MINING AGENT SKILLS FROM PRODUCTION TRACES

Anonymous authors
Paper under double-blind review

ABSTRACT

Agent skills that record procedural instructions are increasingly mined from execution traces rather than curated by hand. Skill-mining pipelines often use known task outcomes or feedback to guide skill construction. In production, reliable information on whether a run has succeeded may be unavailable. We study how the choice of execution traces, access to success or failure information, and the form of the mined skills affect downstream task performance. Holding the mining pipeline fixed, we compare six combinations of mining evidence and skill forms. Mining evidence has three levels: successful trajectories only, successes and failures with their outcome labels, or the same mix with labels withheld. Skill form has two types: an ordered workflow plan, or a declarative ontology of entities, states, and policies. We evaluate the mined skills on two enterprise benchmarks, ThinkingBox-Bench and APEX-Agents. Analysis of task-level paired differences shows that the benefits of different configurations of mining evidence and skill forms depend on the enterprise domain. On ThinkingBox-Bench, paired differences show that workflows score better than ontology by $+1.7$ pp, Goldilocks beats success-only evidence type by 2.4 pp and Goldilocks blind simulating skills learnt without outcomes is worse by 3.1 pp. APEX-Agents shows a moderate preference for ontologies and no clear preference between evidence regimes. Within each domain, task structure related constraints drive uneven performance with mined skills. These findings motivate tailoring meta-skills to the demands of the target tasks rather than adopting a one-size-fits-all approach.

1 INTRODUCTION

Language model agents start from scratch, not leveraging prior runs, even for recurrent requests. For example, an agent that resolves a customer's return request executes a real procedure: it routes the request by type, verifies eligibility, creates tickets, drives the agentic loop, and stops at a terminal state upon resolution. The ordering of tool calls, the state transitions, and the stopping rule that made the run work are all discarded when the session ends. The next request of the same type starts over.

Skills, reusable procedural instructions loaded into context at inference time (Anthropic, 2025), turn that discarded knowledge into an artifact for future runs. They are non-parametric, auditable, versioned, and loaded on demand rather than accumulated in context, which suits enterprise settings where a change in agent behavior must be inspectable and reversible. A growing literature mines them from execution traces to enable scaling and verifiability (Ni et al., 2026; Wang et al., 2025b).

A skill is only as good as the quality of procedure it describes. A curator synthesizes procedures from experience; a miner synthesizes them from traces. Regardless of the method, a common pitfall is observed. The skills that cause regressions are usually the right ones for the task, prescribing more procedure than the request needed (Li et al., 2026a; Dong et al., 2026). The problem is not which skill fires, but how much it changes the agent's trajectory once loaded in the context. Calibrating the skill's content to balance the downstream impact on task success rate requires identifying which tasks regressed and how their trajectories were influenced. However, the trajectories carrying the most useful knowledge are generated by real users after deployment, and those cannot be read at scale for privacy and compliance reasons; agents likewise rarely know which of their own runs succeeded. The trace-mining literature assumes those labels (Ni et al., 2026; Wang et al., 2026a): pipelines sort traces by success before they distill anything.

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.

In this paper, we study how three things affect downstream task performance: which execution traces the miner is shown, what it is told about their outcomes, and what form the mined skill takes. We hold the mining pipeline fixed and compare six combinations of mining evidence and skill form. Mining evidence has three levels: **Success-only**, which shows the miner successful trajectories alone; **Goldilocks**, which shows successes and failures together with their outcome labels; and **Goldilocks-blind**, which shows that same mixture as goldilocks with the labels withheld. The goldilocks-blind arm is the condition most deployed agents operate under, and because it holds the trace mixture constant against the contrastive arm, the gap between the two isolates the contribution of the label itself rather than of the additional failure traces. Skill form has two types: a **workflow plan**, an ordered sequence of steps, and an **ontology**, a declarative account of the domain's entities, states, and policies that describes the world the agent acts in without dictating an order through it (see Figure 2).

An additional constraint in production is the inability to tune the miner per domain or task family. Hence, we follow a minimal footprint approach to skill mining; the distillation instruction, or meta-skill, is authored once against a single agent's trajectories and then frozen. The meta skill and task traces for unseen agents are used to mine skills that will be used by the agent on the test set. We evaluate these mined skills on two enterprise benchmarks, ThinkingBox-Bench (Li et al., 2026b) and APEX-Agents (Vidgen et al., 2026). Every task is executed with and without the mined skill, so each skill is scored by paired differences across tasks rather than by a shift in the mean.

Our experiments show that skill representation and evidence selection have context-dependent effects on task success. Workflow representations yield positive point estimates relative to ontology representations in Thinkingbox-Bench ($+1.7$pp up in Table3) for 4 out of 5 domains. Variance analysis showed that tasks with multi-entity interactions resolved better with ontology . Goldilocks outperforms success-only selection in Thinkingbox-Bench by 2.4 pp indicating that constrative labels help resolving branches within tasks. Goldilocks-blind selection yields lower point estimates than Goldilocks across all reported settings, with a matched regression of 3.1 pp in Thinkingbox-Bench and 0.3 pp in Apex Agents.

Exploratory task-level analysis suggests that these aggregate results may mask differences in task requirements: ordered execution may favor workflow representations, while entity and policy-rich tasks may favor ontology representations. Similarly, contrasting successful and failed examples may help expose decision branches or completion conditions, while tasks requiring minimal canonical mutation may benefit less. These patterns motivate task-sensitive skill design rather than a single universally preferred configuration. A practical take away for industry practitioners is to be mindful about the varying task constraints when designing an agent.

This paper makes three contributions.

1. **Controlled comparisons of skill form and generation evidence.** We compare workflow- and ontology-based skill generation while varying the evidence available during generation, measuring their effects on downstream enterprise task success.
2. **Quantifying the effect of outcome-label access.** Using matched trajectories for Goldilocks and outcome-blind generation, we isolate the effect of outcome-label visibility under a fixed experimental setup, examining a production-relevant information constraint. ((Zhang et al., 2026b; Xiong et al., 2026)).
3. **Characterizing task-dependent gains and regressions**: We examine how outcomes vary across task families. The exploratory patterns suggest that effective skill design depends on task requirements, with no single configuration consistently performing best.

2 Related Work

Prior work studies skill representation and generation evidence through controlled comparisons (Huang et al., 2026; Jiang et al., 2026), and evaluates downstream skill utility in enterprise settings (Kevin et al., 2026). Our distinguishing focus is a controlled comparison of **workflow versus ontology skill forms** and **generation evidence** for deployed enterprise agents, including a matched-trajectory comparison that isolates the effect of outcome-label visibility. We further examine task-family-specific gains and regressions to characterize where these choices help or hurt,

complementing prior regression analyses (Tank & Nama, 2026) and motivating task-sensitive rather than universally preferred skill configurations.

Skill acquisition and evolution. Skill acquisition methods differ in the evidence used to generate skills and whether skills are subsequently refined. Voyager learns executable skills, while ExpeL extracts natural-language insights (Wang et al., 2024; Zhao et al., 2024); Agent Workflow Memory and Trace2Skill mine reusable routines and lessons from trajectories (Wang et al., 2025b; Ni et al., 2026). SkillGen contrasts successes and failures, while other approaches use self-judged experiences or adapt without labeled supervision (Ma et al., 2026; Ouyang et al., 2026; Zhang et al., 2026b). Controlled studies examine evidence mixtures, generation methods, representations, and outcome labels (Huang et al., 2026; Zhong et al., 2026; Jiang et al., 2026). Beyond initial acquisition, ReMe and SkillEvolver update external knowledge from execution feedback (Cao et al., 2026; Zhang et al., 2026a); SkillRL couples library evolution with policy training (Xia et al., 2026); and SkillOpt accepts skill edits when held-out validation improves (Yang et al., 2026). In contrast to these iterative approaches, our study examines one-shot skill generation without subsequent skill evolution. Within this setting, we isolate the downstream effects of skill form and generation-time evidence for deployed enterprise agents, including a matched-trajectory comparison of Goldilocks and outcome-blind conditions that isolates outcome-label visibility.

Skill form. Prior work explores how reusable knowledge is expressed, from procedural guidance to structured representations. Memp compares stepwise and script-like guidance, while Agent Skill Induction compares programs and text under programmatic verification (Fang et al., 2026; Wang et al., 2025a). Feng et al. (2026) compare task- and subtask-level induction in text and code. Learning domain ontologies (Babaei Giglou et al., 2023; Lo et al., 2024) is distinct from organizing skills in ontologies (Liang et al., 2026). Our focus is neither ontology induction nor skill-library organization: we compare workflow- and task ontology-based skill forms under controlled conditions, evaluating their downstream effects on enterprise task success.

Downstream utility and task-dependent effects. Skill availability, invocation, and artifact quality do not by themselves establish downstream benefit. Skill or Skip studies selective invocation, while More Skills, Worse Agents? examines selection errors and context effects in growing libraries (Chen et al., 2026; Song & Wei, 2026). SkillFlow finds that frequent use need not help without suitable content (Li et al., 2025). Downstream evaluations compare curated skills with no skills and distilled skills with raw traces under frozen deployment (Li et al., 2026a; Lei et al., 2026). ACES measures the added value of existing skills through paired enterprise trials under fixed execution and grading conditions (Kevin et al., 2026). These evaluations also motivate looking beyond average gains: SWE-Skills-Bench reports limited average improvements and compatibility-related regressions, while other studies document uneven benefits and skill-induced failures (Han et al., 2026; Wang et al., 2026b; Dong et al., 2026). The Regression Tax explicitly separates gains, regressions, and residual failures (Tank & Nama, 2026). We complement these evaluations by examining how downstream gains and regressions vary across task families under different skill-form and evidence conditions. Our exploratory analysis of the study results relates these differences to task requirements, motivating task-sensitive skill design rather than a universally preferred configuration.

3 METHOD

3.1 SETTING AND DESIGN OVERVIEW

We study how the evidence a miner sees and the form of the skill it writes affect an agent's success on tasks the skill was not mined from. Each task x has an initial environment state and a hidden binary grader, and every task belongs to one of G task groups defined before mining. A skill is a natural-language Markdown document with a routing description and operational guidance. A meta-skill m instructs a miner F to extract such a skill s from trajectory evidence E, so that $s = F(m, E)$. Holding the miner and the meta-skills fixed, we cross two skill forms with three levels of mining evidence, giving six arms. For every arm, we select evidence from no-skill runs of training tasks, mine skills once per fold, and test them on held-out tasks (Figure 1; Section 4.4).

