, and are dynamically loaded into the agent context as needed. As skill libraries grow to thousands or even tens of thousands of heterogeneous entries, however, exposing the entire repository to an agent becomes impractical under finite context budgets (Su et al., 2026; Ling et al., 2026). This creates a fundamental retrieval problem: given a user query, identify the compact set of skills required for execution (Zheng et al., 2026).

Most existing approaches formulate this problem as *relevance ranking* (Cho et al., 2026; Zheng et al., 2026). Sparse retrievers, dense bi-encoders, and specialized retrieve-and-rerank pipelines estimate query–skill relevance and aim to place required skills as high as possible in a ranked list. Accordingly, progress is commonly measured with ranking-oriented metrics such as Recall and NDCG. Yet an agent ultimately does not execute a ranking: it consumes a *discrete skill set* placed in its prompt context. In practice, ranked candidates are therefore converted into an executable context through a top-k prefix or a related truncation rule. This conversion assumes that once the required skills have been successfully retrieved, a suitable prefix of the ranking can isolate them (Jiang et al., 2026).

Our key observation is that candidate completeness does not imply set recoverability. Even when a retriever places all target skills within its candidate pool, interleaving them with irrelevant candidates destroys rank-prefix recoverability: shorter prefixes omit indispensable tools, while longer prefixes admit distractors. We formalize this failure mode through *source-rank separability*, distinguishing

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

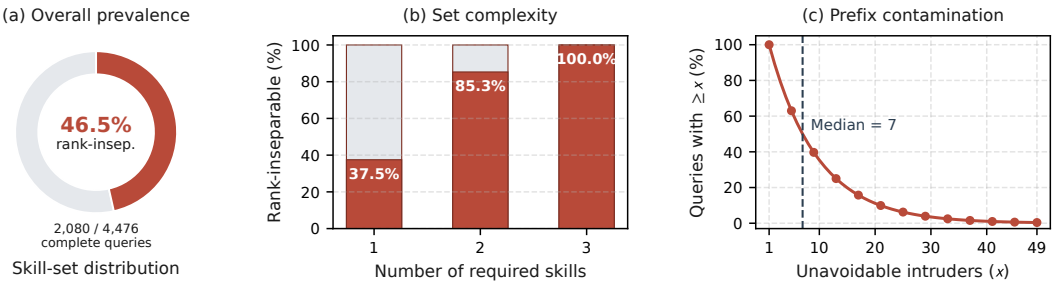


Figure 1: **Candidate completeness does not guarantee recoverable skill sets on SKILLRET.** (a) Overall prevalence: among candidate-complete queries, 46.5% remain source-rank inseparable. (b) Set complexity: inseparability increases from 37.5% for single-skill queries to 85.3% for two-skill queries and 100.0% for three-skill queries. (c) Prefix contamination: for rank-inseparable queries, the shortest ranked prefix containing required skills admits a median of seven unavoidable intruders.

candidate discovery (pool-level availability) from *set resolution* (discrete context admission). On SKILLRET (Cho et al., 2026), despite achieving 89.6% candidate completeness at top-50, 46.5% of candidate-complete queries remain strictly source-rank inseparable (Figure 1a). This inseparability escalates monotonically with set cardinality, affecting 37.5% of single-skill, 85.3% of two-skill, and 100.0% of three-skill queries (Figure 1b). Crucially, recovering all target skills through a contiguous prefix forces the admission of a median of 7 intruders (Figure 1c), demonstrating that improving candidate coverage alone cannot replace a dedicated set-resolution boundary, as distractor contamination consumes unnecessary context and may interfere with downstream execution..

To address this mismatch, we formulate *skill-set retrieval* as a *set-resolution* problem over a retrieved candidate pool and propose SKILLCRAFT. Rather than treating the source ranking as an immutable selection boundary, SKILLCRAFT treats it as evidence for deciding which candidates should belong to the final execution set. For each candidate, it combines source retrieval signals with query–skill compatibility and list-relative competition evidence, and estimates a calibrated membership probability. These membership scores are not constrained to follow the source order, allowing SKILLCRAFT to skip higher-ranked distractors and recover non-contiguous subsets of the candidate pool. Candidate-level membership learning is further coupled with set-level calibration, aligning the final admission threshold with exact set recovery and variable execution cardinality rather than with ranking quality alone under realistic multi-skill execution demands. Our contributions are threefold:

- We identify a structural gap between *candidate discovery* and *skill-set recovery*. We formalize source-rank separability and show that many candidate-complete queries are unrecoverable by any ranking prefix, exposing a failure mode missed by standard Recall and NDCG metrics.
- We propose SKILLCRAFT, a source-conditioned set-resolution framework that maps ranked candidates to variable-cardinality sets using contextualized evidence, non-monotonic membership estimation, and set-level calibration for non-prefix skill-set recovery beyond the source ranking.
- Experimental results show that SKILLCRAFT outperforms specialized skill-retrieval systems (e.g., SkillRet, SkillRouter, and SkillReason) and strong generic baselines in exact set recovery, while maintaining strong candidate retrieval and improving downstream task accuracy.

2 FROM CANDIDATE RETRIEVAL TO SET RESOLUTION

2.1 SKILL RETRIEVAL FORMULATION

Ranked candidate retrieval. Let $\mathcal{L}$ denote a skill library and q a query requiring a target skill set $G_q \subseteq \mathcal{L}$. Following existing skill-retrieval settings (Cho et al., 2026; Zheng et al., 2026; Jiang et al., 2026), an upstream retriever ranks the library and returns a candidate pool of depth K:

$$\mathcal{C}_K(q) = \big[(c_1, \rho_1), \ldots, (c_K, \rho_K)\big], \qquad \rho_1 \geq \cdots \geq \rho_K, \tag{1}$$

where c_i denotes the candidate at source rank i and ρ_i its retrieval score.

Candidate Retrieval Quality. Existing work evaluates candidate retrieval using Recall, Completeness, and NDCG (Cho et al., 2026; Zheng et al., 2026), which characterize whether required

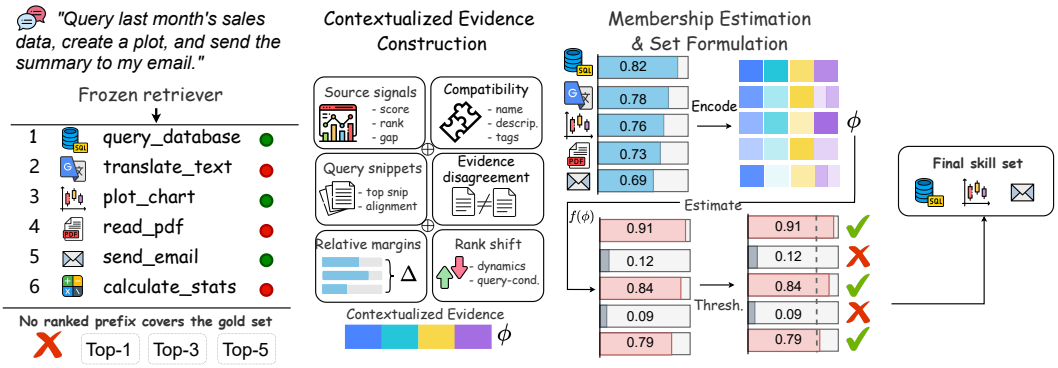


Figure 2: **Overview of SKILLCRAFT.** Given a frozen ranked candidate pool, SKILLCRAFT constructs source-conditioned evidence for each candidate, estimates non-monotonic set-membership scores, and applies calibrated thresholding to recover a non-prefix skill set.

skills are retrieved and how highly they are ranked. A query is *candidate-complete* at depth K if $G_q \subseteq \{c_1, \ldots, c_K\}$, i.e., all required skills are present in the retrieved pool and are therefore available for subsequent set resolution.

2.2 RANK-PREFIX RECOVERABILITY AND SET RESOLUTION

Source-Rank Separability. Ranked candidates are commonly converted into an agent-facing context through a top-k prefix (Cho et al., 2026). We call a candidate-complete query *source-rank separable* if some cutoff $k \leq K$ satisfies $\{c_1, \ldots, c_k\} = G_q$, and *source-rank inseparable* otherwise. Equivalently, every target skill must precede every non-target candidate in the retrieved pool.

When a non-target candidate $c_d \in \{c_1, \ldots, c_K\} \setminus G_q$ precedes a target $c_t \in G_q$, we call (c_d, c_t) a *target–intruder inversion*. Any prefix containing c_t must also contain c_d, making exact recovery impossible. Figure 1 shows that such inversions are prevalent in multi-skill queries and induce substantial distractor contamination, exposing a failure mode not captured by Recall or NDCG.

Set-Resolution Objective. We therefore formulate the post-retrieval stage as *set resolution*. Given $\mathcal{C}_K(q)$, a resolver predicts a variable-cardinality set $\hat{G}_q \subseteq \{c_1, \ldots, c_K\}$ without requiring a source-ranked prefix, with the goal $\hat{G}_q = G_q$. This distinguishes *discovery failure*, where some target skill is absent from the candidate pool, from *resolution failure*, where all targets are available but the predicted set is incorrect. SKILLCRAFT targets the latter through non-prefix candidate-level selection.

3 SKILLCRAFT: SOURCE-CONDITIONED SET RESOLUTION

Given the retrieved candidate pool $\mathcal{C}_K(q)$, SKILLCRAFT resolves which candidates should enter the final execution set. Its key principle is to treat the source ranking as *evidence* rather than as a selection constraint. As illustrated in Figure 2, SKILLCRAFT proceeds in three stages: (1) constructing contextualized evidence for each candidate from query–skill compatibility and list-relative signals; (2) estimating candidate membership without enforcing source-rank monotonicity; and (3) calibrating these membership scores into a variable-cardinality execution set.

3.1 CONTEXTUALIZED EVIDENCE CONSTRUCTION

For each candidate $c_i \in \mathcal{C}_K(q)$, SKILLCRAFT constructs a contextualized evidence representation ϕ_{qi}

$$\phi_{qi} = \phi_{qi}^{\text{cand}} \oplus \phi_{qi}^{\text{ctx}}, \tag{2}$$

where ϕ_{qi}^{cand} captures candidate-specific compatibility and ϕ_{qi}^{ctx} places that evidence in the context of the retrieved list.

