Agentic Skill Router - All Experiments about Retriever

Experiment dates: 2026-05-23 to 2026-05-27 Data: SkillRouter eval-core (arXiv:2603.22455) trimmed anonymized corpus, plus the paper's original Easy 78,361 pool Scope: Claude Code paired/fresh-env experiments, Codex A-M strategy experiments, J/L/M scaling experiments, alignment with the paper's original Hard pool, and SkillRouter 75 core x Easy x multi-skill alignment experiments

---

Executive Summary

`skill-router` is designed to move low-frequency skills out of the host agent's always-on context and route them back from a disabled-skill corpus when needed. This report evaluates accuracy, cost, and scalability across several rounds of strategy iteration.

Key conclusions:

1. **On the 78K Easy x 75 core paper-alignment experiment, the agentic metadata-only router beats the strongest published `nd` baseline.** The best cell, codex/J-bounded-v2, reaches Hit@1 = 40.0%, which is +9.3pp above Qwen3-Emb-8B `nd` (30.7%) and also above BM25 with full body (34.7%). All 6/6 cells beat BM25 `nd`, and 4/6 beat Qwen3-Emb-0.6B `nd`. See section 9.
2. **Host/model choice matters more than variant choice.** Under the same SKILL.md template, Codex (GPT-5.5) averages 36.4%, while Claude Code (Opus 4.7) averages 27.1%, a 9.3pp gap. The largest gap between the three variants is only 7.3pp. The 150-skill fresh-env rerun shows the same pattern: Codex router aggregate is 321/360 (89.2%), Claude Code with-CLAUDE.md router aggregate is 301/360 (83.6%); Codex native is 23/24, while Claude Code native is 14/24.
3. **52% of queries form a structural ceiling for metadata-only routing.** Among 75 queries, 39 miss in all 6 cells, and 7 show the same wrong selection across variants and hosts. Breaking through requires body-on-tie.
4. **Scaling from 150 skills to 1K is still stable, but 79K Hard drops sharply.** L-agentic from 150 to 1K loses only one cell, but M-bm25 on 79K Hard paper-core single reaches 14/24 and J-v2 reaches 11/24 strict. Metadata-only cannot be the sole decision source.
5. **For Claude Code, trigger noise is the first governance issue, but not the only difference after the fresh-env rerun.** In the early paired experiment, CLAUDE.md raised trigger from 78% to 97.6% and accuracy from 69% to 86.9%. In the new 16-variant fresh-env rerun, with-CLAUDE.md trigger already reaches 96.9%, but still trails the Codex router aggregate by 5.6pp. The remaining gap mainly comes from host/model metadata reranking, output contract behavior, and context baseline.

Secondary conclusions: A-router's fixed scorer is not acceptable (Claude Code fresh-env 10/24, Codex new-CLI rerun 11/24); paper comparisons must be made dimension by dimension because the paper does not publish the Single x nd x Hard-only cell; in the Codex 150-skill new-CLI rerun, C-lite and L-agentic reach 24/24 while most strong strategies sit near 23/24; the latest Claude Code L-agentic restore rerun is 22/24, so L is viable on Claude but less stable than on Codex.

This report evaluates the routing layer only. It does not measure downstream task success after the matched skill is loaded. All rankings are single-run strict-match routing-only scores and include ambiguous samples by default.

Background

Recent work on agentic retrieval points to the same engineering judgment: retrieval should not be treated as a one-shot top-k black box. It works better when exposed as a tool interface that the agent can control, iterate, and inspect. This report follows that direction by expressing skill routing as corpus/search/inspect tools, then measuring how different hosts and strategies trade accuracy, context, and cost.

Paper	Relationship to this report
Beyond Semantic Similarity: Rethinking Retrieval for Agentic Search via Direct Corpus Interaction	Reframes retrieval as direct corpus interaction by the agent and argues that search/read/script-style corpus operations fit multi-step reasoning better than fixed embedding top-k.
Is Grep All You Need? How Agent Harnesses Reshape Agentic Search	Shows that retrieval results depend not only on the ranker, but also on the agent harness, tool-output shape, and interactivity; lexical methods remain competitive under the right harness.
AgenticRAG: Agentic Retrieval for Enterprise Knowledge Bases	Focuses on wrapping enterprise search systems as agent-callable tools, allowing models to iteratively gather evidence through search/find/open/summarize actions.
Rethinking Agentic Search with Pi-Serini: Is Lexical Retrieval Sufficient?	Re-evaluates BM25/lexical retrieval inside an agentic loop and highlights retrieval depth, browse/read control, and agent tool loops.
SkillRouter: Skill Routing for LLM Agents at Scale	Provides this report's data and benchmark framing: find the right skill from a large catalog instead of placing every skill body into the prompt.

The common thread is that giving agents better, more controllable retrieval tools can improve outcomes. This report applies that idea to disabled-skill routing and compares native routing, bash-based retrieval, tool-wrapped agentic retrieval, and large-scale BM25 retrieval.

---

1. Overview

1.1 Goal

When no enabled skill clearly covers a user request, the agent must choose the most relevant disabled skill. This experiment answers four questions:

- **RQ1 Accuracy:** Which implementation most reliably selects the ground-truth skill?
- **RQ2 Cost:** What are the tradeoffs across tokens, tool calls, wall time, and dollar cost?
- **RQ3 Mechanism:** What are the boundaries of hand-written bash retrieval, tool-wrapped agentic retrievers, BM25 rankers, and host-native skill selection?
- **RQ4 Scalability:** Can strategy rankings from 150 skills extrapolate to 1K synthetic, 79K Hard, or 78K Easy pools?

1.2 Strategy Families

Category	Variants	Core implementation
Native baseline	G-native	No explicit router; the host model selects directly from the native skill listing.
Bash-based retrieval	B-cc, C-lite, H-bounded, I-meta, J-bounded, J-bounded-v2, K-bounded, K-lite	The agent uses shell primitives such as `find`, `grep`, `sed`, and `awk` over disabled `SKILL.md` files. Later variants add output budgets, metadata-only constraints, frontmatter tie-breaks, and more stable keyword behavior.
Tool-wrapped agentic retrieval	A-router, D-agentic, D-agentic-metadata, E-digest, L-agentic	Retrieval is exposed through stable CLI/tool primitives: one-shot `skills route`, DCI `search/inspect`, `skill-corpus catalog/show`, or a `corpus search/inspect` loop. The model mainly rewrites the query, compares candidates, and makes the final selection.
Large-scale BM25 retrieval	M-bm25	BM25 provides a metadata shortlist for large corpora; the model then reranks candidate metadata. This mainly targets the 78K/79K setting.

1.3 Metrics

- **Accuracy / Hit@1:** The output skill strictly matches the expected skill. In the 78K multi-skill experiment, Hit@1 follows the paper's any-ground-truth definition.
- **Trigger:** Whether the router workflow was actually invoked. Not applicable to G-native.
- **Turns / duration / cost:** Model turns, end-to-end wall time, and estimated or reported cost.
- **ctxStart / ctxEnd / growth:** Context-window size at the first and last model calls, and the growth between them.
- **Tool calls / bash errors / cache hit:** Operational complexity, CLI reliability, and cache economy signals.
- **Failure type / miss distribution:** Distinguishes high-quality distractor selections, hallucinated ids, no-match, and formatting errors.

1.4 Datasets

- **24 queries / 150 skills:** Main controlled comparison for Claude Code and Codex across 16 strategies.
- **1K synthetic:** The 150-skill comparison corpus plus 850 synthetic noise skills, run only for Codex L-agentic to test stability as the candidate count grows.
- **79K Hard:** Original SkillRouter Hard pool, used to test extrapolation boundaries for metadata-only strategies. The report keeps this analysis even though the HTML main view de-emphasizes the standalone section.
- **78K Easy / 75 core:** The direct SkillRouter paper `nd` alignment experiment: K/J-v2/M x Claude/Codex, 6 cells total.

---

2. Data and Corpus

2.1 Corpus Overview

This report uses three increasing corpus scales, all derived from SkillRouter `eval-core` (arXiv:2603.22455):

Layer	Size	Source	Sections	Installation shape
150-skill comparison	150	19 gt + 80 targeted distractors + 51 easy noise, deterministic sampling	sections 6-7	Full SKILL.md (metadata + body)
78K Easy pool	78,361	Full `eval_core/easy/*.jsonl.gz`	section 9	metadata-only (name + description, body stripped)
79K Hard pool	79,141	`eval_core/hard/*.jsonl.gz` (Easy + 780 LLM distractors)	section 8	metadata-only

The three layers share 19 ground-truth skills but differ in pool size and distractor composition. The 150-skill corpus includes only 80 targeted distractors and 51 easy noise skills. The 78K Easy corpus is the paper's full easy pool. The 79K Hard corpus adds 780 high-quality distractors distilled with GPT-4o-mini.

2.2 150-Skill Corpus

Component	Count	Notes
Ground-truth skills	19	GT entities for 24 single-skill tasks; some tasks share the same GT.
Targeted distractors	80	Same-domain near-synonym distractors generated by SkillRouter with GPT-4o-mini.
Noise skills	51	Deterministic random sample from the easy pool.
Total	150	Size of the controlled comparison corpus.