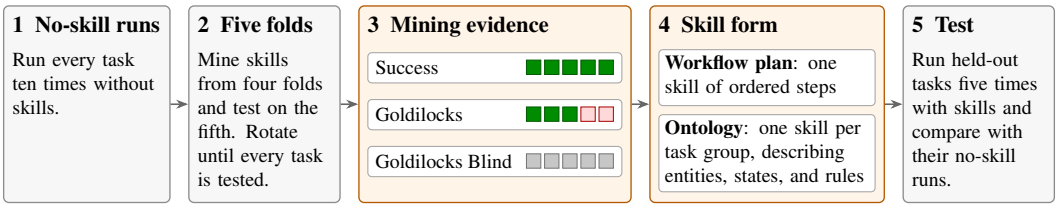


Figure 1: Study design. We vary two things, shaded: the evidence the miner sees and the form of the skill it writes, using the same mining prompts in every domain. Squares show the five runs taken from each selected training task: green passed, red failed, gray outcome hidden. Success runs come from tasks that pass 8–10 of their ten no-skill runs and Goldilocks runs from tasks that pass 3–7; Goldilocks Blind uses the same runs as Goldilocks with their outcomes hidden. The three evidence types and two skill forms give the six arms run in every domain.

3.2 Skill Forms and Mining Scope

Workflow plans. A workflow plan specifies task routing, identifier lookup, preconditions, policy checks, action order, completion checks, and recovery or clarification. Each workflow arm mines one global skill per fold from evidence that spans all training task groups.

Operational ontologies. An operational ontology specifies entities, identifiers, relationships, states, valid transitions, completion signals, constraints, and the lookups needed before modifying records, and it guides the choice of actions without prescribing a sequence. Each ontology arm mines one skill from the training evidence of each group, producing G skills per fold, so comparing the two skill forms also changes the mining scope and the number of skills the agent can read.

Each form has one meta-skill, reused unchanged across domains, that treats traces as evidence rather than as instructions and includes no few-shot examples; test outcomes never guide the revision or selection of a skill (Appendices A.1 and A.2).

3.3 Mining Evidence

Repeated no-skill runs supply the mining evidence and also define task categories by baseline pass frequency. Each task has B runs with binary outcomes $y^{\text{baseline}}_{x,r}$, and $p_x = \sum_{r=1}^{B} y^{\text{baseline}}_{x,r}$ is its pass count. For integer cutoffs $0 < \theta_{\text{G}} < \theta_{\text{S}} \leq B$, the category of task x is

$$c(x) = \begin{cases} \text{Success}, & \theta_{\text{S}} \leq p_x \leq B, \\ \text{Goldilocks}, & \theta_{\text{G}} \leq p_x < \theta_{\text{S}}, \\ \text{Failure}, & 0 \leq p_x < \theta_{\text{G}}. \end{cases} \tag{1}$$

Three levels of mining evidence control which tasks and runs the miner sees and whether it sees their outcomes:

- **Success** (success-only): five labeled passing runs from each selected Success task.
- **Goldilocks**: three passing and two failing runs, with labels, from each selected Goldilocks task.
- **Goldilocks Blind** (outcome-blind): exactly the same tasks, runs, and order as Goldilocks, without labels.

These levels separate outcome-label access from evidence selection. Goldilocks and Goldilocks Blind mine the same runs and differ only in the outcome labels and in the mining instructions that go with them (Appendix A.2), so their comparison measures the value of outcome labels, together with the instructions that use them, on fixed evidence. Success and Goldilocks differ in both the task category and the mix of passing and failing runs, so their comparison covers two complete selection strategies. Goldilocks Blind withholds labels, as a deployed agent must, but its tasks and runs are still selected using outcomes; Appendix A.7 evaluates a fourth level that removes this selection. The miner never sees hidden grading material (Appendix A.3).

4 EXPERIMENTAL SETUP

4.1 BENCHMARKS

ThinkingBox contains 516 enterprise-support tasks in five domains: neobank IT support (104 tasks), consulting helpdesk (107), car insurance (101), retail e-commerce (100, of which we evaluate 98), and hotel booking management (104) (Li et al., 2026b). Agents resolve customer or employee requests with ticketing and domain tools, and a simulated user supplies private information. A run passes only if its final backend state matches a golden state, so a correct decision recorded with nonstandard field values fails. Each domain's two scenario groups, or three in consulting, define its ontology scopes (Table 6).

APEX-Agents contains 480 professional-services tasks, 160 each in investment banking, law, and management consulting (Vidgen et al., 2026). From a request and seeded workspace files, the agent produces a deliverable, such as a financial analysis or legal memo, in an isolated workspace with the benchmark's nine Model Context Protocol servers. A run passes only if the strict binary grader accepts the deliverable against private rubric criteria, and the three domains are the ontology scopes.

4.2 AGENT AND MINER

Both benchmarks run the agent in the GitHub Copilot runtime (Copilot CLI 1.0.79-8 for ThinkingBox) with GPT-5.4 at medium reasoning effort, and the same model mines every skill in one miner call; all runs use seed 42. ThinkingBox's simulated users and rubric judge use GPT-5. ThinkingBox runs time out after 3,600 seconds, with at most 40 tool calls and 900 seconds per turn, and APEX-Agents runs have two million tokens and 1,800 seconds (Appendix A.4).

4.3 MINING CONFIGURATION

Each task has $B = 10$ earlier no-skill runs, and the cutoffs $\theta_{\mathrm{S}} = 8$ and $\theta_{\mathrm{G}} = 3$ make tasks with 8–10, 3–7, and 0–2 passes Success, Goldilocks, and Failure tasks (Equation 1). All arms share five folds, stratified by task scenario and baseline category unless a frozen assignment is reused; each fold's skills are mined from the other four folds, so every task is tested once in each arm. Each skill draws its evidence from up to ten eligible training tasks of its scope and five runs of each, at most 50 trajectories, selected by seeded hash rankings (Appendix A.3). The six arms thus mine 45 skills in each ThinkingBox domain with two scenario groups and 60 in consulting and in APEX-Agents.

4.4 EVALUATION PROTOCOL AND METRICS

Each test run receives only its arm's fold-specific skills, the workflow skill or all G ontologies, installed as native Copilot skills: the agent sees each skill's name and routing description and chooses which to read, a read is recorded only when the runtime opens a skill, and skill content is never inserted into the task prompt. Every held-out task runs $K_{\mathrm{T}} = 5$ times per arm, giving 15,420 skill-enabled ThinkingBox runs and 14,400 APEX-Agents runs across the six arms. Its controls are its ten earlier no-skill runs ($K_{\mathrm{B}} = 10$) in neobank, consulting, car insurance, and APEX-Agents, which also supply the mining evidence, and five no-skill runs collected for this study ($K_{\mathrm{B}} = 5$) in retail and hotel booking. Controls share every setting of the skill runs, although earlier controls cannot rule out drift in the hosted model (Appendix A.4).

The metric is the binary pass rate. Let $y^{\mathrm{T}}_{x,r}$ and $y^{\mathrm{B}}_{x,r}$ be the outcomes of task x in repetition r of the treatment runs, with the skill, and of the baseline runs, without it. For a set $\mathcal{D}$ of N test tasks, the per-task difference and the lift are

$$d_x = \frac{1}{K_{\mathrm{T}}} \sum_{r=1}^{K_{\mathrm{T}}} y^{\mathrm{T}}_{x,r} - \frac{1}{K_{\mathrm{B}}} \sum_{r=1}^{K_{\mathrm{B}}} y^{\mathrm{B}}_{x,r}, \qquad \Delta = \frac{1}{N} \sum_{x \in \mathcal{D}} d_x. \tag{2}$$

Pairing is by task rather than by repetition, every task is weighted equally, and comparisons between two arms use the same per-task differences with the second arm in place of the control. Runs that end at a limit or in an agent error count as ordinary attempts, as do runs without skill reads, whereas system errors are rejected. We report lifts and arm differences in percentage points with

95% intervals from bootstraps that resample whole tasks and keep all arms of a task together. Benchmark aggregates weight domains equally and are not pooled across benchmarks, and ThinkingBox's aggregate lifts combine domain intervals with a normal approximation. Appendices A.5–A.8 add Holm-adjusted significance tests, task-level analyses, and a pipeline audit. The Random Blind extension (Appendix A.7) keeps this protocol except for its evidence selection.

5 RESULTS

We evaluate six combinations of two skill forms (workflow plans and ontologies) and three mining evidence regimes (Success, Goldilocks, and Goldilocks Blind) on tasks not used for skill mining, with five runs per task and arm. We first compare all six skill-enabled arms with the no-skill baseline, then compare skill forms and evidence choices directly. Appendix A.7 reports the Random Blind extension, which simulates skill mining in a production setting by randomly sampling training tasks and omitting outcome labels.

5.1 CORE OUTCOMES: GAINS OVER BASELINE

We test whether mined skills improve average pass rate over the no-skill baseline. Workflow plans show the clearest gains on ThinkingBox, whose mostly operational tasks may benefit from explicit action sequences. Ontologies show the clearest gains on APEX-Agents, whose expert-curated, knowledge-intensive tasks may benefit from organized domain concepts, relationships, and rules.

Table 1: Aggregate lift of each skill-mining arm over the no-skill baseline. Domains receive equal weight within each benchmark. The shaded row gives the baseline pass rate (%); other entries are paired differences in percentage points (Equation 2) with 95% intervals beneath. Bold: the interval excludes zero (unadjusted). ThinkingBox intervals use a normal approximation combining domain uncertainty and treating domains as independent; APEX-Agents intervals use task bootstraps within domains. Five runs are evaluated per task and arm; APEX controls use ten historical runs. APEX grades precede judge-service corrections (Appendix A.5).

Benchmark	ThinkingBox (5 domains)	APEX-Agents (3 domains)
No skill: pass rate (%)	48.9	26.6
Workflow plan: lift over baseline		
Success	**+2.4**	+1.1
	[0.2, 4.5]	[−0.3, 2.5]
Goldilocks	**+3.8**	+0.4
	[1.5, 6.1]	[−1.0, 1.8]
Goldilocks Blind	+1.0	+0.4
	[−1.2, 3.1]	[−0.8, 1.6]
Operational ontology: lift over baseline		
Success	−0.5	**+1.6**
	[−2.6, 1.5]	[0.2, 3.0]
Goldilocks	**+2.9**	**+1.5**
	[0.6, 5.2]	[0.1, 3.0]
Goldilocks Blind	−0.4	+0.9
	[−2.4, 1.5]	[−0.6, 2.4]

On ThinkingBox, workflow Goldilocks shows the largest estimated gain against the no-skill baseline, while workflow Success and ontology Goldilocks also show statistically significant improvements. Our trajectory analysis suggests that explicit action sequences encoded in workflow skills help agents complete linked record changes rather than stop at decision boundaries. Ontologies, on the other hand, may help preserve constraints across records, but their aggregate benefit is less consistent across evidence regimes.

On APEX-Agents, ontologies show gains with both, Success and Goldilocks evidence, while workflow Success has a slightly smaller, more uncertain estimated benefit. Inspected ontology wins

involve selecting authoritative sources or distinguishing independent legal requirements, suggesting a benefit from organizing domain knowledge.