Candidate-Specific Evidence. We combine source retrieval signals with query–skill compatibility at multiple granularities. Source scores and ranks preserve the relevance prior of the upstream retriever. Compact skill metadata provides functional compatibility, which we assess using a frozen query–candidate reranker (Zhang et al., 2025). Query-conditioned snippets further expose fine-grained procedural content that may be absent from the compact description; we retrieve locally relevant content from the skill body using standard lexical matching (Robertson & Zaragoza, 2009). Detailed feature construction is provided in Appendix A.

List-Relative Evidence. Candidate admission is inherently comparative: an absolute relevance score alone does not indicate whether a skill remains necessary relative to competing candidates. We therefore augment each candidate with list-relative signals, including score margins, pool-level statistics, rank disagreement across evidence sources, and relative competition within $\mathcal{C}_K(q)$.

3.2 Non-Monotonic Membership Resolution

Given the contextualized evidence ϕ_{qi}, SkillCraft estimates the probability that candidate c_i belongs to the target execution set:

$$p_{qi} = f_\theta(\phi_{qi}) \approx P(c_i \in G_q \mid q, c_i, \mathcal{C}_K(q)), \qquad p_{qi} \in [0, 1]. \tag{3}$$

Unlike source-ranked selection, the resulting scores $\{p_{qi}\}_{i=1}^{K}$ need not preserve the ordering of $\mathcal{C}_K(q)$. Membership can therefore vary non-monotonically with source rank, allowing lower-ranked targets to outrank earlier non-target candidates.

Candidate-Level Learning. For each retrieved candidate, we define the binary membership target $y_{qi} = \mathbb{I}[c_i \in G_q]$ and train f_θ using binary cross-entropy. This learns candidate membership independently of any fixed source-ranked cutoff; implementation details are provided in Appendix B.

3.3 Set Formation and Set-Level Calibration

Variable-Cardinality Set Formation. Given membership scores $\{p_{qi}\}_{i=1}^{K}$, SkillCraft forms

$$\hat{G}_q(\tau) = \{c_i \in \{c_1, \ldots, c_K\} \mid p_{qi} \geq \tau\}. \tag{4}$$

Unlike fixed top-k selection, this rule determines both set composition and cardinality from candidate membership, allowing non-contiguous subsets of the retrieved pool.

Set-Level Calibration. Candidate-level supervision does not determine the operating point that best recovers complete skill sets. We therefore calibrate τ directly against ExactSet:

$$\tau^* = \arg\max_{\tau} \sum_{q \in \mathcal{Q}_{\text{cal}}} \mathbb{I}\left[\hat{G}_q(\tau) = G_q\right]. \tag{5}$$

This aligns the admission boundary with exact set recovery rather than candidate-wise classification accuracy.

Grouped Cross-Fitting. To avoid calibrating τ on in-sample predictions, we obtain out-of-fold membership scores through grouped cross-fitting. Queries sharing the same target skill set are assigned to the same fold. After selecting τ^*, we refit the resolver on the full training set and keep both f_θ and τ^* fixed at inference time.

4 Experiment

4.1 Experimental Setup

Benchmarks and Evaluation. We primarily evaluate on SkillRet (Cho et al., 2026), with 4,997 test queries over 6,660 skills and 50,474 training queries. We consider (i) full-library retrieval over the complete repository and (ii) matched-pool set resolution over a shared frozen top-50 pool from `SkillRet-Embedding-0.6B`. We evaluate transfer to three external execution tasks from

Table 1: **Skill-set recovery on SKILLRET.** *Matched-pool* methods use the same frozen top-50 candidates from SkillRet-Embedding-0.6B, whereas *native* methods use their original retrieve-and-rerank pipelines. Δ denotes the paired improvement of SKILLCRAFT over each reference.

Method	Set recovery ↑				Diagnostic metrics ↑			
	ExactSet	Δ	Set F1	Δ	AnyHit	Precision	Recall	Jaccard
Matched-pool set resolution								
Relative-score threshold	32.60	+5.28	62.43	+3.82	81.35	76.05	57.81	54.38
SRA (Su et al., 2026)	35.52	+2.36	62.02	+4.22	77.93	77.93	54.95	54.95
Qwen3-32B (Qwen Team, 2025)	9.53	+28.36	53.08	+13.17	87.97	43.65	76.40	40.89
Pointwise Selector	35.06	+2.82	63.21	+3.04	80.19	80.19	55.69	55.69
Native retrieve & rerank								
SkillRet (Cho et al., 2026)	33.60	+4.28	60.48	+5.76	76.73	76.73	53.30	53.30
R3-Skill (Wang et al., 2026)	33.86	+4.02	61.55	+4.69	78.35	78.35	54.14	54.14
SkillRouter (Zheng et al., 2026)	36.32	+1.56	65.28	+0.97	82.67	82.67	57.55	57.55
SkillReason (Jiang et al., 2026)	35.54	+2.34	63.75	+2.49	80.75	80.75	56.22	56.22
SKILLCRAFT	**37.88**	–	**66.24**	–	83.45	75.66	63.94	58.82

SR-Agents (Su et al., 2026): TheoremQA (Chen et al., 2023), CHAMP (Mao et al., 2024), and MedCalc-Bench (Khandekar et al., 2024). We additionally evaluate on SKILLROUTER-EVAL-CORE (Zheng et al., 2026). For retrieval, we report NDCG, Recall, and Completeness at $K \in \{5, 10, 15\}$; for set resolution, we use ExactSet as the primary metric, together with Set F1, Precision, Recall, Jaccard, and AnyHit. Detailed metric definitions are provided in Appendix C.5.

Baselines. For full-library retrieval, we compare sparse retrieval, general-purpose dense retrievers, skill-specialized retrievers, and retrieve-and-rerank pipelines, including BM25, Qwen3, R3, SkillRet, SkillRouter, SkillReason, and SkillSight. For matched-pool resolution, we compare relative-score thresholding, prompted LLM selectors (SRA and Qwen3-32B), and a pointwise learned selector. Existing native retrieve-and-rerank systems are additionally reported as end-to-end references. Full baseline configurations are provided in Appendix C.

Implementation. Unless otherwise specified, set-resolution experiments use the frozen top-50 pool from `SkillRet-Embedding-0.6B`. SKILLCRAFT uses a frozen `Qwen3-Reranker-0.6B` to construct semantic compatibility evidence and trains a lightweight membership resolver. The admission threshold is selected using five-fold grouped cross-fitting on the training set by maximizing ExactSet over out-of-fold predictions, after which the resolver is refitted on all training queries. Feature construction and calibration details are provided in Appendices A and B. We report 95% confidence intervals from 5,000 paired query bootstraps.

4.2 MAIN RESULTS

Skill-Set Recovery. Table 1 reports the primary set-resolution results. Under the matched-pool setting, where all methods receive the same frozen top-50 candidates, SKILLCRAFT achieves 37.88% ExactSet, 66.24% Set F1, and 58.82% Jaccard. Compared with the strongest matched-pool baselines, SKILLCRAFT improves ExactSet by 2.36 pp over SRA LLM Selection (95% CI: [+1.10,+3.58]) and Set F1 by 3.04 pp over the Pointwise Selector (95% CI: [+2.42,+3.64]). The gain is not explained by simply admitting more candidates: direct Qwen3-32B selection attains high recall (76.40%) but over-admits irrelevant skills, reducing precision to 43.65% and ExactSet to 9.53%, whereas SRA and Pointwise are more conservative but recover fewer required skills. SKILLCRAFT achieves a stronger balance, with 75.66% precision and 63.94% recall, leading to higher exact set recovery.

Against native retrieve-and-rerank systems, SKILLCRAFT also exceeds SkillRouter by 1.56 pp in ExactSet (95% CI: [+0.34,+2.80]) and 0.97 pp in Set F1 (95% CI: [+0.10,+1.86]). Since native systems use different candidate pipelines, these results are best interpreted as end-to-end references rather than isolated estimates of the set-resolution effect. Full comparisons are reported in Table 11.

Downstream Task Accuracy. We further evaluate whether improved skill-set selection is reflected in downstream task accuracy by varying only the selected skill context while holding the executor and evaluation protocol fixed (Table 2). Relative to source top-1, SKILLCRAFT improves accuracy by 3.21 pp on TheoremQA, 3.59 pp on CHAMP, and 0.82 pp on MedCalc-Bench, and also outperforms the relative-score cutoff on all three tasks. These results show that the gains extend beyond set-recovery metrics to downstream task accuracy under controlled selected-skill contexts. Detailed configurations and execution protocols are provided in Appendix C.4.

Table 2: **Downstream task accuracy (%).** Within each task, only the selected skill context varies.

Selected Skills	ThQA	CHAMP	MedCalc
No skill	60.51	60.54	54.18
Source top-1	71.75	62.78	73.55
Rel.-score cutoff	73.63	62.78	74.00
SKILLCRAFT	**74.97**	**66.37**	**74.36**
Annotated gold	75.50	65.47	74.18

Equal-Budget Admission. To isolate admission quality from additional skill budget, we compare second-skill allocation policies under matched cohort-level budgets on SKILLROUTER-EVAL-CORE (Zheng et al., 2026). With the first skill fixed, all policies receive the same number of second-skill admissions and differ only in which queries receive them. As shown in Figure 3, SKILLCRAFT improves both ExactSet and Set F1 across Easy and Hard cohorts, including the size-2 subset where exact recovery is attainable. The advantage also persists as the candidate depth K increases, reaching a 17.5% relative Set-F1 improvement over uniform allocation at $K = 30$ and remaining substantial at $K = 50$. These results indicate that learned adaptive admission provides a stronger allocation signal than uniform, source-rank, or relative-score heuristics under the same skill budget.

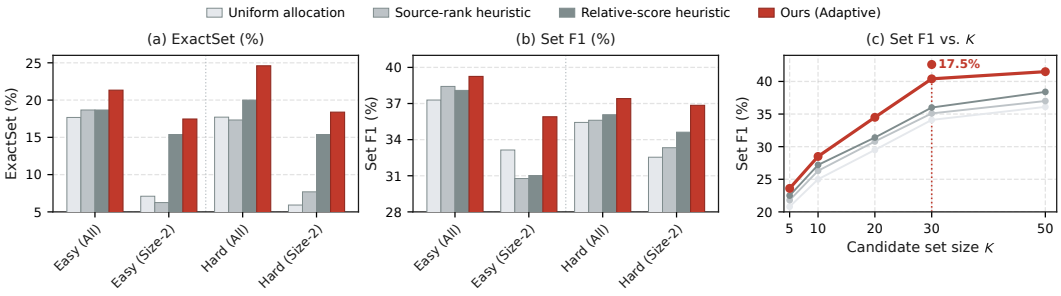


Figure 3: **Equal-budget second-skill admission on SKILLROUTER-EVAL-CORE.** (a) ExactSet and (b) Set F1 across query difficulty and target size. (c) Set F1 across varying candidate set sizes K.