The first corpus version kept `gt-*` / `distractor-*` directory prefixes, which leaked the answer. After the fix, all directory names and frontmatter `name` values are anonymized to `skill-001` through `skill-150` after deterministic shuffling. The mapping exists only in `corpus-manifest.json` for offline analysis and is not exposed to the agent.

The 78K Easy and 79K Hard installations use deterministic shuffled `sr-XXXXX` ids (seed 20260525). Frontmatter `name:` and `description:` keep the upstream strings because the paper's `nd` input includes the name signal. Directory names are anonymized to avoid leaking `gt/` or `distractor/` prefixes.

2.3 Query Set

The 24 queries are directly taken from SkillsBench single-skill task `instruction_text`. These are complete task descriptions, often with file paths, output formats, and constraints. They are not short keyword queries written for this experiment. The full list is in `queries.json`.

The later paper-core single alignment experiment switches to the 24 rows in the SkillRouter paper's 75 core queries where `core_gt_ids.length == 1`. Only 16 overlap with the original `queries.json` set. The initial set contains several generic file-type tasks, while the paper-core single set includes tasks such as `flink-query`, `invoice-fraud-detection`, `latex-formula-extraction`, `manufacturing-*`, `paper-anonymizer`, `simpo-code-reproduction`, and `xlsx-recover-data`.

Two data issues matter:

- The `gh-repo-analytics` GT label is debatable: the GT skill is tool-centric, while several routers choose a task-centric distractor that better matches the query semantics.
- `skill-105` is a generic Excel skill shared by 5 queries, amplifying the effect of spreadsheet description quality on total score.

---

3. Variants

3.1 Implementation Categories

Category	Variant	Routing method
Native baseline	G-native	No router is loaded; the corpus is fully enabled and the host's native skill-selection mechanism chooses.
Bash-based retrieval	B-cc	The agent freely uses shell search over the disabled-skill directory, closest to DCI-Agent-CC's free shell.
Bash-based retrieval	C-lite	Bash only, with bounded grep/sed pipelines and local reads to limit output size.
Bash-based retrieval	H-bounded	Similar to B, but restricts bare `ls` and corpus output to reduce prompt blowup.
Bash-based retrieval	I-meta	Reads only description/frontmatter metadata and forbids reading bodies.
Bash-based retrieval	J-bounded	Extracts 3-5 keywords, then greps descriptions for a shortlist.
Bash-based retrieval	J-bounded-v2	Scale-stable J fix: uses `find -print0
Bash-based retrieval	K-bounded	Stronger J: after grep shortlist, the model can inspect candidate frontmatter for metadata tie-breaks.
Bash-based retrieval	K-lite / K-lite fixed	Reduces K's shell and tool cost; fixed version repairs an accuracy regression.
Tool-wrapped agentic retrieval	A-router	Calls `skill-router skills route`; the CLI performs metadata / DCI / lexical cascade scoring and commits directly.
Tool-wrapped agentic retrieval	D-agentic / D-agentic-metadata	Structured DCI `search/inspect` loop; metadata-only version forbids reading skill bodies.
Tool-wrapped agentic retrieval	E-digest	Two-step `skill-corpus catalog/show` wrapper that supplies catalog plus a few candidate bodies/metadata.
Tool-wrapped agentic retrieval	L-agentic	`corpus search/inspect` abstraction; the CLI handles indexing, pagination, candidate refs, and JSON schema while the model iterates queries and compares evidence.
Large-scale BM25 retrieval	M-bm25	L's BM25-ranker variant: metadata BM25 shortlist plus Codex/Claude metadata rerank, mainly for 78K/79K scale.

3.2 Design Tradeoffs

- **Native baseline** measures the host's built-in skill listing quality; it is not a production router.
- **Bash-based retrieval** is transparent and dependency-free, but sensitive to shell globbing, output truncation, path parsing, and tool-call count. These risks grow at 79K scale.
- **Tool-wrapped agentic retrieval** turns retrieval into a stable tool contract and reduces the model's burden of writing shell. L-agentic's 1K result supports this direction, but 79K Hard still shows that metadata-only decisions are insufficient.
- **Large-scale BM25 retrieval** gives a low-cost shortlist for large corpora, but BM25 order is evidence, not a replacement for final rerank. M-bm25 dropping to 19/24 on the new 150 run is one counterexample.

Router variants use `variants/routing-only/<variant>.SKILL.md`. All router variants share the same frontmatter; differences are only in the body workflow.

---

4. Experiment Design

4.1 Routing-Only Setup

Each query has `STOP_TAIL` appended and asks the agent to output only:

```
{"matched_skill_name":"<skill-id>"}
```

Then it stops. This isolates routing and avoids downstream task execution quality affecting routing metrics.

4.2 Claude Code Paired Experiment

The Claude Code portion uses `routing-only-paired.mjs`:

- 9 variants x 24 queries x 2 conditions = 432 cells.
- Conditions: with-CLAUDE.md and without-CLAUDE.md.
- Each `(variant, condition)` uses an isolated `$HOME`.
- G-native is the exception: the full corpus is enabled and no router plugin is loaded.
- Router variants: the corpus is fully disabled and `skill-router-skills` is loaded.
- Each cell has a 240s timeout.

with-CLAUDE.md contains:

```
# Skill routing

`skill-router-skills` is a routing Skill that searches a catalog of
locally-installed disabled skills.

When no enabled Skill clearly matches the user's query, you must call
`skill-router-skills` before answering. Do not invent a Skill name or
fabricate `matched_skill_name` without a Skill/tool result.
```

Before the full paired experiment, a 64-cell gate probe tested whether this CLAUDE.md text raises trigger. The pre-declared gate criteria were:

```
PASS = trigger_rate(with) >= 95%
     AND trigger_lift >= +20pp
     AND direct_match_preservation(with) >= 90%
     AND no degradation > 10pp vs without
```

The probe passed: fallback trigger lift +62.5pp, direct-match preservation 100%.

After 2026-05-26, a fresh-env 16-variant matrix was added for a more direct host comparison:

- Claude Code: 16 variants x 24 queries x 2 conditions = 768 cells.
- Codex: 16 variants x 24 queries = 384 cells. The official Codex number uses `rerun-codex-newcli-full-2026-05-26`.
- Every `(host, variant, condition, query)` uses a fresh isolated HOME/project to avoid cross-cell state.
- This fresh-env set is used for sections 6.5 and 7.6; the early paired result remains the source for the trigger-policy A/B conclusion.

4.3 Codex 16x24 New-CLI Experiment

The Codex part uses the same 24 queries, same 150-skill corpus, and same routing-only output schema. It is not a secondary appendix to Claude Code; it is a peer host experiment used to compare the same routing strategies across hosts. After 2026-05-26, main results use the new CLI semantics and fresh-env rerun:

- `codex exec`
- model `gpt-5.5`
- `reasoning_effort=high`
- 16 variants x 24 queries = 384 cells
- fresh Codex home/project for each `(variant, query)`

The Codex router adapts host paths from `.claude/skills` to `.codex/skills`, uses the same `.agentic-skill-router-disabled` suffix, and normalizes skill id prefixes. CLI-based strategies such as A-router, D-agentic, L-agentic, and M-bm25 keep their semantics while sharing the unified CLI conventions.

4.4 Scaling Experiments

Scaling tests check whether conclusions from 150 skills extrapolate:

- 150-skill: anonymized controlled comparison corpus.
- 1K synthetic corpus: 150-skill corpus plus 850 synthetic noise skills, a low-cost check for index and prompt scaling.
- 79,141-candidate Hard pool: SkillRouter eval-core Hard tier (section 8, variants J-bounded-v2 and M-bm25).
- 78,361-candidate Easy pool: SkillRouter eval-core Easy tier (section 9, variants K-bounded, J-bounded-v2, M-bm25, 75 core queries).

The 79K original Hard pool uses opaque mappings from the SkillRouter paper's original skill IDs. Candidate ids visible to the agent are `sr-*`; metadata keeps original name/description, so it is not fully anonymized, but it does not leak `gt/` or `distractor/` directory prefixes.

Paper comparison uses two different definitions:

- The paper's `nd` metric means name + description only and is reported on 75 core queries over Easy / Hard / Avg. It is not split into single-only.
- The paper's `Single Hit@1` metric is based on 24 single queries but belongs to the full-skill-text main result, not `nd`.

Therefore this report's `paper-core single x Hard x metadata-only` rerun fills a supplemental cell and cannot be treated as the same denominator as published tables.

---

5. Metrics

The experiment tracks 8 core metrics, 5 supplemental diagnostic metrics, and 1 failure-classification metric. The calculation logic is centralized in `experiments/dci-compare/extract-metrics.mjs`.