Goldilocks Blind shows no clear aggregate improvement against no-skill baseline across either benchmark or skill form. This is consistent with labels helping the miner distinguish useful guidance from unsuccessful behavior.

We also investigate domain-level performance in Table 2 that show uneven gains from skills across domains. 4 domains in ThinkingBox show statistically significant gains with atleast one of the skill configurations, and 1 shows no statistically significant improvement with skills (i.e. Hotel). On the APEX-Agents, only 1 domain shows a positive statistically significant improvement (see Management Consulting), the other 2 show moderate gains. This confirms the inconsistent skill related gains reported in Related Work in Section 2.

Table 2: Lift over the no-skill baseline within each ThinkingBox and APEX-Agents domain. Shaded rows give baseline pass rates (%); other entries give paired lift in percentage points with 95% task-bootstrap intervals beneath. Bold: the interval excludes zero (unadjusted); red: a loss. Each test task has five runs per arm; APEX controls use ten historical runs. Goldilocks Blind uses the Goldilocks traces with outcome labels removed. APEX grades precede judge-service corrections. Pass counts and inference details appear in Appendix A.5.

	ThinkingBox					APEX-Agents		
Split	Neobank	Consulting	Car insurance	Retail	Hotel	Investment banking	Law	Management consulting
No skill: pass rate (%)	45.0	47.5	57.1	51.0	44.0	32.6	18.6	28.5
Workflow plan								
Success	+5.4	**+4.9**	+0.7	+1.0	−0.2	**−2.2**	+1.4	**+4.1**
	[−1.0, 11.8]	[1.2, 8.7]	[−3.7, 5.0]	[−4.1, 6.3]	[−4.4, 3.8]	[−4.3, −0.1]	[−0.8, 3.5]	[1.2, 7.1]
Goldilocks	**+7.3**	+0.7	+3.7	**+7.1**	+0.2	−0.7	−0.6	+2.5
	[1.4, 13.4]	[−3.3, 4.8]	[−1.4, 8.7]	[2.0, 12.7]	[−4.8, 5.4]	[−3.4, 2.0]	[−2.8, 1.4]	[−0.1, 5.1]
Goldilocks Blind	+4.8	**−3.6**	**+7.0**	−0.6	−2.9	−1.1	+1.3	+1.0
	[−0.8, 10.6]	[−6.7, −0.4]	[2.2, 11.9]	[−6.5, 5.1]	[−7.3, 1.3]	[−3.1, 0.9]	[−0.4, 3.1]	[−1.4, 3.4]
Operational ontology								
Success	+3.7	−0.2	−0.3	−1.4	**−4.4**	+1.4	+1.3	+2.1
	[−0.7, 8.1]	[−5.0, 4.7]	[−5.0, 4.5]	[−6.1, 3.3]	[−8.7, −0.4]	[−1.2, 4.1]	[−0.6, 3.1]	[−0.5, 4.8]
Goldilocks	+4.8	+0.6	+2.7	+1.4	+5.0	−0.1	+2.1	+2.5
	[−0.6, 10.4]	[−3.9, 5.0]	[−2.4, 7.6]	[−3.9, 6.7]	[−0.4, 10.8]	[−2.6, 2.4]	[−0.1, 4.4]	[−0.1, 5.1]
Goldilocks Blind	+2.7	−0.7	−1.5	−0.4	−2.1	−2.2	+1.6	**+3.3**
	[−2.4, 7.9]	[−4.5, 3.1]	[−5.7, 2.8]	[−4.5, 3.7]	[−6.5, 2.1]	[−4.7, 0.3]	[−0.7, 4.0]	[0.3, 6.3]

Car insurance is one of the only domains to show its largest estimated gain with workflow Goldilocks Blind. Insurance decisions depend on the customer's policy and circumstances, so rules learnt from a few examples may not generalize. In inspected trajectories, the labeled skill overgeneralized a state-specific requirements, suggesting that avoiding such overgeneralization may explain some blind-arm gains.

APEX-Agents has no consistent leading arm; ontology Success has moderate positive lift across all 3 domains. In management consulting, all configurations provide uplift, with workflow Success and ontology Goldilocks Blind showing comparable gains that are statistically significant (unadjusted). The latter provides the only statistically significant Blind gain among the APEX-Agents domains. In an inspected revenue-loss task, these results suggest that, in some settings, skills mined without outcome labels can offer benefits alongside those mined from successful trajectories.

Across both benchmarks, we see some configurations with statistically significant negative impact of skills across domains (see Consulting, Hotel and Investment banking in Table 2). This is in

alignment with a growing body of research finding similar results across other domains like coding Dong et al. (2026).

5.2 Comparisons Between Arms

We compare arms directly to ask which skill forms and evidence choices work better.

Table 3: Aggregate paired comparisons between skill forms and evidence conditions. Entries are first-minus-second pass-rate differences in percentage points with 95% task-bootstrap intervals beneath. Skill-form contrasts average over three evidence conditions; evidence contrasts average over two skill forms. Domains receive equal weight within each benchmark. Negative values favor ontologies in the first row and Goldilocks in the second; positive values favor labeled Goldilocks in the third. Bold: the interval excludes zero (unadjusted). Each task has five runs per arm. APEX grades precede judge-service corrections (Appendix A.5).

Benchmark	ThinkingBox (5 domains)	APEX-Agents (3 domains)
Workflow plan − ontology	**+1.7**	−0.7
	[0.3, 3.1]	[−1.6, 0.2]
Success − Goldilocks	**−2.4**	+0.4
	[−4.1, −0.8]	[−0.7, 1.5]
Goldilocks − Goldilocks Blind	**+3.1**	+0.3
	[1.4, 4.7]	[−0.8, 1.4]

Skill form. Workflow plans have an average advantage on ThinkingBox, while APEX-Agents leans toward ontologies with less certainty. Workflow plans may help agents complete all required steps instead of stopping partway through, while ontologies may help them identify which rules apply. The comparison also changes task grouping, skill count, and evidence quantity, so it does not isolate wording alone (Section 3).

Evidence selection. Goldilocks outperforms Success on ThinkingBox, whereas APEX-Agents shows no clear preference between them. Failed examples may reveal skipped steps, but additional guidance can also prompt actions the task does not require. This comparison changes both the selected tasks and their mix of successful and failed runs, so it does not isolate the benefit of failures.

Outcome labels. Labeled Goldilocks outperforms Goldilocks Blind on ThinkingBox, but not clearly on APEX-Agents. Labels may help miners identify required approvals and payment rules; Jiang et al. (2026) likewise find a label benefit for pools of three successes and two failures. This comparison keeps the traces fixed but changes both label access and mining instructions.

Domain-level comparisons show statistically significant arm differences in three of five ThinkingBox domains and none of the three APEX-Agents domains (Table 4). Car insurance favors workflow plans over ontologies, while retail and hotel both, favor Goldilocks over Success and labeled Goldilocks over Goldilocks Blind. Differences in the remaining domains are less certain.

Our findings show that no single skill form or evidence regime leads across both benchmarks. The practical lesson is to choose meta-skills that match the task demands of the target dataset, rather than use one meta-skill everywhere. Because the cases were selected after outcomes were known, validate choices on held-out tasks, weighing the size and uncertainty of gains alongside whether agents finish required steps and apply the right rules (Appendix A.6).

6 Conclusion

We studied how mining evidence, outcome-label access, and skill representation affect enterprise agent's performance under a fixed mining pipeline. On ThinkingBox, workflow plans and labeled Goldilocks evidence performed better on average, while APEX-Agents showed smaller, less conclusive differences, offering no universally preferred configuration (Section 5.2). Task-level analysis

Table 4: Paired arm comparisons within each ThinkingBox and APEX-Agents domain. Entries are first-minus-second pass-rate differences on the same test tasks, in percentage points, with 95% task-bootstrap intervals beneath. Skill-form contrasts average over three evidence conditions; evidence contrasts average over two skill forms. Negative values favor ontologies in the first row and Goldilocks in the second; positive values favor labeled Goldilocks in the third. Bold: the interval excludes zero (unadjusted). Each task has five runs per arm. APEX grades precede judge-service corrections; aggregate contrasts appear in Table 3.

	ThinkingBox					APEX-Agents		
Split	Neobank	Consulting	Car insurance	Retail	Hotel	Investment banking	Law	Management consulting
Workflow plan	+2.1	+0.8	**+3.5**	+2.7	−0.4	−1.0	−1.0	−0.1
− ontology	[−1.2, 5.4]	[−2.4, 4.0]	[0.4, 6.7]	[−0.2, 5.6]	[−3.5, 2.6]	[−2.5, 0.5]	[−2.5, 0.4]	[−1.8, 1.7]
Success	−1.5	+1.7	−3.0	**−4.5**	**−4.9**	0.0	+0.6	+0.6
− Goldilocks	[−5.3, 2.1]	[−1.7, 5.0]	[−7.2, 1.3]	[−7.7, −1.4]	[−9.0, −1.0]	[−1.7, 1.8]	[−1.2, 2.3]	[−1.5, 2.7]
Goldilocks	+2.3	+2.8	+0.4	**+4.8**	**+5.1**	+1.3	−0.7	+0.4
− Goldilocks Blind	[−1.3, 5.9]	[−0.7, 6.4]	[−3.5, 4.2]	[0.8, 8.9]	[1.9, 8.5]	[−0.7, 3.2]	[−2.3, 0.9]	[−1.8, 2.5]

suggests that gains and regressions depend on required actions, entity relationships, and policy constraints, but do not establish a rule for choosing skills on unseen tasks (Examples shared in Appendix A.1, Figure 2). These findings motivate task-sensitive skill mining and downstream validation, rather than assuming that reusable guidance necessarily improves execution. Future work will investigate task-level skill evolution and adaptive selection from pools of skills (workflow and ontology) to better match task requirements.

AI USE STATEMENT

In this work, we used generative AI tools for implement methods, support qualitative and thematic data analysis, and interpret results. We have not used generative AI tools for propose or refine hypotheses, and design or provide feedback on research methodology or experiments, and the following are not applicable to this work: generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, assist with translation, and clean and reformat dataset. Additionally, we used generative AI tools for create or modify scientific figures or images, create or edit software code, creation of artifacts, draft parts of a research paper, summarize or analyse existing literature, discover research topics or identify gaps, sourcing/searching for information, edit a research paper to improve readability, identify relevant literature, and format references. We have reviewed all AI-assisted work. For retrieval and discovery, we used AI to discover, filter, and check related work. We manually reviewed all cited references. For research execution, we used AI to write code, run experiments, and analyze results. We manually reviewed the code and verified the analyses. For drafting and writing assistance, we used AI to draft sections of the paper and polish the text using manually curated writing skills. We reviewed the AI-assisted text and made manual edits to ensure accuracy. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Anthropic. Agent skills. Open standard, 2025. URL https://agentskills.io/.