Full-Library Candidate Retrieval. Finally, Table 3 verifies that improving set resolution does not come at the expense of candidate discovery. Across full-library retrieval on SKILLRET, SKILLCRAFT achieves 81.95% Recall@5, 90.50% Recall@15, and 82.81% Completeness@15, while maintaining strong NDCG across all cutoff depths. The results show that the proposed resolution stage preserves competitive candidate ranking while improving the downstream discrete set constructed from those candidates. This distinction is important: the primary gains of SKILLCRAFT arise from better set admission rather than from relying solely on higher source-ranking quality.

4.3 GENERALIZATION AND ROBUSTNESS

Across Retrieval Backbones. In Table 4, we first examine whether SKILLCRAFT remains effective with substantially stronger upstream retrieval. Using the frozen top-50 pool from `SkillRet-Embed-8B`, SKILLCRAFT consistently improves retrieval quality, raising Hit@1 from 80.73% to 82.55%, NDCG@10 from 83.45% to 84.66%, and Completeness@10 from 84.69% to 85.59%. These gains show that the learned set-resolution signals remain complementary to a strong source retriever **at larger scale**, rather than merely compensating for weak candidate ranking.

Robustness to Task Complexity and Threshold Stability. Figure 4 analyzes performance as skill-set construction becomes more difficult. As reasoning depth increases, all methods degrade, but the gap between SKILLCRAFT and pointwise or threshold-based policies widens on the most

Table 3: **Skill retrieval on SKILLRET.** We compare sparse, encoder-only, decoder-only, and two-stage retrieval methods; rerankers operate on candidates from their paired retrievers.

Model	NDCG ↑			Recall ↑			Completeness ↑		
	@5	@10	@15	@5	@10	@15	@5	@10	@15
Sparse & Encoder-only									
BM25 (2009)	44.00	45.99	46.93	47.29	52.67	55.79	32.60	37.44	40.54
bge-small-en-v1.5 (2023)	48.51	50.49	51.49	50.88	56.20	59.45	34.82	39.76	42.63
bge-large-en-v1.5 (2023)	53.75	55.81	56.71	55.96	61.40	64.31	39.30	44.63	47.49
e5-large-v2 (2022)	48.01	50.21	51.28	51.51	57.42	60.86	36.54	41.95	45.17
Decoder-only									
Qwen3-Embed-0.6B (2025)	56.10	58.36	59.34	59.03	65.02	68.19	41.77	47.41	50.55
harrier-oss-v1-0.6b (2026)	64.24	66.53	67.49	67.16	73.09	76.22	50.65	57.37	61.08
jina-embed-v5-text-small (2026)	57.28	59.53	60.48	59.83	65.73	68.76	42.69	49.03	52.29
R3-Embedding-0.6B (2026)	78.52	**81.06**	**81.89**	81.82	87.64	90.20	68.78	77.43	81.33
SkillRouter-Embed-0.6B (2026)	69.06	70.95	71.84	71.16	75.99	78.81	53.65	59.34	62.80
SkillReason-Embed-0.6B (2026)	68.74	70.99	71.87	70.87	76.64	79.39	52.97	60.00	63.64
SkillSight (2026)	58.59	60.88	61.87	61.50	67.54	70.70	44.27	50.79	54.41
SkillRet-Embedding-0.6B (2026)	75.54	77.99	78.84	79.10	85.34	88.06	65.94	75.03	78.89
Retrieve & Rerank									
Qwen3-Embed + Reranker (2025)	64.42	65.69	65.89	66.21	69.62	70.28	48.63	52.07	52.73
R3-Embed + Reranker (2026)	77.90	79.91	80.74	81.23	86.56	89.33	69.30	76.55	80.45
SkillRet + Reranker (2026)	72.90	75.90	76.77	78.16	85.86	88.58	64.56	75.49	79.91
SkillRouter + Reranker (2026)	76.63	77.50	77.78	77.33	79.64	80.56	61.32	63.88	64.96
SkillReason + Reranker (2026)	76.22	77.23	77.43	77.81	80.49	81.14	61.90	65.08	65.78
SKILLCRAFT	**78.79**	80.83	81.66	**81.95**	**87.88**	**90.50**	**69.86**	**78.65**	**82.81**

Table 4: **Large-model retrieval comparison on SKILLRET (%).** Evaluation of 8B+ foundation retrievers; SKILLCRAFT reranks the frozen top-50 candidate pool from SkillRet-Embedding-8B.

Model	Hit@1 ↑	NDCG ↑			Recall ↑			Completeness ↑		
		@5	@10	@15	@5	@10	@15	@5	@10	@15
KaLM-Gemma3-12B (2025)	53.17	52.52	55.27	56.53	56.59	63.88	67.96	40.32	47.81	52.19
Qwen3-Embed-8B (2025)	60.24	57.51	59.94	61.00	60.67	67.10	70.45	43.27	49.99	53.63
Octen-Embed-8B (2025)	64.94	60.48	62.65	63.63	62.68	68.33	71.50	45.21	51.59	55.43
SkillRet-Embed-8B (2026)	80.73	81.25	**83.45**	84.20	85.66	91.25	93.63	75.79	84.69	88.49
SkillRet-Embed-8B + SkillCraft	**82.55**	**82.50**	**84.66**	**85.34**	**86.40**	**91.94**	**94.05**	**76.89**	**85.59**	**89.15**

complex queries. A similar pattern appears as the number of required skills increases: simple policies provide limited gains over fixed Top-1 selection, whereas SKILLCRAFT benefits increasingly from adaptive multi-skill admission. This trend is consistent with the rank-inseparability analysis in subsection 4.4, where multi-skill queries are less likely to admit a clean source-ranked cutoff. Figure 4c evaluates sensitivity to the calibrated admission threshold. ExactSet varies by at most 0.37 pp over $\tau \in [0.44, 0.52]$ on the development set, indicating a relatively broad operating region around the selected threshold rather than reliance on a single finely tuned cutoff.

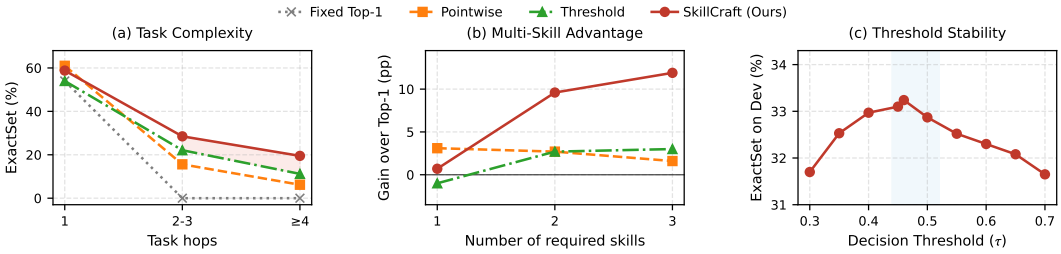


Figure 4: Robustness of set resolution on SkillRet. **(a)** ExactSet across reasoning-depth tiers. (b) Gain over fixed Top-1 as required skills increase. (c) ExactSet sensitivity to the admission threshold.

Across Generative Selectors. We further test whether set recovery can be improved simply by replacing the generative selector with a stronger backbone. In Table 5, Qwen3-32B and DeepSeek-V4.1-Flash (Xu et al., 2026) receive the same frozen top-50 candidates under an identical prompting

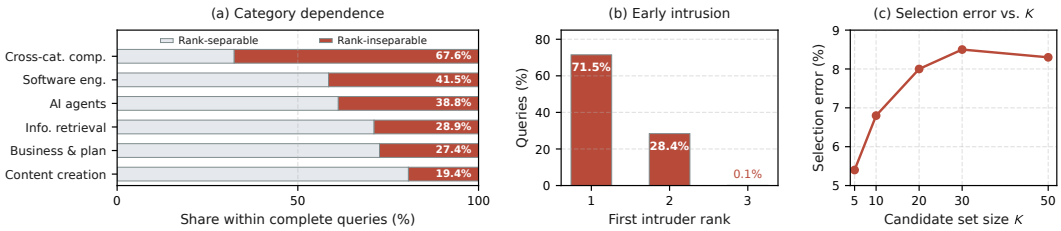


Figure 5: **Structure of source-rank inseparability on SKILLRET.** (a) Inseparability across skill categories. (b) Distribution of the first intruder rank. (c) Selection error under candidate depths K.

and parsing protocol. *Direct set* allows the model to return one to three candidates, whereas *select-one* requires exactly one prediction. Replacing Qwen3-32B with DeepSeek-V4.1-Flash raises direct-set ExactSet from 9.53% to 23.29%, but the resulting predictions remain recall-oriented. Constraining DeepSeek-V4.1-Flash to select one candidate increases ExactSet to 36.56% while substantially reducing recall. In comparison, SKILLCRAFT achieves 37.88% ExactSet and 66.24% Set F1, indicating that stronger generation alone does not remove the need for adaptive variable-cardinality admission. Full prompting, decoding, and parsing details are provided in Appendix C.3.

Table 5: **Generative selector generalization on SKILLRET (%).**

Method	ExactSet ↑	Set F1 ↑	Prec. ↑	Recall ↑
Qwen3-32B (direct set)	9.53	53.08	43.65	76.40
DeepSeek-V4.1-Flash (direct set)	23.29	61.79	54.04	80.49
DeepSeek-V4.1-Flash (select-one)	36.56	65.16	82.41	57.52
SKILLCRAFT	**37.88**	**66.24**	75.66	63.94

4.4 MECHANISM ANALYSIS

Structure of Rank Inseparability. Figure 5 characterizes where rank-prefix recovery breaks down. Rank inseparability is most pronounced for cross-category compositions, indicating that heterogeneous skill requirements are particularly difficult to isolate with a single ranking boundary. Moreover, nearly all rank-inseparable queries encounter their first intruder within the top two positions, showing that these failures cannot be repaired by merely adjusting the tail cutoff. Finally, selection error remains non-negligible across candidate depths K, even when evaluation is restricted to queries whose complete gold skill set is already available within the corresponding top-K pool. Thus, increasing candidate coverage alone does not eliminate the difficulty of recovering the correct skill set. These results reinforce that candidate discovery and set resolution constitute distinct bottlenecks.