5.1 Core Metrics

Metric	Claude source	Codex source	Meaning
accuracy	normalized `matched_skill_name` strictly equals expected skill	same	Core success rate across variants
triggerRate	any `assistant.tool_use.name == "Skill"` and `input.skill` matches `/skill-router-skills/i`	any `command_execution.command` references `/skills/skill-router-skills/SKILL.md`	Whether the router was actually invoked. Not applicable to G-native
totalTurns / avgTurns	`result.num_turns` (official Claude count)	number of rollout `event_msg/token_count` events, i.e. internal Responses API calls	Model turns
ctxStart	first deduped `assistant.message.usage` as `input_tokens + cache_creation_input_tokens + cache_read_input_tokens`	first rollout `event_msg/token_count.info.last_token_usage.input_tokens`; OpenAI convention includes cached tokens	Prompt size at first model call
ctxEnd	last deduped `assistant.message.usage` as `(input + cache_create + cache_read)`. This is not `result.usage`, which is cumulative across turns	last rollout `event_msg/token_count.info.last_token_usage.input_tokens`	Prompt size at final model call
ctxGrowth	`ctxEnd - ctxStart`	same	Context growth during the session
cost	`result.total_cost_usd` (Anthropic reported bill, cache discount included)	`(fresh x $5 + cached x $0.5 + output x $30) / 1M`, using gpt-5.5 list price from 2026-05-24; `fresh = cumInput - cumCached`, from rollout `totalTokenUsage`	Claude is real billing; Codex is list-price estimate, good for Codex-internal comparison but not strict cross-host billing
duration	`result.duration_ms` (Claude reported wall time)	runner's `Date.now() - t0`, since Codex stream-json does not report duration	End-to-end wall time

5.2 Supplemental Metrics

Metric	Meaning
toolCallCount	Deduped `tool_use` total for Claude or `command_execution` total for Codex. Complements `numTurns`: one turn may contain multiple tool calls and vice versa. The main report keeps totals only.
bashErrorCount	Count of bash calls where `command_execution.exit_code != 0` (Codex) or `tool_result.is_error == true` / `tool_use_result.success == false` (Claude). Proxy for agent operation blockage.
cacheHitRatio	Claude: `cumCacheRead / (cumInput + cumCacheRead + cumCacheCreate)`; Codex: `cumCached / cumInput`. Independent token-economy signal.
outputTextLen	Character length of final `result.result` / `agent_message.text`. Short outputs are usually compliant JSON; long outputs often accompany hallucination.
failureType	Miss classification: `distractor` (another legal pool skill), `hallucinated` (out-of-pool id or missing name), `no_match`, or `format_error`. Captures why a selection failed.

5.3 Important Corrections

- **Old `ctx_end` was wrong.** Earlier drafts used `result.usage.cache_read + cache_create + input` as `ctx_end`, but Claude's `result.usage` is cumulative across all turns and inflates multi-turn variants. The new definition uses only the last assistant message usage, reflecting the actual window seen by the final model call. Similarly, Codex `turn.completed.usage.input_tokens` is cumulative across internal calls; true `ctx_end` must come from `event_msg/token_count.info.last_token_usage.input_tokens`.
- **Codex stream-json lacks duration.** `codex exec --json` does not report duration, so the runner measures it.
- **Claude's streaming protocol duplicates messages.** The same `message.id` may appear multiple times because of thinking/tool_use/text chunks. Usage is deduped by `message.id`; content processing reads all chunks.

---

6. Claude Code with 24 Queries / 150 Skills

Claude Code separates with-CLAUDE.md from without-CLAUDE.md because the former explicitly asks the agent to call `skill-router-skills` before answering when no enabled skill matches, while the latter keeps default host behavior. This A/B design separates retriever selection quality from trigger noise.

The table below is the strict-match result under with-CLAUDE.md. D-agentic uses the rerun after the format fix; its absolute value is useful, but it is excluded from the paired aggregate.

6.1 Accuracy with CLAUDE.md

rank	variant	corpus	accuracy	trigger	hallu	cost	duration	turns	avg ctx_end	Notes
1	L-agentic v3	150	24/24 (100%)	24/24	0/24	$4.05	608s	101	35.3K	OR-only + forced inline evaluation of top-3 without inspect; only 24/24 on Claude
2	K-bounded	150	23/24 (95.8%)	24/24	0/24	$3.27	547s	102	30.7K	grep shortlist + candidate metadata; low-cost Pareto point for small/medium pools
2	K-lite	150	23/24 (95.8%)	24/24	0/24	$3.45	569s	101	31.1K	Slimmed K prompt, same accuracy
2	L-agentic v3.1	150	23/24 (95.8%)	24/24	0/24	$3.47	470s	96	32.6K	v3 + `--limit 5`; drops from 24 to 23 cells, see section 6.5
2	C-lite	150	23/24 (95.8%)	24/24	0/24	$4.30	654s	156	32.7K	bounded grep/sed with local body reads
2	B-cc	150	23/24 (95.8%)	24/24	0/24	$5.56	896s	197	34.3K	free-shell DCI, reads body
7	J-bounded	150	22/24 (91.7%)	23/24	1/24	$3.07	374s	96	30.7K	metadata-only bounded
7	L-agentic v1	150 original	22/24 (91.7%)	24/24	0/24	$3.50	596s	102	32.4K	CLI corpus search/inspect, must-AND + conditional inspect
7	L-agentic v1	150 rerun	22/24 (91.7%)	24/24	0/24	$3.51	494s	101	32.4K	Same corpus rerun; total stable, miss set not identical
7	D-agentic (read-body)	150	22/24 (91.7%)	23/24	1/24	$3.65	467s	100	34.1K	paired fixed rerun; reads body
7	K-bounded	1K syn	22/24 (91.7%)	24/24	0/24	$3.74	890s	116	31.2K	1K loses one cell; shell glob does not hit ARG_MAX
7	E-digest	150	22/24 (91.7%)	24/24	0/24	$4.11	355s	96	38.0K	pulls catalog descriptions before choosing
7	H-bounded	150	22/24 (91.7%)	23/24	1/24	$4.16	542s	132	32.9K	like B, but limits bare ls and output
14	L-agentic v1	1K syn	21/24 (87.5%)	24/24	0/24	$3.31	499s	100	31.1K	only one cell below 150; ctx_end barely grows
14	M-bm25	150	21/24 (87.5%)	24/24	0/24	$3.62	537s	99	32.5K	BM25 shortlist + Claude metadata rerank
14	I-meta	150	21/24 (87.5%)	23/24	1/24	$4.16	404s	93	38.0K	description-only, body forbidden
17	D-agentic (metadata-only)	150	20/24 (83.3%)	24/24	0/24	$3.68	476s	100	33.8K	c113b38 metadata-only change regresses
18	G-native	150	15/24 (62.5%)	n/a	0/24	$2.98	130s	24	36.5K	no router; full corpus enabled
19	A-router	150	13/24 (54.2%)	23/24	1/24	$3.90	499s	96	35.3K	fixed lexical/cascade scorer

Interpretation:

- B-cc and C-lite are among the highest-accuracy groups on Claude Code at 150 skills.
- D/E/H/J are only one cell lower; at n=24, this should not be over-read as a stable difference.
- A-router is the only router below G-native.

- G-native at 15/24 shows that Claude Code native metadata selection is insufficient for near-synonym distractors in this corpus.

6.2 Cost and Context with CLAUDE.md

variant	accuracy	cost	duration	turns	avg ctx_end
G-native	15/24	$2.98	130s	24	36.5K
J-bounded	22/24	$3.07	374s	96	30.7K
D-agentic	22/24	$3.65	467s	100	34.1K
A-router	13/24	$3.90	499s	96	35.3K
E-digest	22/24	$4.11	355s	96	38.0K
H-bounded	22/24	$4.16	542s	132	32.9K
I-meta	21/24	$4.16	404s	93	38.0K
C-lite	23/24	$4.30	654s	156	32.7K
B-cc	23/24	$5.56	896s	197	34.3K

Pareto view:

- **New Pareto point K-bounded:** pushes the high-accuracy cost floor from J-bounded's $3.07 (22/24) and C-lite's $4.30 (23/24) to $3.27 (23/24). K-lite has the same accuracy but costs $0.18 more.
- J-bounded remains the low-cost 22/24 option. K-bounded buys one additional cell for $0.20 more and is preferable when accuracy matters more.
- C-lite is the read-body 23/24 / $4.30 representative. B-cc has the same accuracy but is 41% more expensive and 64% slower.
- **1K scaling K vs L:** K-bounded at 1K is 22/24, losing one cell; L-agentic at 1K is 21/24, also losing one cell. K is one cell better but 890s vs L's 499s, 78% slower. K's shell glob does not hit `ARG_MAX` at 1K because 1000 directories are far below the roughly 13K limit, but larger pools remain untested. **Use K for small/medium pools (<=1K) by default and L as the safer large-pool option.**
- **L-agentic 150 rerun variance:** both 150 runs are 22/24, but miss sets differ partly. The original misses `shock-analysis-demand`; the rerun misses `econ-detrending-correlation`; `gh-repo-analytics` is shared. This supports the limitation that single-run N=1 cells have variance near ambiguous boundaries.
- D-agentic metadata-only (20/24) is clearly lower than the read-body D-agentic in the paired run (22/24), a body-on-tie signal for Claude Code.
- G-native is cheap but insufficiently accurate for this corpus.

6.3 with-CLAUDE.md vs without-CLAUDE.md

variant	acc with	acc without	trigger with	trigger without
A-router	13/24	10/24	23/24	20/24
B-cc	23/24	19/24	24/24	19/24
C-lite	23/24	16/24	24/24	16/24
D-agentic (fixed rerun)	22/24	19/24	23/24	20/24
E-digest	22/24	17/24	24/24	19/24
H-bounded	22/24	19/24	23/24	19/24
I-meta	21/24	17/24	23/24	19/24
J-bounded	22/24	18/24	23/24	19/24
paired aggregate, excluding D-agentic	**146/168**	**116/168**	**164/168**	**131/168**

G-native is 15/24 in both arms. Because G-native does not load `skill-router-skills`, CLAUDE.md is effectively a no-op for it. This supports the conclusion that the with-arm improvement mainly comes from trigger behavior, not random cross-run noise. However, each cell is still a single run and should not be treated as statistically significant. D-agentic is kept as a post-fix reference row but excluded from the aggregate because it comes from a separate rerun. Later strategy iterations (D metadata-only / K / L / M) only ran the with-CLAUDE.md arm and do not have paired without-arm data.