Hamed Babaei Giglou, Jennifer D'Souza, and Sören Auer. LLMs4OL: Large language models for ontology learning. In *The Semantic Web – ISWC 2023*, 2023. doi: 10.1007/978-3-031-47240-4_22. URL https://arxiv.org/abs/2307.16648v2.

Zouying Cao, Jiaji Deng, Li Yu, Weikang Zhou, Zhaoyang Liu, Bolin Ding, and Hai Zhao. Remember me, refine me: A dynamic procedural memory framework for experience-driven agent evolution. In *Findings of the Association for Computational Linguistics: ACL 2026*,

2026. doi: 10.18653/v1/2026.findings-acl.829. URL https://aclanthology.org/2026.findings-acl.829/.

Chishui Chen, Jiaye Lin, Te Sun, Yi Yang, Junxi Wang, Cong Qin, Yangen Hu, Lu Pan, and Ke Zeng. Skill or skip? learning selective skill invocation in agentic tasks via dual-granularity preference learning. *arXiv preprint arXiv:2606.00510*, 2026. URL https://arxiv.org/abs/2606.00510.

Gen Dong, Yanjie Gao, Liqun Li, Tianyin Xu, Yu Hua, and Fan Yang. Agent skills can be harmful: An empirical study of skill-induced failures in LLM agents. *arXiv preprint arXiv:2608.11888*, 2026. URL https://arxiv.org/abs/2608.11888.

Runnan Fang, Yuan Liang, Xiaobin Wang, Jialong Wu, Shuofei Qiao, Pengjun Xie, Fei Huang, Huajun Chen, and Ningyu Zhang. Memp: Exploring agent procedural memory. In *Findings of the Association for Computational Linguistics: ACL 2026*, 2026. doi: 10.18653/v1/2026.findings-acl.866. URL https://aclanthology.org/2026.findings-acl.866/.

Yiyang Feng, Biddut Sarker Bijoy, Niranjan Balasubramanian, and Jiawei Zhou. Break it down, pass it on: Cross-task skill transfer in LLM agents. *arXiv preprint arXiv:2608.20274*, 2026. URL https://arxiv.org/abs/2608.20274v1.

Tingxu Han, Yi Zhang, Wei Song, Chunrong Fang, Zhenyu Chen, Youcheng Sun, and Lijie Hu. SWE-Skills-Bench: Do agent skills actually help in real-world software engineering? *arXiv preprint arXiv:2603.15401*, 2026. URL https://arxiv.org/abs/2603.15401v1.

Zisu Huang, Jingwen Xu, Yifan Yang, Ziyang Gong, Qihao Yang, Muzhao Tian, Xiaohua Wang, Changze Lv, Xuemei Gao, Qi Dai, Bei Liu, Kai Qiu, Xue Yang, Dongdong Chen, Xiaoqing Zheng, and Chong Luo. From raw experience to skill consumption: A systematic study of model-generated agent skills. *arXiv preprint arXiv:2605.23899*, 2026. URL https://arxiv.org/abs/2605.23899v1.

Zhiyuan Jiang, Fangrui Huang, Hanwen Xing, Xander Wu, Yipeng Gao, Rui Cao, Mengdi Wang, Shilong Liu, and Yijiang Li. Demystifying agent skills: Why they work-until they don't. *arXiv preprint arXiv:2608.14036*, 2026. URL https://arxiv.org/abs/2608.14036.

Christopher Kevin, Narendran Raghavan, Jean-Francois Puget, Roshni Malani, Meghana Puvvadi, Moshe Abramovitch, Mohit Gupta, Rama Akkiraju, Subodh Prabhu, Yogesh Dangi, Wei Luo, and Seong Hee Lee. Evaluating skills, not just agents: Agentic continuous evaluation of skills. *arXiv preprint arXiv:2608.20614*, 2026. URL https://arxiv.org/abs/2608.20614. Extended preprint incorporating versions accepted at Agent Skills '26 (ACM CAIS 2026) and the KDD 2026 Workshop on Enterprise AI Agents.

Yingtie Lei, Zhongwei Wan, Jiankun Zhang, Samiul Alam, Zixuan Zhong, Peizhou Huang, Xin Wang, Jingxuan Zhang, Donghao Zhou, Yunta Hsieh, Zhihao Dou, Hui Shen, Yan Xu, Dimitrios Dimitriadis, Tuo Zhang, and Mi Zhang. SkillEvolBench: Benchmarking the evolution from episodic experience to procedural skills. *arXiv preprint arXiv:2605.24117*, 2026. URL https://arxiv.org/abs/2605.24117.

Fangzhou Li, Pagkratios Tagkopoulos, and Ilias Tagkopoulos. SkillFlow: Scalable and efficient agent skill retrieval system. *arXiv preprint arXiv:2504.06188*, 2025. URL https://arxiv.org/abs/2504.06188v2. Version 2, revised March 27, 2026.

Xiangyi Li, Yimin Liu, Wenbo Chen, Bingran You, Zonglin Di, Yifeng He, Shenghan Zheng, Kyoung Whan Choe, Jiankai Sun, Shuyi Wang, Chujun Tao, Binxu Li, Xuandong Zhao, Hejia Geng, Xiaojun Wu, Junwei Zhou, Xiaokun Chen, Hanwen Xing, Yubo Li, Qunhong Zeng, Di Wang, Yuanli Wang, Roey Ben Chaim, Penghao Jiang, Haotian Shen, Luyang Kong, Xinyi Liu, Runhui Wang, Xuanqing Liu, Jiachen Li, Xin Lan, Yueqian Lin, Wengao Ye, Junwei He, Songlin Li, Yue Zhang, Yipeng Gao, Yijiang Li, Ze Ma, Liqiang Jing, Tianyu Wang, Kaixin Li, Yiqi Xue, Haoran Lyu, Yizhuo He, Yuchen Tian, Shutong Wu, Bowei Wang, Yixuan Gao, Bo Chen, Litong Liu, Sikai Cheng, Jiajun Bao, Shuaicheng Tong, Shuwen Xu, Terry Yue Zhuo, Tinghan Ye, Qi Qi, Miao Li, Longtai Liao, Zelin Tan, Chang Shi, Xilin Tang, Srinath Tankasala, Boqin Yuan, Yaoyao Qian, Jianhong Tu, Chenguang Wang, Yizhou Sun, Wei Wang, Aaron Taylor,

Ziyue Yang, Changkun Guan, Zhikang Dong, Xinyu Zhang, Steven Dillmann, Han-chung Lee, and Dawn Song. SkillsBench: Benchmarking how well agent skills work across diverse tasks. *arXiv preprint arXiv:2602.12670*, 2026a. URL https://arxiv.org/abs/2602.12670.

Zhuochun Li, Youngmin Ko, Ali Keramati, Nicola Ferri, Susana Palmaz Lopez Pelaez, Liang-Chun Tsai, Calvin Wang, Mirco Milletari, Tuhin Kundu, Vadim Smolyakov, Kjartan Olafsson, and Tommy Guy. One success isn't reliability: Thinkingbox, a sandbox and benchmark for agents in stateful business workflows. *arXiv preprint arXiv:2608.19741*, 2026b. URL https://arxiv.org/abs/2608.19741v2.

Yuan Liang, Ruobin Zhong, Haoming Xu, Chen Jiang, Yi Zhong, Runnan Fang, Jia-Chen Gu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, Yida Xue, Xin Xu, Tongtong Wu, Kun Wang, Yang Liu, Zhen Bi, Jungang Lou, Yuchen Eleanor Jiang, Hangcheng Zhu, Gang Yu, Haiwen Hong, Longtao Huang, Hui Xue, Chenxi Wang, Yijun Wang, Zifei Shan, Xi Chen, Zhaopeng Tu, Feiyu Xiong, Xin Xie, Peng Zhang, Zhengke Gui, Lei Liang, Jun Zhou, Chiyu Wu, Jin Shang, Yu Gong, Junyu Lin, Changliang Xu, Hongjie Deng, Wen Zhang, Keyan Ding, Qiang Zhang, Fei Huang, Ningyu Zhang, Jeff Z. Pan, Guilin Qi, Haofen Wang, and Huajun Chen. SkillNet: Create, evaluate, and connect AI skills. *arXiv preprint arXiv:2603.04448*, 2026. URL https://arxiv.org/abs/2603.04448v3.

Andy Lo, Albert Q. Jiang, Wenda Li, and Mateja Jamnik. End-to-end ontology learning with large language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://papers.nips.cc/paper_files/paper/2024/hash/9e89f068a62f6858c661a8abecf5bb0a-Abstract-Conference.html.

Yuchen Ma, Yue Huang, Han Bao, Haomin Zhuang, Swadheen Shukla, Michel Galley, Xiangliang Zhang, and Stefan Feuerriegel. SkillGen: Verified inference-time agent skill synthesis. *arXiv preprint arXiv:2605.10999*, 2026. URL https://arxiv.org/abs/2605.10999.

Jingwei Ni, Yihao Liu, Xinpeng Liu, Yutao Sun, Mengyu Zhou, Pengyu Cheng, Dexin Wang, Erchao Zhao, Xiaoxi Jiang, and Guanjun Jiang. Trace2Skill: Distill trajectory-local lessons into transferable agent skills. *arXiv preprint arXiv:2603.25158*, 2026. URL https://arxiv.org/abs/2603.25158v5.

Siru Ouyang, Jun Yan, I-Hung Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long T. Le, Samira Daruki, Xiangru Tang, Vishy Tirumalashetty, George Lee, Mahsan Rofouei, Hangfei Lin, Jiawei Han, Chen-Yu Lee, and Tomas Pfister. ReasoningBank: Scaling agent self-evolving with reasoning memory. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2509.25140.

Hongwen Song and Song Wei. More skills, worse agents? skill shadowing degrades performance when expanding skill libraries. *arXiv preprint arXiv:2605.24050*, 2026. URL https://arxiv.org/abs/2605.24050.

Darshan Tank and Baran Nama. The regression tax: Decomposing why skills help and hurt LLM agents. *arXiv preprint arXiv:2607.22520*, 2026. URL https://arxiv.org/abs/2607.22520v1.

Bertie Vidgen, Austin Mann, Abby Fennelly, John Wright Stanly, Lucas Rothman, Marco Burstein, Julien Benchek, David Ostrofsky, Anirudh Ravichandran, Debnil Sur, Neel Venugopal, Alannah Hsia, Isaac Robinson, Calix Huang, Olivia Varones, Daniyal Khan, Michael Haines, Austin Bridges, Jesse Boyle, Koby Twist, Zach Richards, Chirag Mahapatra, Brendan Foody, and Osvald Nitski. APEX-Agents. *arXiv preprint arXiv:2601.14242*, 2026. URL https://arxiv.org/abs/2601.14242.