Table 6: **Contextualized-evidence ablation on SKILLRET.** All variants share the same frozen top-50 candidates and training protocol. Parentheses show changes from the preceding configuration.

Configuration	Scalar Snippet	Source/Card Context	Snippet Context	ExactSet ↑	Set F1 ↑
Raw-score baseline	✓	–	–	35.06	63.21
Source/card contextualization	–	✓	–	36.14 (+1.08)	65.00 (+1.79)
+ Scalar snippet evidence	✓	✓	–	36.42 (+0.28)	65.05 (+0.05)
+ Snippet context **(Full)**	✓	✓	✓	**37.88 (+1.46)**	**66.24 (+1.19)**

Evidence Contextualization. Table 6 examines whether the improvement of SKILLCRAFT comes from simply adding snippet compatibility, or from contextualizing that signal within the candidate pool. Adding the scalar snippet score to the source/card-context representation yields only a modest 0.28 pp ExactSet gain (95% CI: $[-0.22, +0.78]$). In contrast, adding snippet-context features improves ExactSet from 36.42% to 37.88% ($+1.46$ pp; 95% CI: $[+0.68, +2.24]$) and Set F1 from 65.05% to 66.24%. The paired comparison yields 240 ExactSet wins and 167 losses, with gains concentrated on two-skill queries ($+6.13$ pp) despite a 2.67 pp drop on single-skill queries. Since the

selected threshold remains 0.5, the gain reflects improved membership evidence rather than threshold shifting. See Appendix D.2 for detailed breakdowns.

Ranking vs. Admission. Table 7 factorizes ordering and cardinality effects. With predicted cardinalities fixed, replacing the source ordering with SKILLCRAFT's learned membership ordering raises ExactSet from 36.30% to 37.88% ($+1.58$ pp). In contrast, oracle cardinality increases ExactSet to 47.95% and 52.19% under source and learned ordering, respectively. The resulting $+14.31$ pp cardinality gap indicates that accurate admission remains a larger bottleneck than ordering alone.

Table 7: **Ranking vs. admission.** ExactSet under predicted and oracle cardinality.

Ranking	Pred. Size	Oracle Size	Gap
Source	36.30	47.95	+11.65
Learned	**37.88**	52.19	+14.31
Ranking Δ	*+1.58*	*+4.24*	–

Table 8: **Decomposition of ExactSet gain.** Net gain from replacing source ordering.

Query Subset	# Queries	Net Gain	Share
Rank-separable	2,396	+38	48.1%
Rank-inseparable	2,080	**+41**	**51.9%**
Candidate-complete	4,476	+79	100.0%

Rank-Inseparable Queries. Table 8 further decomposes the $+1.58$ pp ExactSet gain over source ordering under identical predicted cardinalities. Among the 79 net additional exact matches, 41 (51.9%) come from source-rank-inseparable queries, for which no source-ranked prefix can recover the gold set. This directly supports the benefit of non-monotonic admission: SKILLCRAFT can skip intervening intruders while retaining later targets, where prefix-based selection is incapable.

5 Related Work

Tool and Skill Retrieval for LLM Agents. External tools and procedural skills extend LLM agents beyond their parametric capabilities, but large tool libraries require retrieval to control context cost and prompt complexity (Schick et al., 2023; Qin et al., 2024; Xu & Yan, 2026; Ling et al., 2026). Recent benchmarks and systems, including SKILLRET, SKILLROUTER, and SKILLREASON, formulate skill selection primarily as relevance ranking (Cho et al., 2026; Zheng et al., 2026; Jiang et al., 2026). Both general-purpose dense retrievers and skill-specialized encoders improve candidate discovery, while recent methods further enhance retrieval through reasoning-aware representations or skill-specific adaptation (Xiao et al., 2026; Wang et al., 2026). Our work is complementary to these advances: rather than replacing candidate retrieval, SKILLCRAFT addresses the subsequent problem of converting a retrieved candidate pool into the discrete skill set consumed by the agent.

Reranking, Selection, and Cardinality Control. Two-stage retrieval systems use cross-encoders or learned rerankers to refine candidate ordering (Zheng et al., 2024; Chen et al., 2026; Team et al., 2026). Other approaches directly select tools through LLM prompting (Su et al., 2026), generative composition (Zhao et al., 2026), dynamic context construction (Li et al., 2026), or adaptive truncation (Taguchi et al., 2025). These methods improve candidate relevance or determine where a ranked list should be truncated, but retain either pointwise scoring or rank-prefix selection. Related multi-label prediction work studies variable-cardinality outputs and threshold calibration (Tsoumakas & Katakis, 2007); however, skill retrieval introduces an additional source-ranking structure and target–intruder interleaving. SKILLCRAFT combines these perspectives by treating the retrieved ranking as evidence rather than as a fixed selection boundary, enabling non-monotonic, variable-cardinality set resolution.

6 Conclusion

We introduced SKILLCRAFT, a skill-set retrieval framework that separates candidate discovery from final set resolution. We show that high candidate completeness does not guarantee recoverable execution sets when target and irrelevant skills are interleaved in the source ranking. By enabling non-monotonic, variable-cardinality admission, SKILLCRAFT improves exact skill-set recovery beyond ranked-prefix selection. Our results establish set resolution as a distinct and necessary stage for constructing execution-ready skill contexts for LLM agents across retrieval and execution settings.

AI USE STATEMENT

LLMs were used for auxiliary support, including literature search assistance, grammatical editing, and code debugging.

ETHICS STATEMENT

This work studies procedural-skill retrieval for LLM agents using existing benchmarks and pretrained models, without human subjects or private data. Real-world deployment should apply appropriate access control and human oversight when retrieved skills can trigger external actions.

REFERENCES

Mohammad Kalim Akram, Saba Sturua, Nastia Havriushenko, Quentin Herreros, Michael Günther, Maximilian Werk, and Han Xiao. jina-embeddings-v5-text: Task-targeted embedding distillation. *arXiv preprint arXiv:2602.15547*, 2026. URL https://arxiv.org/abs/2602.15547.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26, 2013.

Wenhu Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and Tony Xia. Theoremqa: A theorem-driven question answering dataset. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7889–7901, 2023.

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

Hongcheol Cho, Ryangkyung Kang, and Youngeun Kim. SkillRet: A large-scale benchmark for skill retrieval in LLM agents. *arXiv preprint arXiv:2605.05726*, 2026.

Paul Jaccard. Distribution de la flore alpine dans le bassin des dranses et dans quelques régions voisines. *Bulletin de la Société Vaudoise des Sciences Naturelles*, 37(140):241–272, 1901.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002. doi: 10.1145/582415.582418.

Donghong Jiang, Endian Lin, Luoping Cui, Hanqing Liu, Mingjie Liu, Fan Yang, Hong Wang, Zhao Yang, and Chuang Zhu. SkillReason: Reasoning-enhanced agent skill retrieval for implicit user requests. *arXiv preprint arXiv:2608.08640*, 2026.

Nikhil Khandekar, Qiao Jin, Guangzhi Xiong, Soren Dunn, Serina S Applebaum, Zain Anwar, Maame Sarfo-Gyamfi, Conrad W Safranek, Abid A Anwar, Andrew Zhang, et al. Medcalc-bench: Evaluating large language models for medical calculations. *Advances in Neural Information Processing Systems*, 37:84730–84745, 2024.

Yanchao Li, Wanhao Liu, Ben Gao, Jiaqing Xie, Zhehong Ai, Na Zou, Yuqiang Li, and Tianfan Fu. Skillsinjector: Dynamic skill context construction for llm agents. *arXiv preprint arXiv:2605.29794*, 2026.

George Ling, Shanshan Zhong, and Richard Huang. Agent skills: A data-driven analysis of claude skills for extending large language model functionality. *arXiv preprint arXiv:2602.08004*, 2026.

Jinghao Luo, Yuchen Tian, Chuxue Cao, Ziyang Luo, Hongzhan Lin, Kaixin Li, Chuyi Kong, Ruichao Yang, and Jing Ma. From storage to experience: A survey on the evolution of llm agent memory mechanisms. *arXiv preprint arXiv:2605.06716*, 2026.

Yujun Mao, Yoon Kim, and Yilun Zhou. Champ: A competition-level dataset for fine-grained analyses of llms' mathematical reasoning capabilities. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 13256–13274, 2024.

Microsoft. harrier-oss-v1. Hugging Face model release, 2026. URL https://huggingface.co/microsoft/harrier-oss-v1-0.6b. Multilingual text embedding model family.

Octen Team. Octen series: Optimizing embedding models to #1 on rteb leaderboard. Technical blog and model release, 2025. URL https://octen-team.github.io/octen_blog/posts/octen-rteb-first-place/.

Yujia Qin, Shi Liang, Yining Ye, Yaxi Yan, Kun Lu, Yusheng Lin, Ling Cong, Xi Tang, Bill Qian, Xiaofei Han, et al. Toolllm: Facilitating large language models to master 16000+ real-world apis. In *International Conference on Learning Representations*, pp. 9695–9717, 2024.

Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Foundations and trends® in information retrieval*, 4(1-2):1–174, 2009.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551, 2023.

Weihang Su, Jianming Long, Qingyao Ai, Yichen Tang, Changyue Wang, Yiteng Tu, and Yiqun Liu. Skill retrieval augmentation for agentic AI. *arXiv preprint arXiv:2604.24594*, 2026.

Chihiro Taguchi, Seiji Maekawa, and Nikita Bhutani. Efficient context selection for long-context QA: No tuning, no iteration, just adaptive-k. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20105–20130. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.1017.

HONOR Agentic Search Team, Zhengzong Chen, Lei Tang, Lijun Liu, Chuandi Jiang, Fan Yang, Keyun Chu, Chu Zhao, Shihao Liu, Minghang Li, et al. Magicselector: Joint optimization for agent tool selection via counterfactual decomposition and progressive reranking. *arXiv preprint arXiv:2607.17751*, 2026.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *NeurIPS Datasets and Benchmarks*, 2021.

Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3):1–13, 2007.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training, 2022. URL https://arxiv.org/abs/2212.03533.

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

Cort J. Willmott and Kenji Matsuura. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research*, 30:79–82, 2005. doi: 10.3354/cr030079.