6.4 Failure Composition

Under with-CLAUDE.md, no variant selects random noise skills. Errors mostly fall into two categories:

- **Targeted distractors:** G-native 9 times, A-router 10 times, other strong routers 1-2 times.
- **Residual hallucination:** router aggregate 5/192, mostly on strong keyword queries such as `.xlsx`, BibTeX, and GitHub analytics.

This confirms that SkillRouter's targeted distractors are the main challenge; random easy noise is not the dominant error source.

6.5 Fresh-Env 16-Variant Rerun and Claude Diagnostics

The 2026-05-26 fresh-env rerun expands Claude Code to the same 16 variants used by Codex. Every cell uses a new HOME and project, so this run is more appropriate for cross-host attribution. It does not replace the paired trigger-policy A/B result from sections 6.1-6.4.

with-CLAUDE.md main result:

variant	accuracy	trigger	turns total/avg	avg ctx_end	growth	tools	bash err	cost
K-lite	22/24	23/24	97 / 4.0	30.3K	+4.0K	50	0	$4.06
E-digest	22/24	23/24	93 / 3.9	36.8K	+10.7K	46	0	$4.80
J-bounded-v2	22/24	23/24	132 / 5.5	30.5K	+4.3K	85	8	$4.83
H-bounded	22/24	23/24	130 / 5.4	32.1K	+6.2K	83	0	$5.03
M-bm25	22/24	24/24	113 / 4.7	34.6K	+8.5K	65	0	$5.36
K-lite-replicate	21/24	23/24	97 / 4.0	30.2K	+4.1K	50	0	$4.10
D-agentic	21/24	24/24	112 / 4.7	35.4K	+9.3K	64	2	$4.85
I-meta	21/24	24/24	96 / 4.0	37.7K	+11.7K	48	0	$5.17
K-bounded	20/24	23/24	98 / 4.1	29.9K	+3.6K	51	0	$3.82
C-lite	20/24	21/24	126 / 5.3	30.8K	+5.0K	81	1	$4.77
J-bounded	20/24	23/24	153 / 6.4	34.9K	+8.8K	106	0	$6.08
B-cc	20/24	23/24	177 / 7.4	33.5K	+7.7K	130	0	$6.13
D-agentic-metadata	19/24	24/24	101 / 4.2	34.5K	+8.5K	53	1	$4.80
L-agentic	19/24	24/24	110 / 4.6	33.9K	+7.4K	62	2	$4.98
G-native	14/24	n/a	24 / 1.0	34.1K	+0.0K	0	0	$2.82
A-router	10/24	24/24	101 / 4.2	35.8K	+10.1K	53	2	$4.80

Router aggregate for this batch:

condition	accuracy	trigger	cost
with-CLAUDE.md	301/360 (83.6%)	349/360 (96.9%)	$73.59
without-CLAUDE.md	297/360 (82.5%)	339/360 (94.2%)	$73.43

Interpretation:

- In the fresh-env rerun, CLAUDE.md still improves trigger, but less than in the early paired gate because the new router descriptions themselves already strongly instruct the agent to call the router. The without arm reaches 94.2% trigger. Remaining differences are therefore not primarily trigger; they come from candidate evidence comparison and output contract.
- Claude Code's 16-variant ceiling is currently 22/24; it does not show the 24/24 group that Codex does. K-lite is the lowest-cost point in the 22/24 group; M-bm25 also reaches 22/24 with no bash error, suggesting BM25 shortlists help Claude stability.
- Claude Code has a local advantage in bash-based variants: fewer tool calls. Across B/C/H/I/J/J-v2/K/K-lite/K-lite-replicate, Claude uses 684 tool calls and Codex uses 762. For free-shell B-cc, Claude context growth is +7.7K versus Codex +19.6K. But this does not become an overall accuracy advantage: the same bash-based group is 188/216 for Claude and 203/216 for Codex.

- B-cc/C-lite were highest in the early paired experiment but are 20/24 in the fresh-env rerun. The difference is mainly the fresh-env runner, router prompt, and host-version combination, again showing that n=24 single-run rankings move by 2-3 boundary samples.
- `L-agentic` at 19/24 in the full run is a low point from prompt/implementation iteration. Later `rerun-claude-lagentic-newcli-2026-05-26` and `rerun-claude-l-restore-2026-05-27` both recover to 22/24; the current Claude L conclusion should use 22/24.
- `rerun-claude-bash-rerun-2026-05-26` shows J/K aggregate 0/24, but its summary is missing expected fields, so the aggregator reads all expected values as null. This is a scoring bug and is excluded from conclusions.

L/M diagnostic reruns:

run	variant	corpus	accuracy	trigger	avg ctx_end	tools	bash err	cost
`rerun-claude-l-restore-2026-05-27`	L-agentic	150	22/24	23/24	33.7K	58	0	$4.57
`rerun-claude-lagentic-newcli-2026-05-26`	L-agentic	150	22/24	24/24	33.1K	57	0	$4.67
`rerun-claude-mbm25-newcli-2026-05-26`	M-bm25	150	22/24	24/24	34.6K	65	0	$5.36
`rerun-claude-1k-lm-2026-05-26`	L-agentic	1K synthetic	22/24	24/24	31.8K	57	0	$4.54
`rerun-claude-1k-lm-2026-05-26`	M-bm25	1K synthetic	21/24	24/24	31.8K	54	0	$4.32

These reruns show that the main Claude-side issue for L/M is not CLI viability. Under the same candidate evidence, final selection is more easily pulled toward task-centric distractors or broad spreadsheet/economics skills.

---

7. Codex with 24 Queries / 150 Skills

7.1 Results

Codex has no paired with/without arm, so this section reports only the 16x24 host experiment. The full Codex experiment was rerun on 2026-05-26 with new CLI semantics:

- Model: `gpt-5.5`, `reasoning_effort=high`.
- Fresh Codex home/project for every `(variant, query)`.
- Disabled skill suffix: `.agentic-skill-router-disabled`.
- CLI strategies keep their semantics: A-router calls `skills route`; D calls DCI search/inspect; L/M call corpus search/inspect.

The table covers the old Codex 9x24 run and later scattered reruns. `cost est.` uses the definition in section 5.

variant	accuracy	trigger	turns total/avg	avg ctx_end	growth	tools	bash err	cost est.
G-native	23/24	n/a	24 / 1.0	19.3K	+0.0K	0	0	$1.64
A-router	11/24	24/24	90 / 3.8	23.9K	+9.1K	48	0	$4.08
B-cc	23/24	24/24	112 / 4.7	34.5K	+19.6K	87	3	$6.00
C-lite	24/24	24/24	114 / 4.8	19.7K	+4.8K	138	0	$3.71
D-agentic	22/24	24/24	103 / 4.3	19.7K	+4.8K	73	0	$3.25
D-agentic-metadata	23/24	24/24	95 / 4.0	19.3K	+4.5K	69	0	$3.01
E-digest	19/24	24/24	84 / 3.5	20.0K	+5.1K	48	0	$2.84
H-bounded	21/24	24/24	148 / 6.2	20.6K	+5.8K	153	1	$4.68
I-meta	23/24	24/24	95 / 4.0	21.1K	+6.3K	51	0	$3.15
J-bounded	22/24	24/24	99 / 4.1	19.8K	+5.0K	77	0	$3.45
J-bounded-v2	21/24	24/24	102 / 4.3	18.1K	+3.3K	78	1	$3.48
K-bounded	23/24	23/24	125 / 5.2	18.1K	+3.2K	71	1	$3.56

variant	accuracy	trigger	turns total/avg	avg ctx_end	growth	tools	bash err	cost est.
K-lite	23/24	24/24	103 / 4.3	17.4K	+2.6K	54	0	$3.30
K-lite-replicate	23/24	24/24	101 / 4.2	17.4K	+2.5K	53	0	$2.96
L-agentic	24/24	24/24	92 / 3.8	17.5K	+2.7K	60	0	$2.80
L-agentic on 1K synthetic	23/24	24/24	88 / 3.7	18.0K	+3.2K	60	0	$3.13
M-bm25	19/24	24/24	102 / 4.3	18.0K	+3.1K	69	0	$3.05

Main differences between Codex and Claude Code:

- Codex native G-native is 23/24 on the 150-skill benchmark; Claude Code fresh-env native is 14/24 (early paired was 15/24).
- Codex routers basically always trigger. K-bounded's 23/24 trigger is a detector miss; that cell still produces a legal match.
- C-lite and L-agentic reach 24/24. L-agentic is among the cheapest and grows context by only +2.7K.
- `L-agentic on 1K synthetic` is the same L-agentic strategy on the 150-skill corpus plus 850 synthetic noise skills. It loses only one cell and context growth stays at +3.2K.
- A-router has no bash error under new CLI semantics but is only 11/24, showing the fixed scorer remains unreliable.