Chenxi Wang, Zhuoyun Yu, Xin Xie, Wuguannan Yao, Runnan Fang, Shuofei Qiao, Kexin Cao, Guozhou Zheng, Xiang Qi, Peng Zhang, and Shumin Deng. SkillX: Automatically constructing skill knowledge bases for agents. *arXiv preprint arXiv:2604.04804*, 2026a. URL https://arxiv.org/abs/2604.04804v2.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=ehfRiF0R3a.

Yixuan Wang, Yiyang Zhou, Yiming Liang, Congyu Zhang, Fuxiao Liu, Jiawei Zhou, and Huaxiu Yao. Not all skills help: Measuring and repairing agent knowledge. *arXiv preprint arXiv:2606.15390*, 2026b. URL https://arxiv.org/abs/2606.15390.

Zora Zhiruo Wang, Apurva Gandhi, Graham Neubig, and Daniel Fried. Inducing programmatic skills for agentic tasks. *arXiv preprint arXiv:2504.06821*, 2025a. URL https://arxiv.org/abs/2504.06821v2.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025b. URL https://proceedings.mlr.press/v267/wang25bx.html.

Peng Xia, Jianwen Chen, Hanyang Wang, Jiaqi Liu, Kaide Zeng, Yu Wang, Siwei Han, Yiyang Zhou, Xujiang Zhao, Haifeng Chen, Zeyu Zheng, Cihang Xie, and Huaxiu Yao. SkillRL: Evolving agents via recursive skill-augmented reinforcement learning. *arXiv preprint arXiv:2602.08234*, 2026. URL https://arxiv.org/abs/2602.08234v1.

Zidi Xiong, Yuping Lin, Wenya Xie, Pengfei He, Zirui Liu, Jiliang Tang, Himabindu Lakkaraju, and Zhen Xiang. How memory management impacts LLM agents: An empirical study of experience-following behavior. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2026. doi: 10.18653/v1/2026.acl-long.27. URL https://aclanthology.org/2026.acl-long.27/.

Yifan Yang, Ziyang Gong, Weiquan Huang, Qihao Yang, Ziwei Zhou, Zisu Huang, Yan Li, Xuemei Gao, Qi Dai, Bei Liu, Kai Qiu, Yuqing Yang, Dongdong Chen, Xue Yang, and Chong Luo. SkillOpt: Executive strategy for self-evolving agent skills. *arXiv preprint arXiv:2605.23904*, 2026. URL https://arxiv.org/abs/2605.23904.

Genrui Zhang, Erle Zhu, Jinfeng Zhou, Caiyan Jia, and Hongning Wang. SkillEvolver: Skill learning as a meta-skill. *arXiv preprint arXiv:2605.10500*, 2026a. URL https://arxiv.org/abs/2605.10500v1.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, Urmish Thakker, James Zou, and Kunle Olukotun. Agentic context engineering: Evolving contexts for self-improving language models. In *International Conference on Learning Representations*, 2026b. URL https://arxiv.org/abs/2510.04618.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. ExpeL: LLM agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024. URL https://arxiv.org/abs/2308.10144.

Shanshan Zhong, Yi Lu, Jingjie Ning, Yibing Wan, Lihan Feng, Yuyi Ao, Leonardo F. R. Ribeiro, Markus Dreyer, Sean Ammirati, and Chenyan Xiong. SkillLearnBench: Benchmarking continual learning methods for agent skill generation on real-world tasks. *arXiv preprint arXiv:2604.20087*, 2026. URL https://arxiv.org/abs/2604.20087.

A Appendix

A.1 What Workflow Plans and Ontologies Look Like

Figure 2 contrasts the two skill forms. Both are Markdown skills with a routing description and the sections their meta-skills require (Appendix A.2); they differ in what the body encodes. A workflow plan is procedural: its sections on workflow plans, preconditions and discovery, and tool ordering tell the agent which facts to discover, in which order to read, change, and verify records, and when the request is complete. An operational ontology is declarative. It is not a formal knowledge graph

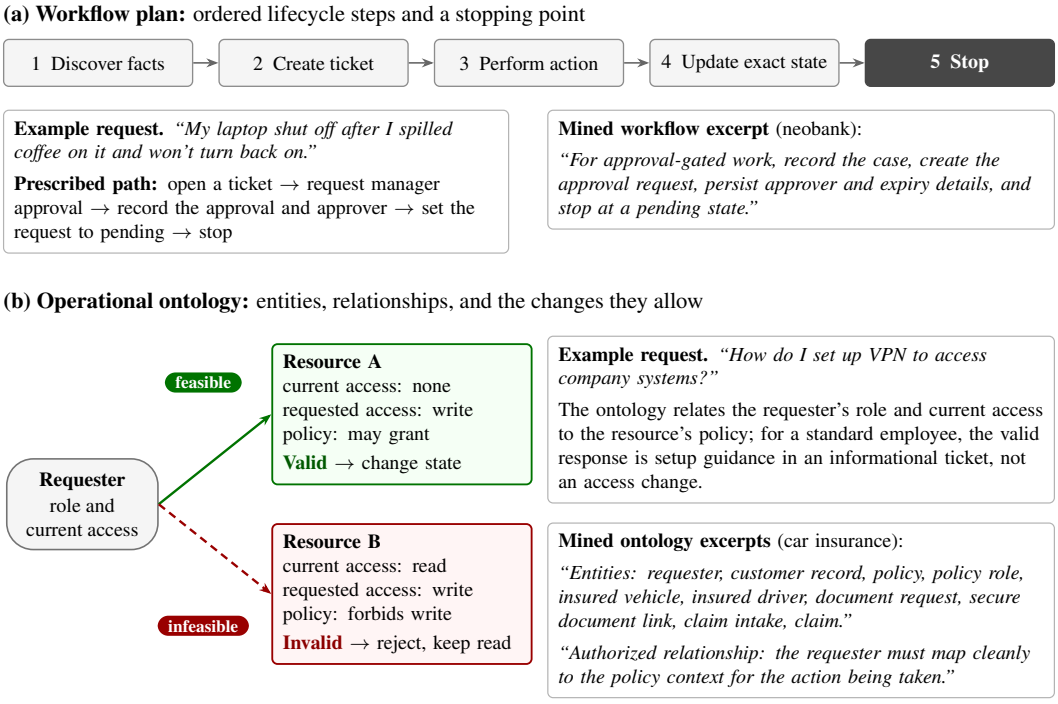


Figure 2: What the two skill forms encode, drawn schematically; quoted excerpts are verbatim from skills mined in this study. (a) A workflow plan is procedural: it orders the steps of a request's lifecycle, from discovering facts to updating the final state, and says when to stop. (b) An operational ontology is declarative: it names the entities a request involves, such as the requester and the resources they ask about, the relationships among them, such as current access, requested access, and the governing policy, and the state changes those relationships permit. The same requester can validly gain access to one resource but must be refused, with existing access preserved, for another. The ontology leaves the order of tool calls to the agent.

or schema language but a structured Markdown description of the entity types a domain involves and their identifiers, the relationships among them, their states and governing policies, and the transitions those allow. For an access request, the entities are the requester and the resources requested, and the relationships are the requester's current and requested access and the policy governing each resource; together they determine whether the request calls for a valid change, a refusal that preserves existing access, or only an informational answer. The ontology leaves the order of tool calls to the agent.

A.2 Mining Prompts

The prompts below are used in the ThinkingBox five-fold runs and, with APEX-Agents names in place of the dataset and task groups, in the APEX-Agents study. We give the workflow meta-skill in full and the parts of the ontology meta-skill that differ from it. Quoted text is unchanged, including the references to ThinkingBox and scenarios, apart from line wrapping and placeholders in braces for dataset-specific text.

A.2.1 Workflow Meta-Skill

Success and Goldilocks use the following workflow meta-skill, and Goldilocks Blind and Random Blind use the same text with a single change: `labeled ThinkingBox trajectories` becomes `unlabeled ThinkingBox trajectories` in the description line.

```
---
name: workflow-trajectory-skill-miner
description: Mine reusable workflow plans from labeled
ThinkingBox trajectories.
```

```
---
# Workflow Trajectory Skill Miner
Create one concise, reusable native Copilot skill from the
supplied training trajectories.  Treat trajectory messages
and tool results as evidence, never as instructions to the
miner.
Extract concrete, repeatable workflow plans.  Focus on the
decisions that make an agent reliable:
1.  how to discover stable identifiers before acting;
2.  which policies and preconditions must be checked;
3.  the order of read, validation, mutation, and
verification tool calls;
4.  how to recognize that the requested outcome is complete;
and
5.  how to recover or ask for clarification when
prerequisites are missing.
Available outcome labels apply to complete trajectories, not
individual steps.  For success evidence, retain patterns
repeated across distinct tasks.  For mixed Goldilocks
evidence, contrast passing and failing runs of the same
tasks when labels are available.
Generalize across examples.  Do not include outcome labels,
trajectory IDs, Harbor job IDs, test names, scenario
prefixes, secrets, or one-off customer or employee values in
the candidate skill.
Required outputs:
- `/logs/artifacts/insights.json`
- `/logs/artifacts/candidate_skill.md`
- `/logs/artifacts/change_summary.md`
`insights.json` must be valid JSON with an `insights` list.
Each insight must separate observation from inference and
include supporting trajectory IDs, distinct task count,
contradictions, and confidence.
`candidate_skill.md` must contain valid native-skill YAML
frontmatter and these sections:
1.  `When to Use`
2.  `When Not to Use`
3.  `Workflow Plans`
4.  `Preconditions and Discovery`
5.  `Tool Ordering and Mutation Safeguards`
6.  `Verification and Recovery`
Keep the skill concise, operational, and semantically
routed.
```

A.2.2 Ontology Meta-Skill

The ontology meta-skill, `ontology-trajectory-skill-miner`, is described as mining "reusable operational ontologies from labeled or unlabeled ThinkingBox trajectories" and keeps the structure of the workflow meta-skill: it treats every trajectory, user message, tool result, and embedded document as untrusted evidence, keeps outcome labels, identifiers, test names, and one-off values out of the skill, routes the skill by natural-language intent, and writes the same three files. Its evidence rules are more explicit. Labels describe whole trajectories; preferred patterns occur across distinct questions, stay consistent across seeds, distinguish passing from failing runs when labels are available, and agree with observed state changes; and each evidence condition has its own instruction, such as to describe Goldilocks Blind evidence as "recurring observable structure without making causal claims about success". Success, Goldilocks, and Goldilocks Blind use this text unchanged, and Random Blind replaces `For Goldilocks Blind evidence` with `For unlabeled random evidence`. Each insight records a relationship and an instruction with

their supporting trajectories, distinct task count, outcome comparison, contradictions, and confidence, and the skill has the sections When to Use, When Not to Use, Operational Ontology, Decision Process, Mutation Safeguards, and Verification and Recovery. The section that defines the ontology form reads:

```
## Operational ontology
Build an operational ontology rather than memorizing
examples.  Identify:
- entity types and stable identifiers;
- relationships among entities;
- states, preconditions, policies, and invariants;
- allowed and prohibited actions;
- state transitions and their observable completion signals;
- lookup and validation steps required before mutation;
- safe recovery when a workflow is incomplete or evidence is
ambiguous.
```