Jinying Xiao, Bin Ji, Shasha Li, Xiaodong Liu, Jun Ma, Jiacheng Jie, Chao Wang, Nyima Tashi, and Jie Yu. SkillSight: Seeing through shared descriptions for accurate skill retrieval. *arXiv preprint arXiv:2607.18785*, 2026.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-Pack: Packed resources for general chinese embeddings. *arXiv preprint arXiv:2309.07597*, 2023.

Anyi Xu, B Li, Bangcai Lin, Bing Xue, BingCheng Xian, Bingzheng Xu, Bochao Wu, Bowei Zhang, Boyi Deng, CC Yu, et al. Deepseek-v4. 1-flash: Pushing the limits of kv cache compression. *arXiv preprint arXiv:2609.19969*, 2026.

Renjun Xu and Yang Yan. Agent skills for large language models: Architecture, acquisition, security, and the path forward. *arXiv preprint arXiv:2602.12430*, 2026.

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

Xinping Zhao, Xinshuo Hu, Zifei Shan, Shouzheng Huang, Yao Zhou, Xin Zhang, Zetian Sun, Zhenyu Liu, Dongfang Li, Xinyuan Wei, Youcheng Pan, Yang Xiang, Meishan Zhang, Haofen Wang, Jun Yu, Baotian Hu, and Min Zhang. Kalm-embedding-v2: Superior training techniques and data inspire a versatile embedding model, 2025. URL https://arxiv.org/abs/2506.20923.

Xinyu Zhao, Zhen Tan, Vaishnav Tadiparthi, Nakul Agarwal, Kwonjoon Lee, Ehsan Moradi Pari, Hossein Nourkhiz Mahjoub, and Tianlong Chen. Generative skill composition for llm agents, 2026. URL https://arxiv.org/abs/2606.32025.

YanZhao Zheng, ZhenTao Zhang, Chao Ma, YuanQiang Yu, JiHuan Zhu, Baohua Dong, and Hangcheng Zhu. SkillRouter: Retrieve-and-rerank skill selection for LLM agents at scale. *arXiv preprint arXiv:2603.22455*, 2026.

Yuanhang Zheng, Peng Li, Wei Liu, Yang Liu, Jian Luan, and Bin Wang. Toolrerank: Adaptive and hierarchy-aware reranking for tool retrieval. In *Proceedings of the 2024 joint international conference on computational linguistics, language resources and evaluation (LREC-COLING 2024)*, pp. 16263–16273, 2024.

A CONTEXTUALIZED SKILL EVIDENCE

This appendix provides implementation details for the contextualized evidence representation introduced in Section 3. For each query–candidate pair (q, c_i), SKILLCRAFT constructs

$$\phi_{qi} = \phi_{qi}^{\text{cand}} \oplus \phi_{qi}^{\text{ctx}}, \tag{6}$$

where ϕ_{qi}^{cand} captures candidate-specific query–skill compatibility and ϕ_{qi}^{ctx} describes the candidate relative to the retrieved list. The full configuration contains 30 scalar features. Table 9 summarizes the four evidence groups.

Table 9: **Contextualized evidence used by SKILLCRAFT.** The 30-dimensional representation combines candidate-specific compatibility signals with list-relative competition evidence. Signals shown in the second column are representative features within each group.

Evidence Group	Representative Signals	Dim.	Role
Source retrieval	ρ_i, normalized rank, reciprocal rank, local score margins	6	Preserve source-retriever confidence
Card compatibility	$s_{\text{card}}(q, c_i)$, cross-scorer rank, rank displacement	5	Measure query–skill semantic fit
Snippet alignment	$s_{\text{snip}}(q, c_i)$, snippet rank and gaps, score/rank disagreement, pool statistics	11	Ground decisions in procedural content
List competition	margin to top-1, pool-relative statistics, cross-evidence disagreement	8	Measure relative necessity within the pool

A.1 CANDIDATE-SPECIFIC EVIDENCE

Source Retrieval Signals. Let ρ_i denote the score assigned to candidate c_i by the frozen source retriever. We retain the retrieval score, normalized source rank, reciprocal rank, and gaps to neighboring and top-ranked candidates. These signals preserve the upstream relevance prior without making source order a hard constraint on final set membership.

Compact Skill Representation. For each skill, we form a compact card from its available name, description, tags, and category. A frozen `Qwen3-Reranker-0.6B` scores the query against this card, producing one compact compatibility score $s_{\text{card}}(q, c_i)$. We also derive the candidate's rank and score gaps within the query's compact-score list.

Query-Conditioned Snippet Evidence. Compact cards can omit procedural details relevant to execution. We therefore partition the available skill body and code into tokenizer-defined, 256-token chunks with a 32-token overlap. BM25 selects the two chunks most relevant to the query from that skill. We concatenate the compact card and these two chunks into a single snippet document and apply the frozen `Qwen3-Reranker-0.6B` once, obtaining one snippet compatibility score $s_{\text{snip}}(q, c_i)$.

The 11 snippet-derived resolver features consist of this score, its rank, reciprocal rank and score gaps within the candidate pool, its difference from the compact score, source–snippet rank displacement, and the mean, standard deviation, and range of snippet scores for the query. BM25 chunk scores and separate neural scores for individual chunks are used neither as resolver features nor as additional membership predictions.

A.2 LIST-RELATIVE COMPETITION EVIDENCE

Absolute query–candidate compatibility does not determine whether a skill should enter the final set. We therefore describe each candidate relative to the other candidates in the frozen pool $\mathcal{C}_K(q)$.

Score Position and Pool Statistics. For source, compact-card, and snippet scores, the representation includes score gaps to the highest-scoring candidate and, where applicable, to an adjacent candidate in the corresponding ranking. It also includes the query-level mean, standard deviation, and range of the score vectors. These summary statistics are separate features; the implementation does not explicitly z-score candidate values or use a median reference.

Cross-Evidence Disagreement. The representation records signed and absolute rank displacement between source order and compact-card order, and between source order and snippet order. It also

Table 10: **Membership resolver hyperparameters.** The same configuration is used across the main SkillRet set-resolution experiments unless otherwise specified.

Hyperparameter	Value
Resolver	`HistGradientBoostingClassifier`
Objective	Binary log-loss
Learning rate	0.08
Boosting iterations	80
Maximum leaves per tree	31
Minimum samples per leaf	100
ℓ_2 regularization	1.0
Candidate-pool depth K	50
Cross-fitting folds	5
Threshold candidates	81 OOF quantiles + 9 fixed anchors
Maximum output size M	3

includes the snippet-minus-compact score difference. These features expose cases in which the source retriever and query-conditioned evidence favor different candidates.

B Resolver Training and Set-Level Calibration

This appendix provides the implementation details for the membership resolver and the set-level calibration procedure introduced in Section 3. The overall training pipeline separates two objectives: the resolver is learned from candidate-level membership labels, whereas the decision threshold is selected from out-of-fold predictions to optimize set-level ExactSet recovery.

B.1 Resolver Instantiation and Optimization

Given the contextualized evidence vector ϕ_{qi} described in Appendix A, the resolver estimates the membership probability

$$p_{qi} = f_\theta(\phi_{qi}) \approx P(c_i \in G_q \mid \phi_{qi}). \tag{7}$$

In our implementation, f_θ is instantiated as a histogram-based gradient-boosted decision tree using `HistGradientBoostingClassifier`. The resolver is trained with binary cross-entropy over candidate-level membership targets

$$y_{qi} = \mathbb{I}[c_i \in G_q]. \tag{8}$$

Table 10 summarizes the resolver configuration used in all main experiments.

All upstream neural components used to construct candidate evidence remain frozen during resolver training. In particular, the source retriever and the `Qwen3-Reranker-0.6B` pair scorer are not updated jointly with f_θ. The resolver therefore learns only the mapping from the fixed contextualized evidence representation to candidate membership probabilities.

B.2 Grouped Cross-Fitting for Calibration

Directly selecting an admission threshold on predictions from the same examples used to fit the resolver can yield an optimistically biased estimate of set-recovery quality. We therefore obtain calibration predictions through grouped cross-fitting.

Let

$$\mathcal{Q}_{\text{train}} = \{q_1, \ldots, q_N\}, \qquad N = 50{,}474, \tag{9}$$

denote the SkillRet training queries. Queries are grouped according to their gold target-set identity:

$$q_a \sim q_b \quad \Longleftrightarrow \quad G_{q_a} = G_{q_b}. \tag{10}$$

All queries belonging to the same group are assigned to the same fold. We then partition the grouped training set into five folds $\{\mathcal{F}_j\}_{j=1}^{5}$.

For each fold j, a resolver $f_{\theta^{(-j)}}$ is fitted using all training queries outside $\mathcal{F}_j$:

$$\theta^{(-j)} = \mathrm{Train}\left(\mathcal{Q}_{\mathrm{train}} \setminus \mathcal{F}_j\right). \tag{11}$$

The resulting model is then applied only to queries in the held-out fold, producing out-of-fold membership probabilities

$$\hat{p}_{qi} = f_{\theta^{(-j)}}(\phi_{qi}), \qquad q \in \mathcal{F}_j. \tag{12}$$

After all five folds have been processed, every training query has a membership prediction produced by a resolver that was not fitted on that query.

Grouping by target-set identity further prevents queries with an identical gold skill set from being split across the fitting and held-out portions of the same cross-fitting round. This makes threshold selection less dependent on repeated target-set patterns shared across folds.

B.3 ExactSet-Oriented Threshold Selection

We select the set-admission threshold using out-of-fold (OOF) membership predictions from the training queries. For each candidate threshold τ, we apply the same thresholding, maximum-cardinality, and non-empty fallback rules used at inference to obtain $\hat{G}_q(\tau; \hat{\mathbf{p}}_q)$.

Let $\mathcal{P}_{\mathrm{OOF}}$ contain all OOF candidate-membership probabilities. The threshold candidates are the unique values in

$$\mathcal{T} = \{\mathrm{Quantile}_\alpha(\mathcal{P}_{\mathrm{OOF}}) : \alpha \in \mathrm{linspace}(0.80, 0.9999, 81)\} \cup \{0.01, 0.02, 0.05, 0.10, 0.20, 0.30, 0.50, 0.70, 0.90\}. \tag{13}$$

We select

$$\tau^\star = \arg\max_{\tau \in \mathcal{T}} \sum_{q \in \mathcal{Q}_{\mathrm{train}}} \mathbb{I}\left[\hat{G}_q(\tau; \hat{\mathbf{p}}_q) = G_q\right]. \tag{14}$$

Thus, candidate-level membership learning and set-level operating-point selection use distinct objectives.