7.2 Error Distribution

Except for A-router, Codex misses are concentrated in a few boundary samples:

variant	miss
G-native	`gh-repo-analytics -> skill-146`
B-cc	`gh-repo-analytics -> skill-046`
C-lite	none
D-agentic	`offer-letter-generator -> skill-072`, `shock-analysis-supply -> skill-026`
D-agentic-metadata	`shock-analysis-demand -> skill-026`
E-digest	`gh-repo-analytics -> skill-046`, `protein-expression-analysis -> skill-026`, `reserves-at-risk-calc -> skill-026`, `shock-analysis-supply -> skill-080`, `weighted-gdp-calc -> skill-026`
H-bounded	`enterprise-information-search -> skill-087`, `gh-repo-analytics -> skill-046`, `protein-expression-analysis -> skill-026`
I-meta	`gh-repo-analytics -> skill-046`
J-bounded	`enterprise-information-search -> skill-013`, `gh-repo-analytics -> skill-046`
J-bounded-v2	`gh-repo-analytics -> skill-046`, `shock-analysis-demand -> skill-026`, `shock-analysis-supply -> skill-080`
K-bounded	`gh-repo-analytics -> skill-046`
K-lite	`gh-repo-analytics -> skill-046`
K-lite-replicate	`gh-repo-analytics -> skill-046`
L-agentic	none
M-bm25	`dialogue-parser -> skill-130`, `gh-repo-analytics -> skill-046`, `pptx-reference-formatting -> skill-103`, `reserves-at-risk-calc -> skill-026`, `shock-analysis-supply -> skill-026`

A-router's misses are not expanded one by one in the table: most of its 13 misses select `skill-140`, with `gh-repo-analytics -> skill-077` and `shock-analysis-supply -> skill-043` also appearing. This looks like fixed-scorer long-query bias rather than runner contamination; A-router had 0 bash errors in this run.

7.3 Evidence for Native Difference

Capturing real native requests for `weighted-gdp-calc` shows:

host	endpoint	skill lines	skill section chars	avg desc chars	execution mechanism
Codex	`/v1/responses`	150	21,117	92	available skills are inlined in instructions
Claude Code	`/v1/messages?beta=true`	150	7,948	20	`Skill` tool + budgeted skill listing

This is not a full causal proof because it covers one representative query and the current CLI/model version. But it is consistent with the 24-query results: under the same corpus and query set, hosts expose different amounts of skill metadata to the model, enough to change native selection quality.

7.4 Later Codex Strategy Iteration

After the old A-J set, D metadata-only, K, K-lite, L, and M were rerun. The new-CLI full rerun now includes all of them in section 7.1, so 150-skill rankings should use section 7.1. `L-agentic on 1K synthetic` is merged into the same table as a scalability row, not an independent strategy.

Interpretation:

- On 150 skills, strong Codex strategies are near ceiling, but not all metadata-only/metadata-first variants are perfect: E-digest and M-bm25 are 19/24, while H/J-v2 are 21/24.
- L-agentic is 24/24 on the 150-skill new-CLI rerun, loses only one cell on 1K synthetic, and barely increases context. This suggests the CLI abstraction is better than hand-written shell for indexing and pagination.
- K-lite and K-lite-replicate are both 23/24 and only miss `gh-repo-analytics`, supporting the earlier ambiguous-sample interpretation.
- M-bm25 is 19/24 under new CLI semantics, so the old claim that it was a low-cost perfect 150-skill strategy no longer holds. BM25 shortlist is sensitive to `skill-105` spreadsheet tasks and the `pptx-reference-formatting` boundary sample.

7.5 L Implementation Notes

L-agentic moves retrieval work into the CLI:

- `corpus search`: returns stable `ref`, `shortId`, score, description summary, and pagination metadata.
- `corpus inspect`: returns full metadata for a small number of candidates, avoiding ad hoc shell pipelines and path parsing.
- The model's job changes from "write a grep/awk/sed retriever" to "generate query terms, compare candidate evidence, and make the final selection."

The main difference from pure shell is clearer scalability boundaries: large-corpus indexing, cache, pagination, top-k, JSON schema, and fingerprinting live in the CLI. The model consumes bounded structured candidates. The 1K result supports this direction, but 79K original Hard still needs more validation.

7.6 Claude Code vs Codex Fresh-Env Attribution

Comparing Claude fresh-env results from section 6.5 with Codex new-CLI results from section 7.1, the difference decomposes into six layers:

Attribution layer	Evidence	Conclusion
Data and CLI semantics	Both use the same 24 queries / 150-skill corpus; L/M both use `corpus search/inspect`; L-agentic SKILL.md body is identical except host metadata and paths.	The gap is not mainly due to query, GT mapping, or L/M CLI semantic differences.
Native skill listing	Codex native 23/24, Claude Code fresh-env native 14/24; packet capture shows Codex native skill section 21,117 chars and avg desc 92 chars, while Claude Code has 7,948 chars and avg desc 20 chars.	Native difference mainly comes from host-specific skill metadata exposure and execution mechanism.
Trigger	Claude with-CLAUDE.md router aggregate 349/360 (96.9%), without 339/360 (94.2%); Codex router aggregate 359/360 (99.7%).	Early Claude's first problem was trigger, but fresh-env trigger is near saturation. The remaining 5.6pp aggregate gap is not explained by CLAUDE.md alone.
Model rerank / candidate comparison	Codex L-agentic 24/24, Claude L restore 22/24; Codex C-lite 24/24, Claude C-lite 20/24; but M-bm25 is Claude 22/24 and Codex 19/24.	Codex is better at final discrimination among small metadata candidate sets; Claude benefits more from strong ranking priors such as BM25 shortlist.
Bash harness efficiency	Bash-based 9 variants combined: Claude 188/216, 684 tool calls, avg ctx growth +6.1K; Codex 203/216, 762 tool calls, avg ctx growth +5.9K. B-cc alone: Claude +7.7K growth, Codex +19.6K.	Claude's advantage is fewer tool calls and lower incremental context in some free-bash cases, but Codex still has higher accuracy, lower absolute ctxEnd, and lower cost.
Output contract	Claude misses still include aliases or the router itself, such as `citation-check`, `d3`, `xlsx`, `agentic-skill-router:agentic-skill-router-skills`; Codex new-CLI main run mostly outputs legal `skill-NNN` ids outside older bad runs.	Claude needs stronger JSON schema/regex guard and automatic retry; some errors are not retrieval failures but final-answer contract leakage.
Context and cost baseline	L-agentic: Claude 22/24, ctxStart 27.1K, ctxEnd 33.7K, cost $4.57; Codex 24/24, ctxStart 14.8K, ctxEnd 17.5K, cost $2.80. Native: Claude ctx 34.1K vs Codex 19.3K.	The same strategy costs roughly 1.6x more and uses 1.8-1.9x context on Claude Code. Codex cost/context conclusions cannot be transferred directly to Claude.

Typical cases:

- `gh-repo-analytics`: Claude L/M and many metadata-only variants choose `skill-046`, a task-centric description; Codex L selects the GT `skill-021`. The sample is ambiguous, but clearly shows Claude being pulled toward task-centric distractors.

- `shock-analysis-demand/supply`: GT is generic spreadsheet skill `skill-105`; Claude L/M are often attracted to economics/spreadsheet distractors (`skill-026`, `skill-080`, `skill-109`), while Codex L consistently selects `skill-105`. This indicates that body-on-tie or stronger "tool substrate vs domain method" rules matter more for Claude.
- Native `weighted-gdp-calc`: Claude native selects `skill-026`, Codex native selects `skill-105`, matching section 7.3's prompt-capture evidence.

Current safest attribution:

1. **The native gap is caused by host prompt/listing mechanics.**
2. **The router gap is caused by model rerank plus output contract, not corpus or CLI semantics.**
3. **A production Claude Code default needs stronger guardrails:** candidate generation can continue with K-lite/L-agentic/M-bm25, but final output must validate legal ids; spreadsheet, GitHub analytics, PPTX, and other ambiguous samples need body-on-tie or second-pass rerank.

---

8. Scaling: J-bounded

8.1 Why Scale It

The 150-skill comparison can rank variants, but it does not represent real large installations. J-bounded is the lowest-cost Claude Code router among variants that reach 22/24, so it needs validation on larger skill pools.

8.2 v1 Failure Mechanism

J-bounded v1 has two scale problems:

1. **Shell glob hits `ARG_MAX`:** `~/.claude/skills/*/SKILL.md.skill-router-disabled` exceeds the Linux argv limit after roughly 13K paths.
2. **`head -20` truncates the correct answer:** On the 79K Hard pool, keywords can match tens or hundreds of candidates; if GT ranks after position 20, it is systematically removed.