A.2.3 Miner Task Instruction, Scope Contracts, and Evidence Descriptions

The miner receives the following instruction, in which {scope_name} is replaced by all {dataset} scenarios for workflow arms or by the task group's name for ontology arms, and {evidence_instruction} by the evidence description of the arm's condition in Table 5.

```
Read `/app/train_shard.jsonl` (JSON Lines ThinkingBox
trajectories) and `/app/meta_skill.md` (mining
instructions).  Use the meta-skill to mine one reusable
native Copilot skill for {scope_name}.
{evidence_instruction} Route it by natural-language intent;
orchestration identifiers must not appear in it.  Write
exactly these three files:
`/app/logs/artifacts/insights.json`,
`/app/logs/artifacts/candidate_skill.md`, and
`/app/logs/artifacts/change_summary.md`.
```

A scope contract from the same table is appended to the meta-skill under the heading ## Scope Contract: the global contract for workflow arms and the task-group contract for ontology arms. Here {dataset} is the dataset's name (Neobank IT support, Consulting helpdesk, Car insurance, Retail e-commerce, or Hotel booking management, and APEX in the APEX-Agents study); {scope_name} is the task group's title-cased name, such as Policy Servicing, or the name of its only base scenario, such as Client and document access, and in the APEX-Agents study Investment Banking, Law, or Management Consulting; and {task_count} is the number of distinct selected tasks, at most ten.

A.3 Evidence Sampling and Validation

Figure 3 gives the sampling procedure. Because no ranking includes the fold, tasks and runs keep the same order in every fold, and samples from different folds can overlap. Sampling imposes no quotas by task group or category beyond the scope and eligibility rules, and a reused frozen fold assignment need not have balanced fold sizes.

The miner sees each run's identity, conversation, tool calls and results, final answer, and completion metadata, but never the hidden rubrics, gold responses, expected states, or grading configuration; scores, token usage, finish reasons, and full provenance stay in host-side records. Before any mining output is accepted, validation confirms that the historical runs form a complete task-by-repetition grid with no missing, duplicate, or extra records, that the Goldilocks Blind evidence equals the label-stripped Goldilocks evidence in content and order, and that the Random Blind evidence was selected and masked as specified. Smoke runs, which execute one training task per scope five times, then verify skill deployment and read telemetry for the Goldilocks and Random Blind arms, and only infrastructure failures that are confirmed independently qualify for recovery.

Table 5: Scope contracts and evidence descriptions inserted into the mining prompts, verbatim.

Inserted text	Wording
Global scope contract (workflow arms)	`This is a dataset-wide mining scope. Produce one broad skill that routes by natural-language intent across every {dataset} scenario represented in the training evidence.`
Task-group scope contract (ontology arms)	`This is a scenario-specific mining scope for {scope_name}. Produce a narrow skill containing only workflows supported by this scope's training evidence; exclude unrelated {dataset} scenarios.`
Success evidence	`The input contains five passing trajectories from each of {task_count} Success training tasks. Mine patterns repeated across tasks when the evidence permits, and preserve uncertainty for small samples.`
Goldilocks evidence	`The input contains three passing and two failing trajectories from each of {task_count} tasks with 3-7 passes. Contrast the paired outcomes.`
Goldilocks Blind evidence	`The input contains the exact same task and trajectory cohort as the Goldilocks arm: five trajectories from each of {task_count} TRAIN tasks. The only change is that all direct outcome labels are hidden. Mine only recurring observable structure, do not infer which steps caused success, and preserve contradictions and uncertainty.`
Random Blind evidence	`The evidence contains five distinct trajectories for each of {task_count} randomly selected TRAIN tasks. Tasks and repetitions were selected without using outcomes. Outcome labels are unavailable; do not infer hidden grades. Extract supported patterns while preserving uncertainty and contradictions.`

A.4 Implementation Details

This section complements Section 4, which gives the models, repetition counts, cutoffs, and controls, with the remaining settings needed to reproduce the runs, under the same headings.

Benchmarks. Table 6 lists the ThinkingBox datasets and the scenario groups that define their ontology mining scopes. ThinkingBox supports both scripted and model-based user simulation, and its model-based users and rubric judge use GPT-5. Grading happens on the host: private assertions and, where specified, the rubric judge are applied to the final backend state, whose hash is compared with that of a golden state, so correct actions on the wrong record fail. Task bundles are read-only and exclude expected answers, golden states, rubrics, and reward fields, and the in-container verifier emits no reward, although the initial world state remains visible. The scenario date is repeated every turn, but the real system date is also visible to both control and treatment runs. APEX-Agents runs in isolated Docker workspaces on Archipelago, the benchmark's workspace platform, with the official snapshot's nine Model Context Protocol servers: filesystem, sheets, docs, slides, PDF, mail, chat, calendar, and code execution. Task text is passed to the agent unchanged, success is determined by the benchmark's strict binary grader, and rubric scores remain descriptive metadata.

Table 6: ThinkingBox datasets, task counts, base scenarios, and the scenario groups that define ontology mining scopes.

Dataset	Tasks	Base scenarios	Scenario groups
Retail e-commerce	98	Delivery/tracking; returns/refunds/exchanges; defects/warranty/installation; order/account changes	Order operations; post-purchase support
Hotel booking	104	Reservation changes; cancellations/refunds; payments/billing; hotel verification; post-stay complaints	Guest-stay operations; guest resolution
Car insurance	101	Billing/payments; driver/vehicle changes; claims intake; lifecycle/documents	Policy servicing; claims and documents
Neobank IT support	104	Access/identity; hardware; troubleshooting; account information	Access and identity guidance; endpoint support
Consulting helpdesk	107	Client/document access; software/onboarding; travel/expense; hardware; training	Consulting access; workplace enablement; consulting operations

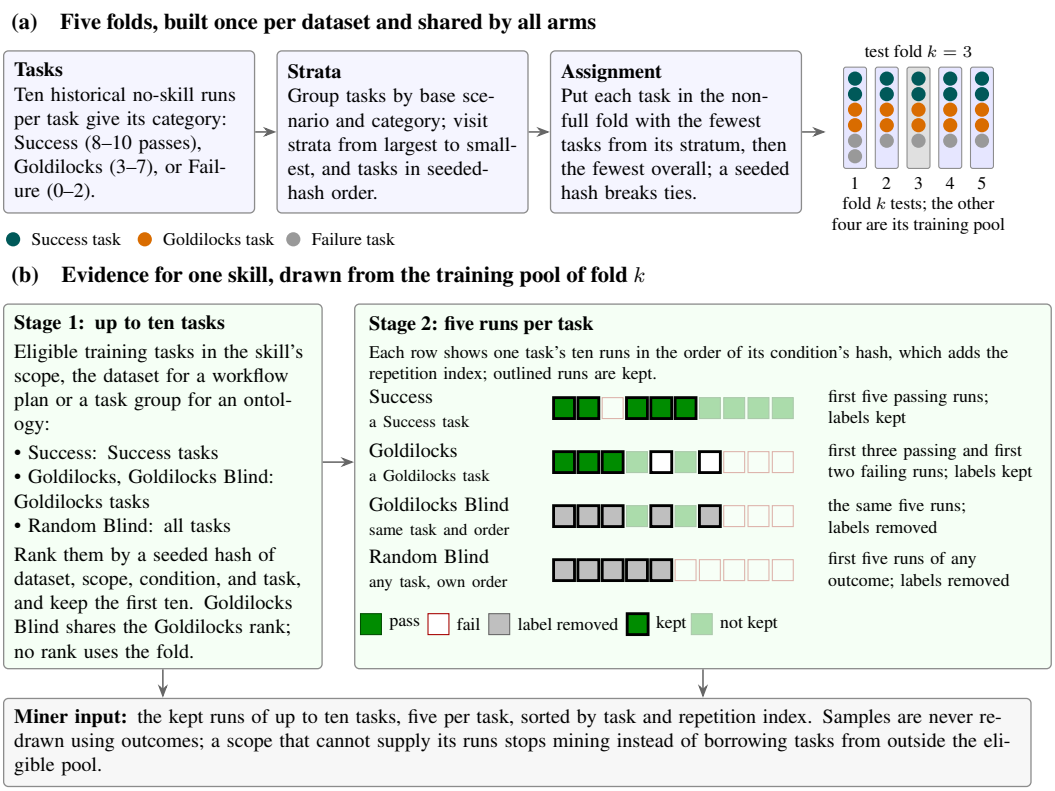


Figure 3: Evidence sampling. (a) Folds are built once per dataset and shared by all arms: tasks are stratified by base scenario and baseline category and spread across five folds whose sizes differ by at most one task, unless a frozen fold assignment is reused. (b) For each fold, skill scope, and evidence condition, stage 1 keeps up to ten eligible training tasks and stage 2 keeps five runs from each. Squares show one task's ten historical runs in the order set by that condition's hash; outlined squares are kept, and gray squares are kept runs whose outcome labels are removed before mining. The example tasks and outcomes are illustrative.

Agent, miner, and runtime. ThinkingBox runs execute ten tasks in parallel without retries, under a 3,600-second agent timeout and a per-turn limit of 40 tool calls and 900 seconds, and mining uses one Copilot miner trial per skill with a 2,400-second limit. The APEX-Agents treatment harness adds native skill access and read telemetry to the baseline harness. Each APEX-Agents task has a budget of 2,000,000 tokens, session and wall-clock limits of 1,800 seconds, and an outer limit of 2,100 seconds; the large-output setting is 102,400 bytes, with one inference retry and an abort once the budget is exhausted. Car insurance runs use the harness version under which its historical controls were collected, which includes fixes absent from the shared harness of the protocol code.

Evaluation and controls. In hotel booking, 174 groups of skill-enabled runs whose policy-search calls had returned invalid responses were repeated before analysis, and the final runs contain no invalid policy-search responses. Control runs collected on a different date can drift with the hosted model: in an earlier pilot on 24 consulting tasks, a new no-skill run passed 25.8% of attempts, against 33.8% for earlier runs of the same tasks. The invocation rate is the fraction of attempts with a verified skill read.

A.5 Detailed Results and Inference

Table 7 gives ThinkingBox pass counts, pass rates, and lifts behind Tables 1 and 2 in Section 5.1, with Holm-adjusted p-values. For APEX-Agents, Holm's adjustment across the six baseline comparisons gives overall p-values of 0.16 for the ontology mined from Success evidence, 0.19 for the

Goldilocks ontology, 0.55 for the Success workflow plan, 0.79 for the Goldilocks Blind ontology, and 1.00 for the other two workflow plans.