Ties in ExactSet are resolved, in order, by higher macro Set F1, smaller mean predicted set size, and finally a larger threshold. After selecting $\tau^\star$, we refit the resolver on all training queries. The refitted resolver and $\tau^\star$ remain fixed for test-time evaluation.

B.4 Inference-Time Set Formation

At inference, the final resolver produces membership probabilities $\{p_{qi}\}_{i=1}^{K}$ for the frozen candidate pool. We first admit all candidates satisfying

$$\mathcal{A}_q = \{c_i \in \mathcal{C}_K(q) \mid p_{qi} \geq \tau^\star\}. \tag{15}$$

Two deterministic constraints are subsequently applied.

Maximum Cardinality. If more than $M = 3$ candidates cross the threshold, only the three candidates with the highest membership probabilities are retained:

$$\hat{G}_q = \mathrm{Top}_3\left(\mathcal{A}_q; \mathbf{p}_q\right), \qquad |\mathcal{A}_q| > 3. \tag{16}$$

Otherwise, all admitted candidates are preserved.

Non-Empty Fallback. If no candidate reaches the calibrated threshold,

$$\mathcal{A}_q = \varnothing, \tag{17}$$

we return the candidate with the highest predicted membership probability:

$$\hat{G}_q = \{c_{i^\star}\}, \qquad i^\star = \arg\max_i p_{qi}. \tag{18}$$

The complete decision rule therefore guarantees

$$1 \leq |\hat{G}_q| \leq M. \tag{19}$$

No threshold, cardinality constraint, or resolver hyperparameter is tuned on the test split. All calibration choices are determined from training data before final evaluation.

Algorithm 1 SKILLCRAFT Inference-Time Set Resolution

Require: Query q, ranked pool $\mathcal{C}_K(q)$, resolver f_θ, calibrated threshold τ^*, cardinality cap M
Ensure: Predicted skill set $\hat{G}_q$
{**Stage 1: Contextualized Evidence Construction**}
for each $c_i \in \mathcal{C}_K(q)$ **do**
  Construct $\phi_{qi} \leftarrow \phi_{qi}^{\text{cand}} \oplus \phi_{qi}^{\text{ctx}}$
end for
{**Stage 2: Non-Monotonic Membership Estimation**}
for each $c_i \in \mathcal{C}_K(q)$ **do**
  $p_{qi} \leftarrow f_\theta(\phi_{qi})$
end for
{**Stage 3: Thresholded Set Formation**}
$A_q \leftarrow \{c_i \in \mathcal{C}_K(q) : p_{qi} \geq \tau^*\}$
if $A_q \neq \varnothing$ **then**
  $\hat{G}_q \leftarrow \text{TopM}(A_q; p_q)$
else
  $\hat{G}_q \leftarrow \{\arg\max_{c_i \in \mathcal{C}_K(q)} p_{qi}\}$
end if
return $\hat{G}_q$

C BASELINES AND EVALUATION PROTOCOL

In this section, we provide exhaustive descriptions, model checkpoints, input formats, and implementation configurations for all competing baselines evaluated in Section 4.

C.1 FULL-LIBRARY RETRIEVAL MODELS

All full-library retrieval models are evaluated over the same repository of 6,660 skills. For the generic BM25 and embedding baselines, we index each skill using its available name, description, and `skill_md` or `body` field. This indexed text is distinct from the compact metadata card used for subsequent pair scoring and set selection; specialized models follow their corresponding input conventions.

Lexical Baseline.

- **BM25** (Robertson & Zaragoza, 2009): Evaluated using standard Okapi BM25 with parameters $k_1 = 1.2$ and $b = 0.75$. Queries and skill texts are tokenized using standard English whitespace and punctuation tokenization.

Encoder-Only Bi-Encoders. Dense encoder-only dual-encoders map queries and skills independently into a fixed-dimensional dense vector space and compute relevance via inner-product or cosine similarity:

- **bge-small-en-v1.5 & bge-large-en-v1.5** (Xiao et al., 2023): Standard BERT-based bi-encoders with 33M and 335M parameters, respectively. Following official guidelines, queries are prefixed with `"Represent this sentence for searching relevant passages: "`, while skill metadata texts are encoded without instructions.
- **e5-large-v2** (Wang et al., 2022): A 335M-parameter RoBERTa-based dual-encoder. Queries are prefixed with `"query: "` and skill bodies are prefixed with `"passage: "`.

Decoder-Only Embedding Retrievers. Decoder-only embedding models leverage causal language model backbones with bidirectional attention masks or EOS pooling to produce dense representations:

- **Qwen3-Embed-0.6B** (Zhang et al., 2025): A 0.6B-parameter embedding retriever. We encode queries and skill texts through the checkpoint's SentenceTransformer inference interface and rank skills by normalized embedding similarity.
- **harrier-oss-v1-0.6b** (Microsoft, 2026): An open-source 0.6B retrieval model optimized for multi-task instruction following and semantic retrieval.

- **jina-embed-v5-text-small** (Akram et al., 2026): A lightweight embedding model designed for variable-length semantic text matching with task-specific adapter embeddings.
- **R3-Embedding-0.6B** (Wang et al., 2026): A skill-retrieval model trained with relational reasoning contrastive loss, emphasizing structural and multi-hop API semantics.
- **SkillRouter-Embed-0.6B** (Zheng et al., 2026): A router-oriented bi-encoder model fine-tuned to align user intent with specialized tool execution routes.
- **SkillReason-Embed-0.6B** (Jiang et al., 2026): A reasoning-augmented embedding model incorporating step-by-step rationales into dual-encoder representation spaces.
- **SkillSight** (Xiao et al., 2026): A multi-view representation retriever modeling both hierarchical category trees and operational API schema descriptions.
- **SkillRet-Embedding-0.6B** (Cho et al., 2026): The official bi-encoder retriever accompanying the SkillRet benchmark, trained specifically for multi-tool selection under semantic complementarity.

Retrieve & Rerank Pipelines. Two-stage pipelines retrieve candidates with their corresponding first-stage models and then refine the candidate order with a paired reranker. For the generic Qwen3-Embed-0.6B + Qwen3-Reranker-0.6B baseline, the first stage retrieves 50 candidates; the 0.6B reranker rescores the first 20 using the difference between its "yes" and "no" logits. These 20 candidates are reordered by reranker score, while the remaining 30 retain their first-stage order. The combined pipeline has approximately 1.2B parameters; this is not the size of a single cross-encoder. The R3, SkillRet, SkillRouter, and SkillReason two-stage baselines use their respective paired retrievers and rerankers rather than a shared scoring head or reranking depth.

C.2 Matched-Pool Set Resolution Baselines

All methods in this comparison receive the same frozen top-50 candidate pool from SkillRet-Embedding-0.6B. Only the post-retrieval selection policy varies.

Relative-Score Threshold. This baseline admits candidates whose source retrieval score exceeds a fraction α of the top source score:

$$\hat{S}_q = \{c_i \in \mathcal{C}_{50}(q) : \rho_i \geq \alpha \rho_1\}\,. \tag{20}$$

The operating point is selected on training data. As with the other set-selection policies, the output is capped at three skills.

Prompted LLM Selectors. The prompted selectors receive the query and the 50 source-ordered candidates as numbered compact cards, together with source scores. *SRA-style select-one* is a static adaptation that asks Qwen3-32B to return exactly one numbered candidate. It does not execute SRA's interactive progressive-disclosure procedure or request intermediate tool-requirement reasoning. *Qwen3-32B direct-set selection* instead asks the same model to return one to three candidate indices in a single generation. Their common prompting, parsing, and fallback rules are specified in Appendix C.3.

Pointwise Selector. The Pointwise Selector uses only three raw candidate scores: source retrieval, compact-card compatibility, and snippet compatibility. It uses the same histogram gradient-boosted model class, grouped training protocol, train-only OOF threshold-selection procedure, three-skill cap, and non-empty fallback as SkillCraft. The comparison therefore isolates the added value of contextualized evidence rather than changing the model class or output budget.

C.3 Generative Selector Protocol

This section provides the evaluation protocol for the generative selector comparison in Table 5.

Models. We evaluate Qwen3-32B (Qwen Team, 2025) and DeepSeek-V4.1-Flash (Xu et al., 2026). DeepSeek-V4.1-Flash is a 552B-parameter Mixture-of-Experts model that activates 16B parameters per token during decoding and 8B during prefill. Qwen3-32B is evaluated from the locally hosted checkpoint, while DeepSeek-V4.1-Flash is accessed through the official DeepSeek API using the

`deepseek-flash` model alias. Both models are evaluated over all 4,997 SKILLRET test queries and receive exactly the same frozen top-50 candidate pool.

Candidate Representation. For every query, candidates are presented in the original source-retriever order. Each candidate is identified by a 1-based index and rendered with its source retrieval score together with a compact skill card containing the available skill name, description, tags, and category:

```
[1] retrieval_score=[score]
[compact skill card]
...
[50] retrieval_score=[score]
[compact skill card]
```

Candidate blocks are separated by a fixed delimiter. Skill bodies, query-conditioned snippets, gold labels, and SKILLCRAFT membership scores are not exposed to the generative selectors. Thus, the comparison evaluates prompted selection from the same source-ranked candidate information rather than giving either model access to additional evidence.

Selection Policies. We evaluate two selection interfaces. *Select-one* requires the model to return exactly one candidate and therefore represents a conservative single-skill policy. *Direct set* asks the model to construct the smallest sufficient subset and permits between one and three candidates, matching the maximum target-set cardinality in the benchmark.

For select-one, the model receives the following instruction:

```
Select the single best skill from the candidate list for
solving the problem.  Return exactly one JSON object with
this schema and no other text:
{"selected":  [<one 1-based candidate number>]}.
```

For direct-set prediction, we use:

```
Resolve the problem to the smallest sufficient set of
skills from the candidate list.  Select between one and
three skills.  A skill should be included only when it
is needed to solve the problem; do not add merely related
or redundant skills.  Candidates are ordered by a strong
retriever.  Preserve your preferred order in the returned
list.
Return exactly one JSON object with this schema and no other
text:
{"selected":  [<one to three distinct 1-based candidate
numbers>]}.
```

The corresponding system messages are "`You are a precise skill resolver`" for select-one and "`You are a precise skill-set resolver`" for direct-set prediction, followed in both cases by an instruction to follow the JSON output contract exactly. Each user prompt contains the original query followed by the same 50 rendered candidate cards.