8.3 v2 Fix

J-bounded-v2 changes only the body workflow. Frontmatter description is unchanged:

Item	v1	v2
Enumeration	shell glob	`find -print0
Candidate limit	`head -20`	no truncation; if too many candidates match, narrow keywords and rerun
Keyword guidance	distinctive keywords	narrow technical terms, avoid broad words
Shortlist payload	description text	path-only first, then query descriptions

8.4 Results

Config	host	accuracy	trigger	cost	avg cost/cell	duration	avg ctx_end
J-v1 x 150	Claude Code	22/24	23/24	$3.07	$0.128	374s	30.7K
J-v2 x 150	Claude Code	22/24	23/24	$3.95	$0.164	665s	30.8K
J-v2 x 79K Hard	Claude Code	12/24	23/24	$5.33	$0.222	1577s	36.6K
J-v2 x paper-core single Hard	Codex	11/24 strict; 14/24 alias-normalized	24/24	$12.44	$0.518	3056s	30.3K

Explanation:

- v2 keeps 150-skill accuracy but is slower and more expensive.
- v2 keeps payload bounded at 79K; cost rises only about 35% and ctx_end about 19%.

- Accuracy drops from 22/24 to 12/24 or 11/24 mainly because description-only evidence is inadequate against many near-synonym or catch-all skills.
- In the Codex paper-core single run, J-v2 outputs original skill-name aliases instead of `sr-*` opaque ids for 3 rows. Strict score is 11/24; mapping aliases back to GT gives 14/24. This exposes an output-format risk caused by visible `name:` metadata.

This corrects the extrapolation boundary of the 150-skill conclusion: J-bounded is a low-cost Pareto point for Claude Code at 150 skills, but not a high-accuracy 80K-scale solution. The production direction should be metadata-first + body-on-tie, not permanent metadata-only.

8.5 Paper Metadata-Only Comparison

The SkillRouter paper's `nd` metric uses name + description only, but the denominator is 75 core queries, not single-only:

source	method	input	query / corpus	Hard Hit@1	Avg Hit@1
Paper Table 9	BM25	nd	75 core, Easy/Hard	0.0%	0.0%
Paper Table 9	Qwen3-Emb-0.6B	nd	75 core, Easy/Hard	14.7%	18.7%
Paper Table 9	Qwen3-Emb-8B	nd	75 core, Easy/Hard	20.0%	25.3%
Paper Table 21	Qwen3-Emb-8B x Qwen3-Rank-8B	nd	75 core, Easy/Hard	18.7%	24.0%
Paper Table 21	Qwen3-Emb-0.6B x GPT-5.4-mini	nd	75 core, Easy/Hard	29.3%	33.3%

The paper's `Single Hit@1` uses a different denominator and belongs to the full-skill-text main result:

source	method	input	Single Hit@1
Paper Table 4 / 22	Qwen3-Emb-0.6B x Qwen3-Rank-0.6B	full	62.5%
Paper Table 4 / 22	Qwen3-Emb-8B x Qwen3-Rank-8B	full	66.7%
Paper Table 4 / 22	SR-Emb-0.6B x SR-Rank-0.6B	full	72.9%

Our supplemental rerun fills a cell not published by the paper: `paper-core single x Hard x metadata-only agentic router`:

method	input	query / corpus	accuracy	trigger	duration	cost est.
M-bm25	metadata-only	24 paper-core single, 79,141 Hard	14/24 (58.3%)	24/24	964s	$6.822
J-bounded-v2	metadata-only	24 paper-core single, 79,141 Hard	11/24 strict; 14/24 alias-normalized	24/24	3056s	$12.436
M-bm25	metadata-only	current 24-query subset, 79,141 Hard	11/24 (45.8%)	24/24	1016s	$6.347

These results show that an agentic metadata-only router can far exceed paper BM25-nd top-1 at 0%, but remains well below the paper's 62.5%-72.9% full-text single routing range. The missing piece is body-on-tie or full-text rerank, not more metadata-only tuning as the sole decision source.

---

9. SkillRouter 75 Core x Easy 78K x Multi-Skill Scaling

9.1 Why Extend

Section 8's paper-core single Hard rerun filled the single-skill x Hard x metadata-only cell, but left two gaps:

- **The query scale is small and covers only 24 single-skill tasks.** The paper's default scored eval uses 75 core queries (24 single + 51 multi). Section 12 #2 explicitly listed "extend to 75 core and implement multi-label Hit@1" as follow-up.
- **Only one variant, one host, and one tier were run** (M-bm25/J-v2 on Codex on Hard). There was no K/J/M x Claude/Codex x Easy comparison.

This section fills those gaps. Artifacts live in `experiments/skillrouter-easy/`.

9.2 Setup

Item	Choice
Corpus	Full SkillRouter `eval_core/easy/*.jsonl.gz`, 78,361 skills
Installation shape	metadata-only: each skill is written as `<HOME>/.<host>/skills/sr-XXXXX/SKILL.md.agentic-skill-router-disabled`, with only `name` + `description` frontmatter and no body
Anonymization	Deterministically shuffled `sr-XXXXX` directory ids (seed 20260525); frontmatter `name:` and `description:` keep upstream strings because paper `nd` input includes name
Query set	75 core queries from `relevance.json` where `core_gt_ids` is non-empty (24 single + 51 multi, paper Section 2 / Appendix A)
Metric	Hit@1: top-1 is correct if any `core_gt_ids[*]` appears, matching the paper's multi-skill Hit@1 definition
Output contract	one-line JSON `{"matched_skill_names":["sr-AAAAA",...,"sr-JJJJJ"]}`, ordered top-10; position 0 counts for Hit@1
Variants	K-bounded / J-bounded-v2 / M-bm25, forked from section 3 variants and changed from "pick one" to "rank top-10", max 4 tool calls
Hosts	Claude Code (claude-opus-4-7) and Codex (gpt-5.5, reasoning=high), 3 variants each, 6 cells
Total scale	6 cells x 75 queries = 450 agent calls

To keep K/J keyword grep sub-second across 78K files, setup also generates `<skillsRoot>/.flat-metadata.tsv` with 78,361 rows (17MB): `<sr-id>\t<name>\t<description>`. Variant SKILL.md files grep this TSV directly instead of doing `find ... | xargs grep` over 78K files, which took 24s per call and would time out over 4 iterations. M-bm25 warms the BM25 index cache at setup. These optimizations do not change the routing signal (name + description); they only reduce metadata-only routing latency at 80K scale from minutes to seconds.

9.3 Main Result

variant	host	Hit@1	single (n=24)	multi (n=51)	answer rate	timeout	wall
J-bounded-v2	codex	**40.0% (30/75)**	9/24 (37.5%)	21/51 (41.2%)	75/75	0	2532s
M-bm25	codex	37.3% (28/75)	7/24 (29.2%)	21/51 (41.2%)	75/75	0	3444s
M-bm25	claude	34.7% (26/75)	7/24 (29.2%)	19/51 (37.3%)	75/75	0	1329s
K-bounded	codex	32.0% (24/75)	5/24 (20.8%)	19/51 (37.3%)	74/75	1	2966s
K-bounded	claude	25.3% (19/75)	4/24 (16.7%)	15/51 (29.4%)	75/75	0	1037s
J-bounded-v2	claude	21.3% (16/75)	5/24 (20.8%)	11/51 (21.6%)	75/75	0	1009s

Average by Variant (Across Hosts)

variant	avg Hit@1	cells
M-bm25	36.0%	2
J-bounded-v2	30.7%	2
K-bounded	28.7%	2

Average by Host (Across Variants)

host	avg Hit@1	avg wall	cells
Codex	36.4%	2981s	3
Claude Code	27.1%	1125s	3

Codex is 9.3pp higher than Claude Code, but wall time is 2.6x longer. Claude Code completes all three cells in 17-22 minutes; Codex needs 42-57 minutes. The slower time may come from deeper Codex reasoning (GPT-5.5 high effort) or tool-execution overhead. Higher accuracy is likely the model's reasoning ability rather than host prompt difference, because section 9 uses the same SKILL.md template for all cells. This differs from the native prompt explanation in section 7.3 and needs further packet capture (see section 12 #7).

Total wall time is about 57 minutes, with 449/450 answered, 1 timeout, and 0 errors.

9.4 Comparison with Paper Easy Baselines

source	method	input	Easy Hit@1
Paper Table 9	BM25	nd	0.0%
Paper Table 9	Qwen3-Emb-0.6B	nd	22.7%
Paper Table 9	Qwen3-Emb-8B	nd	30.7%
Paper Table 9	BM25	**full** (with body)	34.7%
Paper Table 9	Qwen3-Emb-0.6B	full	58.7%
Paper Table 9	Qwen3-Emb-8B	full	65.3%
Paper Table 2	SR-Emb-0.6B	full	66.7%
Paper Table 2/3	SR-Emb-0.6B x SR-Rank-0.6B	full (1.2B pipeline)	76.0% (A-Hit@1, Easy+Hard average; Easy-only not published)
This section	**codex / J-bounded-v2**	**nd (agentic loop)**	**40.0%**
This section	**codex / M-bm25**	nd (agentic loop)	37.3%
This section	**claude / M-bm25**	nd (agentic loop)	34.7%

The directly comparable rows are the three `nd` baselines at the top. **All 6 cells beat BM25 nd by at least +21pp.** 4/6 cells beat Qwen3-Emb-0.6B nd. 3/6 cells beat the strongest nd baseline, Qwen3-Emb-8B (30.7%). The best codex/J-v2 cell exceeds it by +9.3pp. The best cell (40.0%) also exceeds the paper's BM25 with full body at 34.7%, despite being structurally body-free.