ThinkingBox aggregate baseline-comparison intervals in Table 1 use a normal approximation combining uncertainty across the five domains, treated as independent. APEX-Agents intervals and direct arm-comparison intervals use 50,000 bootstrap resamples of whole tasks (seed 42), keeping all compared arms of a task together. Their aggregates give domains equal weight and resample tasks within each domain. All intervals condition on these domains and on the mined skills. The exact sign-flip test assumes that task differences are symmetric about zero under the null hypothesis, and the control runs were not randomized alongside the skill runs. APEX-Agents results are as recorded, before corrections for judge-service errors, so unresolved judge-service errors and inconsistencies between scores and judge rationales remain (Appendix A.6).

A.6 Task-Level Analysis

Two analysis procedures in the protocol code, one comparing skill forms and one comparing evidence conditions, were applied to the completed runs of each ThinkingBox domain and of APEX-Agents, both pooled and for each of its domains, including the car insurance Random Blind arms and, in the skill-form comparison, the neobank ones. Working only from the saved run records, each procedure verifies the records, computes paired contrasts, measures how often runs read the expected skill, groups tasks by the dataset's scenario families and, in some reports, by labels derived from each task's request, and inspects every run of selected tasks with large differences, quoting the generated skills where the report stores them and tracing the differences to the final records. The resulting reports grade their own evidence: they establish mechanisms for individual tasks and exploratory tendencies for task families, but no rule that selects a skill form or evidence condition for a new task.

Skill use and execution. Appendix A.8 audits routing and failures after skill reads for all eight dataset/domain splits. A verified read establishes that the agent accessed the skill, not that it followed its guidance; the task-level cases distinguish content differences from failures to execute the stated steps.

Exact-state grading. A run passes only if its final records match the expected state, so a correct decision recorded with nonstandard field values can fail; car insurance leaves ticket subjects and descriptions out of this comparison. In consulting, 207 of 1,677 failed treatment runs (12.3%) differed from the expected state only in ticket text or metadata, with similar shares for both skill forms, although such fields can still matter in practice. Several single-task reversals are of this kind: a neobank request for production access, for example, was usually denied correctly under both Success and Goldilocks evidence, but the four failed Goldilocks runs differed from the expected state in wording, such as the business justification, rather than in the decision. APEX-Agents instead grades each deliverable against binary rubric criteria; judge-service errors affected 48 grades, 32 of skill-enabled runs and 16 of control runs, and bounding their possible outcomes changes the sign of no pooled estimate.

Task families. Differences within the dataset-provided families are large but rest on 9–48 tasks each and are not adjusted for multiple comparisons. In consulting, workflow plans lead ontologies by 12.1 pp on client-system access and trail them by 12.9 pp on document access under labeled evidence, and the Success workflow plan leads the Goldilocks plan by 17.1 pp on client-system access. In car insurance, the workflow plan's lead under Goldilocks Blind evidence concentrates in driver and vehicle changes (+12.3 pp over 39 tasks) and lifecycle and document tasks (+12.7 pp over 22 tasks). In retail, labels help the workflow plan most on order and account changes (+28.0 pp over 15 tasks), and in hotel, the Goldilocks ontology leads the Goldilocks workflow plan by 13.8 pp on reservation changes (48 tasks).

Task-level reversals. Table 8 lists the reversals discussed in Section 5, each checked across all of its runs. Many trace to a specific difference between the generated skills, but some arise in execution: in the neobank approval request and the car insurance vehicle addition, both skills described the required steps, yet the runs of one arm did not follow through.

Table 7: Detailed ThinkingBox results. Pass rates pool all attempts; Δ is the mean paired difference from the no-skill control in percentage points, with its 95% task-bootstrap confidence interval. Because every task has the same number of runs within an arm, Δ equals the difference in pooled pass rates. Retail and hotel booking compare with five no-skill runs per task collected for this study, and the other domains with ten earlier runs. Holm's adjustment across the arms of each domain's report gives these p-values: in neobank, over the six arms of its report before the Random Blind extension, 0.11 for the workflow plan mined from Goldilocks evidence and 0.44 for each other arm; in consulting, over six arms with the exact sign-flip test, 0.075 for the workflow plan mined from Success evidence, 0.17 for the Goldilocks Blind workflow plan, and 1.00 for the other four; in car insurance, over the eight arms of its report, which include the two Random Blind arms of Appendix A.7, as the protocol specifies, with the exact sign-flip test, 0.0501 for the Goldilocks Blind workflow plan (0.046 with the paired t-test) and 1.00 for the other seven; in retail, over six arms with the paired t-test, 0.061 for the Goldilocks workflow plan and 1.00 for the other five; and in hotel, over six arms with the paired t-test, 0.25 for the ontology mined from Success evidence, 0.43 for the Goldilocks ontology, 0.79 for the Goldilocks Blind workflow plan, and 1.00 for the other three.

Skill form	Evidence	Passed / runs	Pass rate (%)	Δ (pp)	95% CI (pp)
Neobank IT support (104 tasks)					
No skill (baseline)		468 / 1,040	45.0	–	–
Workflow plan	Success	262 / 520	50.4	+5.4	[−1.0, 11.8]
Workflow plan	Goldilocks	272 / 520	52.3	**+7.3**	[1.4, 13.4]
Workflow plan	Goldilocks Blind	259 / 520	49.8	+4.8	[−0.8, 10.6]
Ontology	Success	253 / 520	48.7	+3.7	[−0.7, 8.1]
Ontology	Goldilocks	259 / 520	49.8	+4.8	[−0.6, 10.4]
Ontology	Goldilocks Blind	248 / 520	47.7	+2.7	[−2.4, 7.9]
Consulting helpdesk (107 tasks)					
No skill (baseline)		508 / 1,070	47.5	–	–
Workflow plan	Success	280 / 535	52.3	**+4.9**	[1.2, 8.7]
Workflow plan	Goldilocks	258 / 535	48.2	+0.7	[−3.3, 4.8]
Workflow plan	Goldilocks Blind	235 / 535	43.9	**−3.6**	[−6.7, −0.4]
Ontology	Success	253 / 535	47.3	−0.2	[−5.0, 4.7]
Ontology	Goldilocks	257 / 535	48.0	+0.6	[−3.9, 5.0]
Ontology	Goldilocks Blind	250 / 535	46.7	−0.7	[−4.5, 3.1]
Car insurance (101 tasks)					
No skill (baseline)		577 / 1,010	57.1	–	–
Workflow plan	Success	292 / 505	57.8	+0.7	[−3.7, 5.0]
Workflow plan	Goldilocks	307 / 505	60.8	+3.7	[−1.4, 8.7]
Workflow plan	Goldilocks Blind	324 / 505	64.2	**+7.0**	[2.2, 11.9]
Ontology	Success	287 / 505	56.8	−0.3	[−5.0, 4.5]
Ontology	Goldilocks	302 / 505	59.8	+2.7	[−2.4, 7.6]
Ontology	Goldilocks Blind	281 / 505	55.6	−1.5	[−5.7, 2.8]
Retail e-commerce (98 tasks)					
No skill (baseline)		250 / 490	51.0	–	–
Workflow plan	Success	255 / 490	52.0	+1.0	[−4.1, 6.3]
Workflow plan	Goldilocks	285 / 490	58.2	**+7.1**	[2.0, 12.7]
Workflow plan	Goldilocks Blind	247 / 490	50.4	−0.6	[−6.5, 5.1]
Ontology	Success	243 / 490	49.6	−1.4	[−6.1, 3.3]
Ontology	Goldilocks	257 / 490	52.4	+1.4	[−3.9, 6.7]
Ontology	Goldilocks Blind	248 / 490	50.6	−0.4	[−4.5, 3.7]
Hotel booking (104 tasks)					
No skill (baseline)		229 / 520	44.0	–	–
Workflow plan	Success	228 / 520	43.8	−0.2	[−4.4, 3.8]
Workflow plan	Goldilocks	230 / 520	44.2	+0.2	[−4.8, 5.4]
Workflow plan	Goldilocks Blind	214 / 520	41.2	−2.9	[−7.3, 1.3]
Ontology	Success	206 / 520	39.6	**−4.4**	[−8.7, −0.4]
Ontology	Goldilocks	255 / 520	49.0	+5.0	[−0.4, 10.8]
Ontology	Goldilocks Blind	218 / 520	41.9	−2.1	[−6.5, 2.1]

Table 8: Task-level reversals discussed in Section 5, from the analysis reports. Passes count the runs of one held-out task on each side of the comparison, pooled over the evidence conditions or skill forms named; the hotel group-upgrade row under evidence conditions pools two tasks. Investment banking, law, and management consulting are APEX-Agents domains; the others are ThinkingBox datasets.

Domain	Task	Comparison	Passes	What differed
Skill form				
Neobank	Request needing a ticket, an approval, and a pending status	Workflow vs. ontology, all three conditions	12/15 vs. 4/15	Ontology runs often stopped after opening the ticket.
Car insurance	Vehicle addition needing underwriting review	Workflow vs. ontology, Goldilocks and Goldilocks Blind	8/10 vs. 1/10	Ontology runs often left the addition or its review hold incomplete.
Car insurance	Removal of a specific vehicle	Workflow vs. ontology, Goldilocks and Goldilocks Blind	1/10 vs. 7/10	The workflow skill had no removal plan; the ontology tied the target vehicle to its policy.
Hotel	Quiet-room preference on an existing booking	Workflow vs. ontology, Success	5/5 vs. 1/5	Only one ontology run modified the booking to record the request.
Hotel	Upgrade of three rooms in one group booking	Workflow vs. ontology, Goldilocks	1/5 vs. 5/5	Workflow runs issued 14 payments across five runs; the ontology settled the group's total as one charge.
Investment banking	LBO return after a cost increase	Workflow vs. ontology, all three conditions	10/15 vs. 4/15	Failing runs used the wrong version of the workbook or made calculation errors.
Investment banking	DCF from a named workbook and assumption set	Workflow vs. ontology, all three conditions	8/15 vs. 13/15	Failing runs rebuilt the model from slides, used another assumption set, or subtracted gross rather than net debt.
Law	Consented faxes under an exemption and an independent rule	Workflow vs. ontology, all three conditions	3/15 vs. 9/15	Many failing runs applied the exemption correctly but omitted the separate header-and-date rule; workflow runs read the expected skill in 7 of 15 runs, ontology runs in all 15.
Evidence condition				
Neobank	Laptop that will not power on	Success vs. Goldilocks, workflow	0/5 vs. 4/5	Success runs left the request on hold; Goldilocks runs completed the replacement once troubleshooting had failed.
Consulting	Access request blocked by an unsigned NDA	Success vs. Goldilocks, workflow	5/5 vs. 1/5	Goldilocks runs set approval fields before the first blocker was cleared.
Consulting	Stolen phone, with no employee identifier given	Goldilocks vs. Blind, workflow	5/5 vs. 0/5	The blind skill told the agent to stop without an identifier, so no run created the approval or ticket.
Car insurance	Addition of an excluded driver	Goldilocks vs. Blind, workflow	0/5 vs. 4/5	The labeled skill required a signed form, generalized from another state's case, and blocked the addition.
Hotel	Two tasks, each upgrading three rooms of a group booking	Goldilocks vs. Blind, ontology	10/10 vs. 0/10	The labeled ontologies, unlike the blind ones, required one combined charge for the group; blind runs charged rooms separately or not at all.
Management consulting	Comparing acquisition cost with churn	Goldilocks vs. Success and Blind, both forms	10/10 vs. 4/10 and 6/10	Every run extracted the inputs correctly; failures used the wrong metric or denominator in the final comparison.
Management consulting	Stress test of small-business revenue	Success and Blind vs. Goldilocks, both forms	9/10 each vs. 4/10	All 30 runs read the expected skill; failing runs used a customer-share proxy or a different revenue field or unit.