Decoding Configuration. Generation is deterministic for both model families. We use temperature 0, disable explicit thinking, request JSON-object output, and limit the completion to 128 tokens. Each query requires exactly one semantic generation call. Qwen3-32B is served locally, whereas DeepSeek-V4.1-Flash is queried through the official API. Apart from the serving backend, the generation contract and selector-facing input are kept fixed.

Output Parsing and Fallback. We apply the same parser to both model families. Before JSON parsing, optional `<think>...</think>` content and Markdown code fences are removed. The parser then requires a JSON object containing exactly one field, `selected`. For select-one, this field must contain exactly one candidate index; for direct-set prediction, it must contain between one and three indices. All indices must be distinct integers in the valid range $[1, 50]$.

Outputs violating any of these constraints are treated as invalid. In such cases, we deterministically fall back to the source retriever's top-1 candidate. The same parsing and fallback implementation is reused for Qwen3-32B and DeepSeek-V4.1-Flash, ensuring that differences in Table 5 are attributable to the generative selector rather than to model-specific post-processing.

Evaluation. The selected candidate indices are mapped back to skill IDs and evaluated against the gold skill set using the same set-resolution metrics as the main experiment. We report ExactSet, Set F1, Precision, and Recall. No gold labels are available to the selector during generation, and no prompt, decoding parameter, or parsing rule is tuned separately for DeepSeek-V4.1-Flash.

C.4 Downstream Task-Accuracy Evaluation

Tasks and Skill Library. We evaluate all 747 TheoremQA, 223 CHAMP, and 1,100 MedCalc-Bench instances using the 26,262-skill library distributed with SR-Agents. For each query, SkillRet-Embedding-0.6B retrieves a frozen top-50 candidate pool, after which each selection policy constructs the skill context. Neither the selector nor its threshold is retrained or tuned using labels from these downstream tasks.

Executor and Context Construction. All three tasks use the same locally hosted Qwen3-32B checkpoint served through vLLM, with temperature 0, thinking disabled, and at most 2,048 completion tokens. We use the pinned SR-Agents task-specific prompt builder for each benchmark. Selected skill IDs are resolved to their complete `content` fields and supplied to the corresponding prompt builder in selection order. Compact retrieval cards and query-conditioned snippets used during skill selection are not exposed to the executor.

Selection Conditions. We compare five skill-context conditions. *No skill* supplies an empty skill list; *Source top-1* supplies the highest-ranked source candidate; *Relative-score cutoff* uses the pre-specified $\alpha = 0.96$ rule; and SkillCraft uses the frozen learned resolver. The *gold-skill reference* uses the dataset-annotated gold skill IDs and is included only as a reference condition. It is not an upper bound on executor accuracy. An additional `high_full` diagnostic was executed but is omitted from the compact main-text table.

Label Isolation and Scoring. Gold skill annotations are accessed only when constructing the gold-skill reference condition and do not affect the other four selectors. Gold answers and evaluation-only fields are excluded from generation requests and are revealed only after generation for offline scoring. Each benchmark is scored with its corresponding SR-Agents official evaluator, and task accuracy is the fraction of correctly answered instances. Within a task, the question, executor, decoding configuration, prompt builder, and evaluation rule are fixed across skill-selection conditions.

Response Reuse. For TheoremQA, responses are generated independently for each selection condition. For CHAMP and MedCalc-Bench, byte-identical serialized prompts within the same task are executed once and the resulting response is reused across matching conditions. This avoids repeated-call variation when two selectors induce identical executor inputs.

C.5 Evaluation Metrics

ExactSet (Tsoumakas & Katakis, 2007; Cho et al., 2026). ExactSet is our primary set-recovery metric and measures strict agreement between the predicted and gold skill sets:

$$\mathrm{ExactSet}_q = \mathbb{I}\left[\hat{S}_q = S_q^\star\right]. \tag{21}$$

A prediction is correct only when all required skills are recovered and no irrelevant skill is admitted.

Set Precision, Recall, and F1 (Tsoumakas & Katakis, 2007). For each query, we define the counts of true positives (TP_q), false positives (FP_q), and false negatives (FN_q) as:

$$\mathrm{TP}_q = |\hat{S}_q \cap S_q^\star|, \qquad \mathrm{FP}_q = |\hat{S}_q \setminus S_q^\star|, \qquad \mathrm{FN}_q = |S_q^\star \setminus \hat{S}_q|. \tag{22}$$

The set-level Precision (P_q) and Recall (R_q) are computed as:

$$P_q = \frac{\mathrm{TP}_q}{|\hat{S}_q|}, \qquad R_q = \frac{\mathrm{TP}_q}{|S_q^\star|}. \tag{23}$$

If a model predicts an empty set ($\hat{S}_q = \emptyset$), P_q is evaluated as 1 if $S_q^\star = \emptyset$ and 0 otherwise. The query-level Set F1 score is:

$$F1_q = \frac{2P_qR_q}{P_q + R_q} = \frac{2\mathrm{TP}_q}{|\hat{S}_q| + |S_q^\star|}. \tag{24}$$

Jaccard Similarity (Jaccard, 1901). We report query-level Jaccard similarity as the standard intersection-over-union metric for finite sets:

$$J_q = \frac{|\hat{S}_q \cap S_q^\star|}{|\hat{S}_q \cup S_q^\star|}. \tag{25}$$

If both $\hat{S}_q$ and $S_q^\star$ are empty, $J_q = 1$. At the individual query level, Jaccard and Set F1 satisfy $F1_q = \frac{2J_q}{1+J_q}$.

AnyHit. AnyHit is a lenient coverage diagnostic that measures whether the predicted skill set contains at least one gold skill:

$$\mathrm{AnyHit}_q = \mathbb{I}\left[\hat{S}_q \cap S_q^\star \neq \emptyset\right]. \tag{26}$$

Unlike Hit@1, AnyHit is defined over the final predicted set and does not require an intrinsic ordering among selected skills.

NDCG@K (Järvelin & Kekäläinen, 2002; Thakur et al., 2021). Normalized Discounted Cumulative Gain evaluates ranking quality with logarithmic position discounting. For a ranked list $\mathcal{R}_q$, the binary relevance of the skill retrieved at rank k is defined as $\mathrm{rel}_{q,k} = \mathbb{I}[r_{q,k} \in S_q^\star]$. The Discounted Cumulative Gain (DCG) is given by:

$$\mathrm{DCG}_q@K = \sum_{k=1}^{K} \frac{2^{\mathrm{rel}_{q,k}} - 1}{\log_2(k+1)}. \tag{27}$$

Let $\mathrm{IDCG}_q@K$ denote the ideal DCG obtained by placing all relevant skills before non-relevant skills. The normalized metric is:

$$\mathrm{NDCG}_q@K = \frac{\mathrm{DCG}_q@K}{\mathrm{IDCG}_q@K}. \tag{28}$$

Recall@K (Thakur et al., 2021). Ranked Recall quantifies the fraction of annotated gold skills that appear within the top-K candidate positions:

$$\mathrm{Recall}_q^{\mathrm{rank}}@K = \frac{|S_q^\star \cap \{r_{q,1}, \ldots, r_{q,K}\}|}{|S_q^\star|}. \tag{29}$$

This candidate-retrieval metric measures pool coverage and is distinguished from set-level recall evaluated over the final predicted set $\hat{S}_q$ (Thakur et al., 2021; Cho et al., 2026).

Completeness@K (Cho et al., 2026). Completeness is a strict, query-level coverage metric tailored for multi-skill agent tasks:

$$\mathrm{Completeness}_q@K = \mathbb{I}\left[S_q^\star \subseteq \{r_{q,1}, \ldots, r_{q,K}\}\right]. \tag{30}$$

A query is considered complete only when the top-K list contains the *entire* necessary skill combination $S_q^\star$. This characterizes whether the candidate pool is sufficient for downstream execution.

Hit@1 (Bordes et al., 2013; Cho et al., 2026). Hit@1 measures whether the highest-ranked retrieved skill is relevant to the query:

$$\mathrm{Hit@1}_q = \mathbb{I}\big[r_{q,1} \in S_q^\star\big]. \tag{31}$$

We use Hit@1 only for ranking-oriented retrieval evaluation.

C.6 Cardinality Metrics

Cardinality metrics evaluate whether a selection policy recovers the appropriate set size.

Cardinality Accuracy. Cardinality Accuracy evaluates exact size matching between predicted and gold sets:

$$\mathrm{CardAcc} = \frac{1}{N}\sum_{q\in\mathcal{Q}} \mathbb{I}\left[|\hat{S}_q| = |S_q^\star|\right]. \tag{32}$$

Cardinality MAE (Willmott & Matsuura, 2005). Cardinality Mean Absolute Error measures the average deviation in set size:

$$\mathrm{CardMAE} = \frac{1}{N}\sum_{q\in\mathcal{Q}} \left||\hat{S}_q| - |S_q^\star|\right|. \tag{33}$$

Average Predicted Set Size. The average predicted cardinality is defined as $\mathrm{AvgSize} = \frac{1}{N}\sum_{q\in\mathcal{Q}} |\hat{S}_q|$, characterizing the context footprint introduced into downstream agent execution.

C.7 Averaging and Evaluation Protocol

Macro Averaging. All query-level metrics are macro-averaged across the evaluation benchmark:

$$\mathrm{MacroMetric} = \frac{1}{N}\sum_{q\in\mathcal{Q}} \mathrm{Metric}_q. \tag{34}$$

ExactSet, Set F1, AnyHit, NDCG@K, Recall@K, Completeness@K, Precision, Recall, Jaccard, and Hit@1 are computed independently per query before averaging, rather than from globally pooled counts.

Train-Only Selection. All decision thresholds, prefix cutoffs, and calibration parameters are selected exclusively on training splits and kept strictly frozen prior to test evaluation. Test labels are never accessed during tuning.

Matched-Pool vs. Native Pipelines. The matched-pool evaluation operates on a frozen candidate pool produced by SkillRet-Embedding-0.6B (Cho et al., 2026) ($K = 50$) to isolate set-formation policies. Conversely, native retrieve-and-rerank pipelines evaluate candidate discovery using their respective retrieval models, followed by their dedicated set-conversion rules.

C.8 Paired Statistical Testing

We quantify uncertainty using query-level paired comparisons on the 4,997 SkillRet test queries. For each reference method b, we define the paired metric difference as

$$\Delta_m = m(\textsc{SkillCraft}) - m(b), \tag{35}$$

where m denotes either ExactSet or Set F1. Positive values therefore indicate higher performance for SkillCraft.