Claude/M-bm25 at 34.7% exactly matches paper BM25-full. This means a metadata-only agentic loop reaches BM25 with full body. The improvement likely comes from LLM query rewriting and multiple bounded searches compensating for sparse descriptions. In the paper's Easy pool, 18.7% of descriptions are shorter than 10 words, yet the best cell still recovers GTs consistently.

However, the best cell (40.0%) remains 25-36pp below strong full-body baselines (Qwen3-Emb-8B full 65.3%, SR-pipeline 74-76%). That gap is structurally unavailable to metadata-only routing, consistent with section 8.5.

9.5 Relationship to Section 8 Paper-Core Single Hard

Section 8.5 gave J-v2 strict 11/24 (alias-normalized 14/24) and M-bm25 14/24 on paper-core single Hard. This section extends the same variant family to Easy + 75 core and adds three facts:

1. **The multi-skill subset scores slightly higher than the single-skill subset.** M-bm25/codex is 7/24 (29.2%) on single and 21/51 (41.2%) on multi. This is partly due to metric definition: multi-skill queries have multiple GTs, and any one in top-1 counts as a hit. Paper Table 4 also shows a similar gap where R@10 is much higher than single Hit@1.
2. **Host differences are more stable at 75 queries than at 24.** Section 8 could run only on Codex due to cost/time. This section gives the first same-corpus same-variant side-by-side comparison across hosts, with Codex averaging +9.3pp. But section 7.3's native prompt capture explanation (skill listing density) should not be reused directly because section 9 uses router SKILL.md, not native listing. The cause is more likely model difference (GPT-5.5 vs Opus 4.7 metadata rerank ability) and needs packet-capture confirmation.
3. **K-bounded's 78K shell-glob issue is solved by corpus shape, not prompt magic.** Section 8.2 shows v1 fails around 13K paths. Here setup writes `.flat-metadata.tsv`, and K/J grep that single file instead of enumerating 78K files. K-bounded completes at 78K and scores 24/75 = 32.0% on Codex, 19/75 = 25.3% on Claude. This means K-bounded remains usable when the corpus layout cooperates.

9.6 Engineering Optimization Notes

Five issues were fixed while running 6 cells x 75 queries. They are unrelated to prompt/agent behavior and directly related to making metadata-only routing work at 80K scale. Details are in `experiments/skillrouter-easy/IMPLEMENTATION_PLAN.md`.

1. **Missing auth:** tmp HOME lacked `.credentials.json` / `auth.json`, causing "Not logged in" on first startup. Fix: copy corresponding auth files from real HOME during setup.
2. **K-bounded shell glob hits ARG_MAX:** solved by the flat TSV index in section 9.2.
3. **78K `find | xargs grep` takes 24s per call, and 4 iterations time out:** same fix; grepping the 17MB TSV takes under 100ms, about 250x faster.
4. **M-bm25 first query cold-starts at 17s:** setup warms the BM25 index cache and sets `AGENTIC_SKILL_ROUTER_CORPUS_CACHE_TTL_MS` to 24h.
5. **Codex shell session closes stdin during long agent loops and stalls in retry:** after issue #3 reduced tool calls to 4 or fewer, this mostly disappears. Across 75 queries there is one remaining timeout (`codex/K-bounded` on `xlsx-recover-data`, killed after the 600s limit).

9.7 Error Distribution

The 75 queries fall into three layers by hits across the 6 cells:

Hit type	query count	share	Notes
Hit in all 6 cells	12	16.0%	stable easy samples
Hit in some cells (mixed)	24	32.0%	variant/host split region
Miss in all 6 cells	39	52.0%	metadata-only ceiling

The **39 all-miss queries** are the structural ceiling for metadata-only routing: no variant/host combination can hit GT at top-1. By tier, 14/39 are single-skill (58.3% of single queries miss) and 25/39 are multi-skill (49.0% of multi queries miss). Seven queries (18%) show **all 6 cells agreeing on the same wrong skill**, such as `earthquake-plate-calculation` always selecting `sr-26212` and `quantum-numerical-simulation` always selecting `sr-68239`. These consensus misses indicate that metadata descriptions can make a distractor look more relevant than the GT, either because of annotation dispute or weak GT description.

Among the **24 mixed queries**, 15 are hit at least once by both hosts, 6 only by Codex, and 3 only by Claude Code. Codex-only hits are twice Claude-only hits, further supporting Codex's systematic metadata-rerank advantage. Within mixed queries, 5 are hit in 5/6 cells and 6 are hit in only 1/6 cells.

Therefore, to move Easy 78K metadata-only Hit@1 from 40% to 50%+, the router must solve at least 8 of the 39 all-miss queries. Tuning variant parameters can only recover 1-3 additional hits from the 24 mixed queries. Body-on-tie or full-text rerank is necessary to break the 52% ceiling.

Every `(query, cell)` tool chain is available in `experiments/skillrouter-easy/runs/traces-report.html` and `traces-compact.json`.

---

10. Key Failure Cases

10.1 `gh-repo-analytics`

Eight Claude router variants miss this query in both arms. The GT `skill-021` description is tool-centric: use gh CLI to operate repos/issues/PRs. Several variants choose `skill-046`, whose description is task-centric: track and visualize GitHub contributions, PRs, and issues resolved over time. The query asks for a December community pulse with PRs, issues, and top contributors. By description semantics, `skill-046` is closer to the query. This sample should be marked ambiguous and should not be used alone to reject a router design.

The main table uses strict GT and includes the ambiguous sample by default. Sensitivity: if this single row is removed from the Claude Code router table, every router denominator becomes 23; B-cc/C-lite become 23/23, D/E/H/J become 22/23, I-meta becomes 21/23, and A-router becomes 13/23. Therefore one-cell ranking differences should be read as strict scores with a disputed sample included.

10.2 `shock-analysis-supply`

GT is the generic Excel skill `skill-105`. With CLAUDE.md forcing routing, some metadata-only variants are attracted to more specialized economics/timeseries skills. In the without arm, some zero-tool direct outputs happen to return `skill-105` and score correctly. Body-reading B-cc/C-lite recover the correct selection. This is a canonical body-on-tie case.

10.3 A-router

A-router passes the long query directly to a fixed scorer. Real queries contain many procedural details, paths, and format constraints that dilute high-signal keywords. In contrast, J/B/C/H let the LLM perform query rewrite or keyword extraction. A-router should be changed into a candidate generator plus LLM reranker, not a CLI that commits directly.

---

11. Limitations

1. **Each cell runs only once.** With 24 queries, one cell is 4.2pp; 23/24 versus 22/24 is not statistically significant.
2. **CLAUDE.md is not a system prompt.** It is a user-message context block. It is strong, but not equivalent to forced `tool_choice`.

3. **The 150-skill comparison is small.** Scaling experiments show metadata-only accuracy drops substantially with corpus size.
4. **Only routing is measured.** Whether the downstream skill body completes the task is outside this report.
5. **Codex cost is estimated.** `cost est.` is not a same-billing-basis invoice value and should not be used for exact finance decisions.
6. **The data has ambiguous / overloaded samples.** `gh-repo-analytics` and Excel-related queries affect overall rankings.
7. **Host versions drift.** Claude Code and Codex native skill display strategies may change across releases.
8. **Language coverage is limited.** All 24 original queries are English; Chinese and mixed-language requests are not tested.
9. **Paper comparison denominators are not identical.** The paper does not publish `Single x nd x Hard-only`; this report's paper-core single Hard rerun (section 8.5) and 75 core x Easy x multi-skill rerun (section 9) are supplemental experiments, not reproductions of the paper's BM25/embedding pipeline. The best cell is directly comparable to paper `nd` baselines; full-body strong baselines should be read as upper references.
10. **Section 9 uses the paper's any-GT Hit@1 definition for multi-skill queries, not strict set match.** Multi-skill queries have 2-7 GTs; any GT in top-1 counts as a hit, so the score is systematically easier than single-skill. R@10 / FC@10 / nDCG@10 can be recomputed offline from saved top-10 outputs in `runs/<host>-<variant>/<query>.jsonl`, but this report does not include those scores.
11. **Section 9 runs Easy only, not Hard.** Hard adds 780 LLM-distilled distractors to Easy 78,361. Paper Table 9 shows method-specific Easy-to-Hard drops (BM25 no drop, Emb-8B down 10.7pp), so agentic-router Hard performance requires an actual run. The 40.0% best result is an Easy number.

---

12. Recommendations

Short-term engineering recommendations:

- **Claude Code default:** In the fresh-env rerun, K-lite is the lowest-cost member of the 22/24 group. L-agentic/M-bm25 also stabilize at 22/24 in restore/newcli diagnostics. A production path should first add legal-id guard + retry, then choose among K-lite/L-agentic/M-bm25 by dependency and cost; low confidence or failure should escalate to body-on-tie.
- **Codex default:** L-agentic is the current first choice in the new CLI rerun: 24/24 on 150 skills, low context growth, and only one-cell loss on 1K synthetic. C-lite also reaches 24/24 but uses more tools. M-bm25 drops to 19/24 in the new run and should not be default.
- **All hosts:** Validate `matched_skill_name` by regex and existence. If it is not a legal installed disabled skill, automatically retry or use a rescue fallback.
- **A-router:** Do not use as a default. Rebuild it as candidate generator + LLM reranker.
- **Large scale:** Keep metadata-first as a fast path, but 79K Hard shows that body-on-tie / full-text rerank is mandatory; otherwise similar natural-pool skills will consistently absorb traffic.
- **Cross-host interpretation:** Do not assume Codex 24/24 L-agentic transfers to Claude Code. Claude's native listing, context baseline, and final id contract differ and need separate gating.