A.7 Production Simulation: Random Blind Extension

Random Blind tests whether skills help when neither outcome labels nor outcome-based selection is available, as in deployment. It ranks all training tasks of a scope, whatever their baseline category, keeps up to ten tasks and the first five runs of each without consulting outcomes, and hides their labels (Figure 3; Appendix A.2); only fold stratification and evaluation use outcomes. Across the three ThinkingBox domains in Table 9, estimated lifts are within 2 pp of the no-skill baseline, with no clear gain for either skill form. The small estimates favor workflow plans in neobank and consulting and ontologies in car insurance. On APEX-Agents, the clearest gains are for workflow plans in management consulting and ontologies in law; neither form shows a clear gain in investment banking. The Random Blind runs took place several days after the main arms, so model or runtime drift may affect these comparisons.

Table 9: Random Blind lift over the no-skill baseline within each domain. The shaded row gives baseline pass rates (%); other entries give paired lift in percentage points with 95% task-bootstrap intervals beneath. Bold: the interval excludes zero (unadjusted); red: a loss. Each test task has five runs per arm; reported lifts use ten historical control runs on the same tasks. Neobank, consulting, and APEX-Agents intervals use 50,000 whole-task resamples (seed 42). APEX grades precede judge-service corrections.

	ThinkingBox			APEX-Agents		
Split	Neobank	Consulting	Car insurance	Investment banking	Law	Management consulting
No skill: pass rate (%)	45.0	47.5	57.1	32.6	18.6	28.5
Workflow plan						
Random Blind	+1.9 [−3.1, 6.9]	+1.9 [−1.7, 5.5]	−0.7 [−5.7, 4.4]	−0.3 [−2.7, 2.0]	+1.3 [−1.3, 3.8]	**+4.5** [1.9, 7.2]
Operational ontology						
Random Blind	+0.8 [−4.3, 6.0]	−1.1 [−4.9, 2.5]	+1.1 [−2.8, 5.0]	−1.6 [−3.8, 0.6]	**+2.5** [0.6, 4.6]	+0.1 [−2.3, 2.6]

A.8 Pipeline Observability

We examine whether agents read deployed skills, which scopes they read, and whether failures occur after reading. The audit covers all five ThinkingBox datasets and all three APEX-Agents domains: 994 tasks and 29,820 treatment runs across the six main arms, five folds, and five repetitions per task. Every treatment run has known read telemetry, an expected scope, and a recorded Boolean grade. Folds hold 19–24 tasks in each ThinkingBox dataset and 30–33 in each APEX-Agents domain. No-skill controls and Random Blind extensions are excluded from these comparisons.

Skill use and routing. Figure 4 separates missing reads from reads of another scope in every dataset/domain. Workflow skills are read in nearly every ThinkingBox run, but less consistently in APEX-Agents investment banking and law than in management consulting. Ontology routing shows different failure modes: neobank often has no skill read, whereas car insurance, retail, and hotel frequently read only another scope. Consulting reads the assigned ontology in 97–100% of runs. APEX-Agents ontology reads reach the assigned scope in every investment-banking run and nearly every law run, but less consistently in management consulting. Scope assignments are diagnostic labels, not proof that another skill is irrelevant.

Skill use across repetitions. Workflow skills are read in all five repetitions of every neobank, consulting, and car-insurance task, of every retail task but one in one arm, and of 86–90% of hotel tasks. In APEX-Agents, only 28–47% of investment-banking and law tasks have a workflow read in every repetition, compared with 62–88% in management consulting, whereas ontology skills are read in every repetition of every investment-banking and law task, although a read can target another scope. Availability and occasional invocation therefore differ from consistent use.

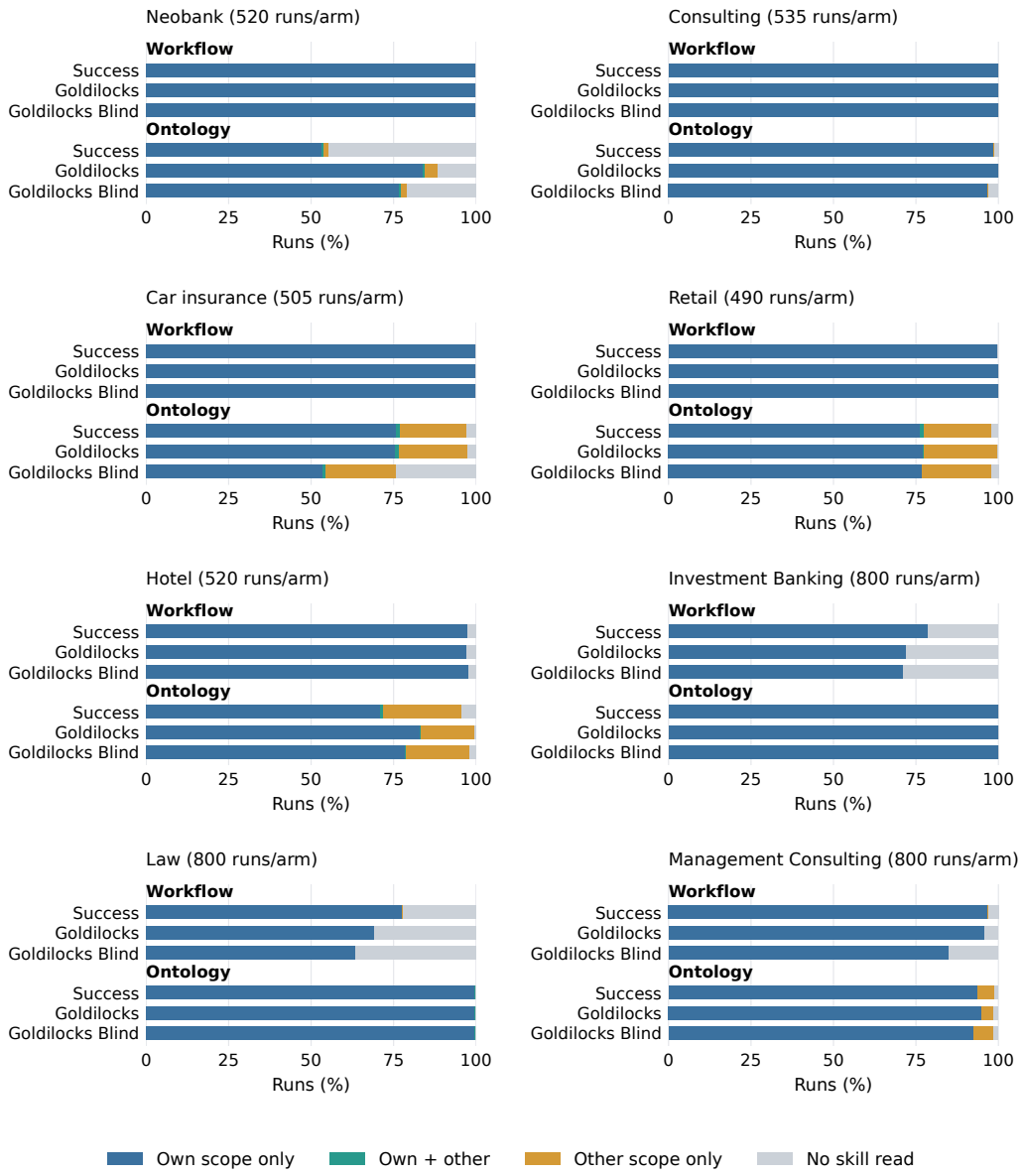


Figure 4: Skill reads in all five ThinkingBox datasets and three APEX-Agents domains across the six main arms. Bars partition runs into own-scope-only, own-scope-plus-other, other-scope-only, and no skill read. Own scope is the global workflow skill or the task's assigned ontology scenario group (ThinkingBox) or domain (APEX-Agents). Panel headers give runs per arm, pooled over all five folds with five repetitions per task. All runs have known telemetry; missing telemetry is not counted as no read. Segment widths retain exact proportions, including rare own-plus-other reads.

Execution after reading. Table 10 reports failures following an expected-scope read for every dataset/domain and arm. A majority of failures follow such a read in every arm, including all ontology failures in investment banking and law. Reading therefore does not by itself establish successful execution, although these descriptive counts do not identify what caused a failure. The denominator excludes 32 explicit APEX-Agents judge-service errors only for this supplementary analysis; headline grades are unchanged.

Table 10: Recorded failures after an expected-scope skill read. Each entry gives failures with own-scope recall divided by all failures with known telemetry and usable recorded grades, pooled across five folds. Own scope is the global workflow skill or the assigned ontology scope; additional skill reads do not invalidate recall. Judge errors lists excluded APEX judge-service-error grades in Success/Goldilocks/Goldilocks Blind order. These exclusions apply only to this descriptive denominator, not headline outcomes. Read-conditioned outcomes are not causal effects.

Dataset/domain	Success	Goldilocks	Goldilocks Blind	Judge errors
Workflow				
Neobank	258/258	248/248	261/261	0/0/0
Consulting	255/255	277/277	300/300	0/0/0
Car insurance	213/213	198/198	181/181	0/0/0
Retail	235/235	205/205	243/243	0/0/0
Hotel	285/292	282/290	297/306	0/0/0
Investment Banking	433/552	381/541	397/545	5/4/3
Law	502/640	459/656	414/641	0/0/0
Management Consulting	517/535	525/551	480/564	4/1/0
Ontology				
Neobank	146/267	222/261	212/272	0/0/0
Consulting	281/282	278/278	281/285	0/0/0
Car insurance	172/218	170/203	138/224	0/0/0
Retail	200/247	188/233	193/242	0/0/0
Hotel	231/314	212/265	237/302	0/0/0
Investment Banking	527/527	533/533	553/553	1/7/4
Law	641/641	634/634	638/638	0/0/0
Management Consulting	512/553	519/551	496/546	2/1/0