Paired Bootstrap Confidence Intervals. We estimate 95% confidence intervals using 5,000 paired bootstrap replicates. In each replicate, query IDs are sampled with replacement from the test set, and the same resampled query indices are used for both SkillCraft and the reference method. We recompute the query-macro metric on the resampled queries and record the paired difference Δ_m. The reported confidence interval is given by the 2.5th and 97.5th percentiles of the resulting bootstrap distribution. These intervals capture uncertainty due to the finite evaluation-query sample; they do not measure variation from model retraining or alternative random seeds.

Table 11: **Paired set-recovery comparisons on SKILLRET v1.0.** Differences are SKILLCRAFT minus the reference method in percentage points. Win/Loss and McNemar p apply to ExactSet only.

Reference	ΔExact	95% CI	Win/Loss	McNemar p	ΔF1	95% CI
Matched-pool controls						
Relative-score threshold	+5.28	$[+4.24, +6.34]$	509/245	3.72×10^{-22}	+3.82	$[+3.22, +4.40]$
SRA LLM Selection	+2.36	$[+1.10, +3.58]$	542/424	1.64×10^{-4}	+4.22	$[+3.33, +5.12]$
Qwen3-32B Selection	+28.36	$[+26.76, +29.92]$	1737/320	4.99×10^{-235}	+13.17	$[+12.22, +14.09]$
Pointwise Selector	+2.82	$[+1.72, +3.92]$	465/324	5.83×10^{-7}	+3.04	$[+2.42, +3.64]$
Retrieve-and-rerank references						
SkillRet top-1	+4.28	$[+3.24, +5.34]$	489/275	8.73×10^{-15}	+5.76	$[+5.11, +6.39]$
R3-Skill	+4.02	$[+2.78, +5.28]$	625/424	5.88×10^{-10}	+4.69	$[+3.67, +5.69]$
SkillRouter	+1.56	$[+0.34, +2.80]$	541/463	0.0151	+0.97	$[+0.10, +1.86]$
SkillReason	+2.34	$[+1.14, +3.56]$	545/428	1.97×10^{-4}	+2.49	$[+1.61, +3.39]$

ExactSet Discordance Tests. Because ExactSet is a binary query-level outcome, we additionally summarize each comparison by its discordant pairs. A *win* denotes a query for which SKILLCRAFT recovers the exact gold set while the reference method does not, whereas a *loss* denotes the converse. We apply a two-sided exact McNemar test to these discordant outcomes and report the resulting unadjusted p-value.

Matched-Pool and Native Comparisons. For matched-pool controls, both methods operate on the same frozen top-50 candidate pool, so the paired comparison isolates differences in post-retrieval selection under identical candidate availability. Native retrieve-and-rerank systems are also paired by query ID, but use their original retrieval pipelines and therefore may receive different candidate pools. Their differences should consequently be interpreted as end-to-end performance comparisons rather than isolated estimates of the set-resolution effect.

Complete paired results, including confidence intervals, ExactSet win/loss counts, and McNemar tests, are reported in Table 11.

D CONTROLLED DIAGNOSTICS

D.1 EQUAL-BUDGET ADMISSION

The equal-budget experiment on SKILLROUTER-EVAL-CORE (Zheng et al., 2026) is designed to isolate *where* an additional skill should be admitted from *how many* additional skills a method is allowed to select. It is a controlled diagnostic and does not retrain SKILLCRAFT or modify its primary set-resolution endpoint.

Controlled Second-Skill Admission. For each query, candidates are ordered by the frozen learned membership scores produced by SKILLCRAFT. The first selected skill is fixed to the highest-scoring candidate for every policy. The only optional action is to admit the second-ranked candidate. Thus, all compared policies share the same first prediction and candidate ordering, and differ only in which queries are allocated a second skill.

For a cohort $\mathcal{Q}$, let B denote the number of two-skill predictions produced by the frozen adaptive SKILLCRAFT endpoint. Every competing policy is constrained to admit the second candidate for exactly B queries in the same cohort. This matched-budget construction removes differences in total admission count and isolates the quality of the admission decision itself.

Evaluation Cohorts. We evaluate the Easy and Hard partitions of SKILLROUTER-EVAL-CORE under two complementary views. *All* evaluates the complete cohort, whereas *Size-2* restricts evaluation to queries whose annotated gold set contains exactly two skills. The latter is particularly diagnostic because exact recovery is attainable under the controlled two-skill output protocol: the fixed first candidate and one correctly allocated second candidate can recover the complete gold set.

Admission Policies. Let $c_{q,1}$ and $c_{q,2}$ denote the first two candidates under the frozen learned membership ordering, with membership scores $p_{q,1}$ and $p_{q,2}$. We compare four allocation policies under the same budget B:

- **SKILLCRAFT adaptive.** Uses the second-skill admission decisions produced by the frozen SKILLCRAFT endpoint.
- **Relative-score heuristic.** Allocates the B second-skill admissions to queries with the largest relative second-candidate score $p_{q,2}/p_{q,1}$.
- **Source-rank heuristic.** Prioritizes queries whose learned second candidate $c_{q,2}$ appears earliest in the original source-retriever ranking. Remaining ties are resolved using the learned membership score.
- **Uniform allocation.** Distributes the same B second-skill admissions uniformly across the evaluation cohort. We report the expected cohort-level performance under uniform allocation rather than relying on a single random draw.

For a query-level metric m_q, the expected metric under uniform allocation can be written as

$$\mathbb{E}[\bar{m}] = \bar{m}_{\text{top1}} + \frac{B}{|\mathcal{Q}|}\left(\bar{m}_{\text{top2}} - \bar{m}_{\text{top1}}\right), \tag{36}$$

where $\bar{m}_{\text{top1}}$ and $\bar{m}_{\text{top2}}$ denote the cohort-level metric when every query receives one or two selected skills, respectively.

Candidate-Depth Analysis. To examine whether the admission behavior persists as candidate availability changes, we repeat the matched-budget comparison for $K \in \{5, 10, 20, 30, 50\}$. At each candidate depth, all policies operate on the same truncated candidate pool and are assigned the same number of second-skill admissions. We report ExactSet and Set F1 as query-macro averages. Figure 3 reports the resulting cohort-level comparisons and the sensitivity to candidate depth.

D.2 EVIDENCE ATTRIBUTION AND FEATURE ABLATIONS

We conduct a controlled mechanism audit to further examine which parts of the contextualized evidence representation contribute to set recovery. All variants use the same frozen top-50 candidate pools, precomputed retrieval and compatibility scores, training queries, grouped folds, resolver hyperparameters, cardinality cap, and non-empty fallback rule. For each feature configuration, the resolver is retrained and its admission threshold is selected exclusively from training out-of-fold predictions. No retrieval or neural reranking component is rerun.

Evidence Decomposition. Table 12(a) provides a finer-grained decomposition of the evidence configurations used in the main-text ablation. The raw-score baseline uses only the source, compact-card, and snippet compatibility scores. We then introduce contextualized source/card features, add back the raw snippet score, and finally include the complete snippet-context representation.

Adding the raw snippet score to contextualized source/card evidence produces only a small improvement, from 36.14% to 36.42% ExactSet and from 65.00% to 65.05% Set F1. In contrast, adding the full snippet-context representation increases ExactSet to 37.88% and Set F1 to 66.24%. The ExactSet improvement over the immediately preceding configuration is 1.46 pp, with a query-paired 95% confidence interval of $[+0.68, +2.24]$. This indicates that the benefit of snippet evidence is not explained by exposing an additional scalar compatibility score alone; the larger gain appears when snippet signals are contextualized relative to the candidate pool.

Feature-Group Redundancy. Table 12(b) further removes each contextual feature group from the full representation while retaining the underlying raw compatibility scores. Each ablated variant is retrained from scratch and recalibrated using training out-of-fold predictions. No individual removal consistently reduces both ExactSet and Set F1: some variants slightly increase ExactSet while decreasing Set F1, whereas others exhibit the opposite pattern. We therefore do not attribute the full-model improvement to any single feature family. Rather, the results suggest substantial redundancy and interaction among rank-relative, cross-evidence, and list-context signals.

Table 12: **Detailed evidence ablations on SKILLRET.** All variants share the same frozen candidate pool and training protocol. **(a)** decomposes raw and contextualized evidence. **(b)** removes one contextual feature group at a time from the full representation; parenthesized values denote changes relative to the full model.

(a) Evidence decomposition

Configuration	Raw Snippet	Source/Card Context	Snippet Context	ExactSet	Set F1	Avg. Size
Raw-score baseline	✓	–	–	35.06	63.21	–
Source/card contextualization	–	✓	–	36.14	65.00	–
+ Raw snippet score	✓	✓	–	36.42	65.05	–
Full contextualized evidence	✓	✓	✓	**37.88**	**66.24**	1.325

(b) Leave-one-group-out from full contextualized evidence

Configuration	ExactSet	Change	Set F1	Change
Full contextualized evidence	37.88	–	66.24	–
w/o source rank/margin features	38.10	+0.22	65.98	-0.26
w/o compact rank/margin features	37.76	-0.12	66.20	-0.04
w/o snippet rank/margin features	37.90	+0.02	66.13	-0.11
w/o cross-evidence disagreement	37.86	-0.02	66.27	+0.03
w/o query/list statistics	38.14	+0.26	66.16	-0.08

Table 13: **Cardinality-specific effect of snippet contextualization.** Both configurations share the same remaining 19-dimensional evidence representation; the latter augments the scalar snippet signal with snippet-relative contextual features.

Gold Size	n	Scalar Snippet Evidence	+ Snippet Context	Change
1	2,319	65.16	62.48	-2.67
2	2,006	15.35	21.49	**+6.13**
3	672	0.15	1.93	+1.79
All	4,997	36.42	**37.88**	**+1.46**

Threshold Selection. The train-only selected admission threshold of the full contextualized model is exactly 0.5. Therefore, the improvement of the full endpoint over the raw-score baseline cannot be explained by shifting the decision threshold. The measured difference instead reflects changes in the learned membership scores induced by the richer evidence representation.

Cardinality-Specific Effect. The improvement is not uniform across gold-set sizes. Relative to the configuration with source/card context and the raw snippet score, the full contextualized representation improves ExactSet by 6.13 pp on two-skill queries while decreasing it by 2.67 pp on single-skill queries. The aggregate improvement therefore comes primarily from stronger recovery of multi-skill sets, consistent with the role of contextual evidence in resolving interleaved candidates.