Follow-up experiments:

1. Run N=3 or N=5 repetitions for boundary queries and report confidence intervals.
2. ~~Extend to SkillRouter 75 core queries and implement multi-label Hit@1.~~ Completed in section 9 (K/J-v2/M x Claude/Codex x Easy 78K x 75 core, best codex/J-v2 40.0%). Next step: run the same matrix on Hard 79,141 to compare directly with paper Avg = (Easy+Hard)/2.
3. Implement metadata-first + body-on-tie and compare against J-v2, L-agentic, and M-bm25. Expected to recover part of the 25-36pp gap to full-body strong baselines.
4. Add end-to-end execution validation for at least Excel, PDF, PPTX, and GitHub analytics.
5. Mark `gh-repo-analytics` as ambiguous or relabel GT to avoid interpreting a data dispute as retriever failure.
6. Recompute R@10 / FC@10 / nDCG@10 / MRR@10 offline from section 9 saved top-10 outputs and compare with paper Table 4 multi-skill metrics. Raw data is already available.
7. Investigate why Codex is about 9pp above Claude Code in section 9. Section 7.3 explains native skill listing density, but section 9 uses the same SKILL.md template, so the more likely source is model difference (GPT-5.5 vs Opus 4.7), not host listing. Confirm with packet capture like section 7.3.

---

13. Conclusion

Starting from the 150-skill comparison and scaling through 1K synthetic, 79K Hard, and 78K Easy 75-core, this report evaluates the boundaries of metadata-only agentic routers. Four core takeaways:

1. **Agentic metadata-only routers can consistently beat traditional BM25/embedding `nd` baselines on the 78K Easy pool.** The best cell, codex/J-v2, reaches 40.0% Hit@1, +9.3pp over the strongest paper `nd` baseline Qwen3-Emb-8B (30.7%) and above BM25 with full body (34.7%). This demonstrates that LLM-driven query rewrite plus multi-step bounded retrieval adds real information over sparse metadata.
2. **52% of queries form a structural ceiling for metadata-only routing.** Among the 39/75 queries missed by all 6 cells, 7 show cross-variant cross-host consensus on the same wrong skill. Description signal alone is not enough to distinguish GT from high-quality distractors. Breaking past 40% requires body-on-tie or full-text rerank, consistent with the section 8 Hard-pool result.
3. **Host/model choice matters more than variant choice.** Under the same SKILL.md template, Codex (GPT-5.5) is 9.3pp above Claude Code (Opus 4.7), 36.4% vs 27.1%. The largest variant gap is 7.3pp (M-bm25 36.0% vs K-bounded 28.7%). Production should choose the model first and tune prompts second.
4. **The 150-skill difference between Claude Code and Codex is not a single-strategy issue.** Fresh-env rerun shows Codex router aggregate 321/360 and Claude Code with-CLAUDE.md 301/360; native gap is even larger (23/24 vs 14/24). Native gap mainly comes from host skill-listing information density. Router gap mainly comes from model rerank, output-id contract, and context/cost baseline differences. Claude Code needs legal-id guard, retry, and body-on-tie to approach Codex stability.

Appendix A. File Index

Content	Path
All-experiments visualization (HTML)	`experiments/dci-compare/runs/report-all-experiments.html`
All-experiments renderer	`experiments/dci-compare/render-all-experiments.mjs`
Claude paired driver	`experiments/dci-compare/routing-only-paired.mjs`
Claude paired raw summary	`experiments/dci-compare/runs/routing-only-9x24-claudemd/summary.json`
Claude paired HTML report	`experiments/dci-compare/runs/routing-only-9x24-claudemd/report.html`
Claude/Codex fresh-env 16-variant raw run	`experiments/dci-compare/runs/rerun-150-full-2026-05-26/`
Claude L-agentic restore rerun	`experiments/dci-compare/runs/rerun-claude-l-restore-2026-05-27/aggregates.json`
Claude L/M 1K synthetic rerun	`experiments/dci-compare/runs/rerun-claude-1k-lm-2026-05-26/aggregates.json`
Claude newvariants driver	`experiments/dci-compare/routing-only-newvariants.mjs`
Claude newvariants render	`experiments/dci-compare/render-newvariants.mjs`
Claude D-meta/K/K-lite/L/M 150	`experiments/dci-compare/runs/claude-routing-only-newvariants-150-20260525/summary.json`
Claude K-bounded 1K	`experiments/dci-compare/runs/claude-routing-only-newvariants-k-1k-20260525/summary.json`
Claude L-agentic 1K	`experiments/dci-compare/runs/claude-routing-only-newvariants-1k-20260525/summary.json`
Claude L-agentic 150 rerun	`experiments/dci-compare/runs/claude-routing-only-newvariants-l-150-rerun-20260525/summary.json`
Claude L-agentic v2 150	`experiments/dci-compare/runs/claude-routing-only-newvariants-l-v2-150-20260525/summary.json`
Claude L-agentic v3 150	`experiments/dci-compare/runs/claude-routing-only-newvariants-l-v3-150-20260525/summary.json`
Claude L-agentic v3.1 150	`experiments/dci-compare/runs/claude-routing-only-newvariants-l-v3.1-150-20260525/summary.json`
L-agentic v1 / v2 / v3 SKILL.md backups	`experiments/dci-compare/variants/routing-only/L-agentic-v{1,2,3}.SKILL.md.backup`
Codex 9x24 summary	`experiments/dci-compare/runs/codex-routing-only-9x24/summary.json`
Codex new-CLI 16x24 rerun	`experiments/dci-compare/runs/rerun-codex-newcli-full-2026-05-26/aggregates.json`
Codex D-agentic metadata-only	`experiments/dci-compare/runs/codex-routing-only-d-agentic-metadata-24-20260524/summary.json`
Codex K-bounded	`experiments/dci-compare/runs/codex-routing-only-k-24-20260524/summary.json`
Codex K-lite fixed	`experiments/dci-compare/runs/codex-routing-only-k-lite-fix-24-20260524/summary.json`
Codex L-agentic 150	`experiments/dci-compare/runs/codex-routing-only-l-agentic-24-20260525-v2/summary.json`
Codex L-agentic 1K new-CLI rerun	`experiments/dci-compare/runs/rerun-codex-newcli-l-agentic-1k-2026-05-26/aggregates.json`
Codex M-bm25 150	`experiments/dci-compare/runs/codex-routing-only-m-bm25-index-fix-24-20260525/summary.json`

Content	Path
Codex M-bm25 original Hard current 24	`experiments/dci-compare/runs/codex-routing-only-m-bm25-skillrouter-hard-24-20260525/summary.json`
Codex M-bm25 paper-core single Hard	`experiments/dci-compare/runs/codex-routing-only-paper-single-hard-m-bm25-24-20260525/summary.json`
Codex J-v2 paper-core single Hard	`experiments/dci-compare/runs/codex-routing-only-paper-single-hard-j-v2-24-20260525/summary.json`
Native prompt capture	`experiments/dci-compare/runs/native-prompt-capture/analysis.md`
Query set	`experiments/dci-compare/queries.json`
Corpus manifest	`experiments/dci-compare/corpus-manifest.json`
Router variants	`experiments/dci-compare/variants/routing-only/*.SKILL.md`
Codex router variants	`experiments/dci-compare/variants/routing-only-codex/*.SKILL.md`
Scaling notes	`experiments/scaling-jbounded/EXPERIMENT_NOTES.md`
J-bounded-v2	`experiments/scaling-jbounded/variants/J-bounded-v2.SKILL.md`
150-scale v2 report	`experiments/scaling-jbounded/runs/sweep24-v2-150-cmd/report.md`
79K-scale v2 report	`experiments/scaling-jbounded/runs/sweep24-v2-full-cmd/report.md`
Section 9 README + protocol	`experiments/skillrouter-easy/README.md`
Section 9 implementation plan	`experiments/skillrouter-easy/IMPLEMENTATION_PLAN.md`
Section 9 install script	`experiments/skillrouter-easy/install-easy-pool.mjs`
Section 9 variants (Claude/Codex x K/J-v2/M)	`experiments/skillrouter-easy/variants/{claude,codex}/{K-bounded,J-bounded-v2,M-bm25}.SKILL.md`
Section 9 runner + scorer	`experiments/skillrouter-easy/run.mjs`
Section 9 report renderer	`experiments/skillrouter-easy/render-report.mjs`
Section 9 aggregated report	`experiments/skillrouter-easy/runs/report.md`
Section 9 cross-cell summary	`experiments/skillrouter-easy/runs/all-summary.json`
Section 9 per-cell summary (metrics + per-query top-10)	`experiments/skillrouter-easy/runs/<host>-<variant>/summary.json`
Section 9 per-query execution traces (HTML)	`experiments/skillrouter-easy/runs/traces-report.html`
Section 9 per-query execution traces (JSON)	`experiments/skillrouter-easy/runs/traces-compact.json`
Section 9 full run log	`experiments/skillrouter-easy/runs/full-run.log`
Section 9 install manifests	`experiments/skillrouter-easy/runs/install-<host>-<variant>/{manifest.json,queries.json,install.log}`